aws

Guida per l'utente

Amazon Relational Database Service

Amazon Relational Database Service: Guida per l'utente

Table of Contents

Cos'è Amazon Relational Database Service (Amazon RDS)?

Amazon Relational Database Service (Amazon RDS) è un servizio Web che semplifica la configurazione, l'uso e il dimensionamento di un database relazionale in Cloud AWS. Offre una capacità ridimensionabile a un costo conveniente per un database relazionale standard del settore e gestisce task comuni di amministrazione del database.

> **Note**
>
> In questa guida vengono descritti i motori di database Amazon RDS diversi da Amazon Aurora. Per ulteriori informazioni sull'utilizzo di Amazon Aurora, consulta la Guida per l'utente di Amazon Aurora.

Se non hai familiarità con i prodotti e i servizi AWS, consulta le risorse seguenti per informazioni di base.

- Per una panoramica di tutti i prodotti AWS, consulta Che cos'è il cloud computing?.
- Amazon Web Services fornisce una serie di servizi di database. Per ulteriori informazioni sulle varie opzioni di database disponibili su AWS, consulta Scelta di un servizio di database AWS e Esecuzione dei database su AWS.

Panoramica di Amazon RDS

Perché desideri eseguire un database relazionale in Cloud AWS? Poiché AWS esegue automaticamente molte delle difficili e noiose attività di gestione di un database relazionale.

Argomenti

- Database On-Premise e di Amazon EC2
- Amazon EC2 e Amazon RDS
- Amazon RDS Custom per Oracle e Microsoft SQL Server
- Amazon RDS su AWS Outposts

Database On-Premise e di Amazon EC2

Amazon Elastic Compute Cloud (Amazon EC2): fornisce capacità di calcolo scalabile e sicura in Cloud AWS. Amazon EC2 elimina la necessità di investimenti anticipati in hardware e ti permette di sviluppare e distribuire più rapidamente le applicazioni.

Quando si acquista un server On-Premise, CPU, memoria, storage e IOPS sono tutti disponibili nel bundle. Con Amazon EC2, sono forniti separatamente in modo da poterli dimensionare in modo indipendente l'uno dall'altro. Se hai bisogno di una maggiore quantità di CPU, meno IOPS o più storage, puoi effettuarne l'allocazione con estrema facilità.

Per un database relazionale in un server On-Premise, l'utente assume la piena responsabilità del server, del sistema operativo e del software. Per un database su un'istanza Amazon EC2, AWS gestisce i livelli al di sotto del sistema operativo. In questo modo, Amazon EC2 elimina parte dell'onere della gestione di un server di database On-Premise.

Nella tabella seguente, sono riportati i modelli di gestione dei database On-Premise e Amazon EC2.

Funzionalità	Gestione On-Premise	Gestione di Amazon EC2
Ottimizzazione dell'appl icazione	Customer	Customer
Dimensionamento	Customer	Customer
Elevata disponibilità	Customer	Customer
Backup del database	Customer	Customer
Patching del software del database	Customer	Customer
Installazione del software del database	Customer	Customer
Patching del sistema operativo	Customer	Customer
Installazione del sistema operativo	Customer	Customer

Funzionalità	Gestione On-Premise	Gestione di Amazon EC2
Manutenzione del server	Customer	AWS
Ciclo di vita hardware	Customer	AWS
Alimentazione, rete e raffreddamento	Customer	AWS

Amazon EC2 non è un servizio completamente gestito. Pertanto, quando esegui un database su Amazon EC2, sei più soggetto a errori dell'utente. Ad esempio, quando si aggiorna manualmente il sistema operativo o il software del database, è possibile causare accidentalmente tempi di inattività dell'applicazione. Potresti passare ore a controllare ogni modifica per identificare e risolvere un problema.

Amazon EC2 e Amazon RDS

Amazon RDS è un servizio di database gestito. È responsabile della maggior parte delle attività di gestione. Eliminando le noiose attività manuali, Amazon RDS ti permette di concentrarti sulla tua applicazione e sui tuoi utenti. Consigliamo Amazon RDS su Amazon EC2 come scelta predefinita per la maggior parte delle distribuzioni di database.

Nella tabella seguente, sono riportati i modelli di gestione in Amazon EC2 e Amazon RDS.

Funzionalità	Gestione di Amazon EC2	Gestione di Amazon RDS
Ottimizzazione dell'applicazione	Customer	Customer
Dimensionamento	Customer	AWS
Elevata disponibilità	Customer	AWS
Backup del database	Customer	AWS
Patching del software del database	Customer	AWS

Funzionalità	Gestione di Amazon EC2	Gestione di Amazon RDS
Installazione del software del database	Customer	AWS
Patching del sistema operativo	Customer	AWS
Installazione del sistema operativo	Customer	AWS
Manutenzione del server	AWS	AWS
Ciclo di vita hardware	AWS	AWS
Alimentazione, rete e raffreddamento	AWS	AWS

Amazon RDS offre i seguenti vantaggi specifici rispetto alle distribuzioni di database non completamente gestite:

- Puoi utilizzare i prodotti di database che già conosci: Db2, MariaDB, Microsoft SQL Server, MySQL, Oracle e PostgreSQL.
- Amazon RDS gestisce i backup, l'applicazione di patch software, il rilevamento automatico dei guasti e il ripristino.
- È possibile attivare backup automatici o creare manualmente snapshot di backup personalizzate. Tali backup possono essere utilizzati per ripristinare un database. Il processo di ripristino di Amazon RDS è affidabile ed efficiente.
- È possibile ottenere elevata disponibilità con un'istanza principale e un'istanza secondaria sincrona su cui puoi eseguire il failover in caso di problemi. Puoi anche utilizzare repliche di lettura per aumentare il dimensionamento della lettura.
- Oltre alla sicurezza nel pacchetto di database, puoi controllare chi accede ai database RDS. Per farlo, puoi utilizzare AWS Identity and Access Management (IAM) per definire utenti e autorizzazioni. È anche possibile proteggere i database inserendoli in un Virtual Private Cloud (VPC).

Amazon RDS Custom per Oracle e Microsoft SQL Server

Amazon RDS Custom è un tipo di gestione RDS che offre accesso completo al database e al sistema operativo.

È possibile utilizzare le funzionalità di controllo di RDS Custom per accedere e personalizzare l'ambiente di database e il sistema operativo per applicazioni aziendali legacy e in pacchetti. Nel frattempo, Amazon RDS automatizza le attività e le operazioni di amministrazione del database.

In questo modello di implementazione, è possibile installare applicazioni e modificare le impostazioni di configurazione in base alle applicazioni. Allo stesso tempo, è possibile scaricare le attività di amministrazione del database come provisioning, dimensionamento, aggiornamento e backup su AWS. Puoi sfruttare i vantaggi di gestione del database di Amazon RDS, con maggiore controllo e flessibilità.

Per Oracle Database e Microsoft SQL Server, RDS Custom combina l'automazione di Amazon RDS con la flessibilità di Amazon EC2. Per ulteriori informazioni sui RDS Custom, consulta Utilizzo di Amazon RDS Custom.

Con il modello di responsabilità condivisa di RDS Custom, ottieni più controllo rispetto ad Amazon RDS, ma anche maggiore responsabilità. Per ulteriori informazioni, consulta Modello di responsabilità condivisa in RDS Custom.

Amazon RDS su AWS Outposts

Amazon RDS su AWS Outposts estende i database RDS for SQL Server, RDS for MySQL e RDS for PostgreSQL agli ambienti AWS Outposts. AWS Outposts utilizza lo stesso hardware delle Regioni AWS pubbliche per portare i servizi, l'infrastruttura e i modelli operativi AWS On-Premise. Con RDS in Outposts, è possibile eseguire il provisioning di istanze DB gestite vicino alle applicazioni aziendali che devono essere eseguite in locale. Per ulteriori informazioni, consulta Lavorare con Amazon RDS su AWS Outposts.

Istanze DB

Una istanza database è un ambiente di database isolato in esecuzione in Cloud AWS. L'istanza database rappresenta l'elemento di base di Amazon RDS.

L'istanza database può contenere uno o più database creati dall'utente. Puoi accedere all'istanza database utilizzando gli stessi strumenti e applicazioni che utilizzi con un'istanza database autonoma.

Puoi creare e modificare un'istanza database utilizzando AWS Command Line Interface (AWS CLI), l'API Amazon RDS o la AWS Management Console.

Motori database

Un Motore database è il software di database relazionale specifico in esecuzione nell'istanza database. Attualmente Amazon RDS supporta i seguenti motori:

- Db2
- MariaDB
- Microsoft SQL Server
- MySQL
- Oracle
- PostgreSQL

Ciascun motore di database supporta funzionalità specifiche e ogni versione di un motore di database può includere funzionalità specifiche. Il supporto delle funzionalità Amazon RDS varia tra Regioni AWS e versioni specifiche di ciascun motore di database. Per verificare il supporto delle funzionalità nelle diverse versioni del motore e Regioni, consulta Funzionalità supportate in Amazon RDS by Regione AWS e DB engine.

Inoltre, ogni motore di database dispone di un set di parametri in un gruppo di parametri database in grado di controllare il comportamento dei database gestiti.

Classi di istanze database

La classe di istanza database determina la capacità di calcolo e di memoria di un'istanza database. Una classe di istanza database è costituita sia dal tipo di istanza database che dalla dimensione. Ogni tipo di istanza offre diverse capacità di calcolo, memoria e storage. Ad esempio, db.m6g è un tipo di istanza database per uso generale con processori Graviton2 AWS. Nel tipo di istanza db.m6g, db.m6g.2xlarge è una classe di istanza database.

È possibile selezionare l'istanza database più adatta alle proprie esigenze. Se le tue esigenze cambiano nel tempo, potrai modificare le istanze database. Per informazioni, consulta Classi di istanze database.

> **Note**
>
> Per informazioni sui prezzi delle classi di istanza database, consulta la sezione relativa ai prezzi nella pagina del prodotto Amazon RDS.

Storage delle istanze database

Amazon EBS fornisce volumi di archiviazione durevoli a livello di blocchi, che possono essere collegati a un'istanza in esecuzione. Lo storage di istanza database è disponibile nei seguenti tipi:

- General Purpose (SSD)
- Provisioned IOPS (PIOPS)
- Magnetico

I tipi di storage differiscono per caratteristiche prestazionali e prezzo. È possibile personalizzare le prestazioni e i costi di storage in base alle esigenze del database.

Ciascuna istanza database ha requisiti di storage minimi e massimi in base al tipo di storage e al motore di database supportato. È importante disporre di storage sufficiente in modo che le dimensioni dei database possano aumentare. Inoltre, uno storage sufficiente garantisce che le caratteristiche per il motore database dispongano di spazio sufficiente per scrivere contenuti o registrare voci. Per ulteriori informazioni, consulta Storage delle istanze di database Amazon RDS.

Amazon Virtual Private Cloud (Amazon VPC) (Amazon VPC)

Puoi eseguire un'istanza database su un VPC tramite il servizio Amazon Virtual Private Cloud (Amazon VPC). Quando utilizzi un VPC, hai il controllo completo sull'ambiente virtuale di rete. Puoi scegliere il tuo intervallo di indirizzi IP, creare sottoreti e configurare liste di routing e di controllo accessi. La funzionalità di base di Amazon RDS è la stessa indipendentemente che l'esecuzione avvenga o meno in un VPC. Amazon RDS gestisce i backup, l'applicazione di patch software, il rilevamento automatico dei guasti e il ripristino. Non è previsto alcun costo aggiuntivo per eseguire la tua istanza database in un VPC. Per ulteriori informazioni sull'utilizzo di Amazon VPC con RDS, consulta VPC di Amazon VPC e Amazon RDS.

Amazon RDS utilizza NTP (Network Time Protocol) per sincronizzare l'ora nelle istanze database.

AWSRegioni e zone di disponibilità

Le risorse di cloud computing Amazon sono ospitate in strutture dei data center disponibili in diverse aree nel mondo, ad esempio Nord America, Europa o Asia. La posizione di ogni data center è detta regione AWS.

Ogni regione AWS contiene diverse posizioni chiamate zone di disponibilità o AZ (Availability Zone). Ogni zona di disponibilità è progettata per rimanere isolata dai guasti che si verificano in altre zone di disponibilità. Ciascuna è progettata per fornire una connettività di rete non costosa e a bassa latenza ad altre zone di disponibilità nella stessa regione AWS. Avviando istanze in zone di disponibilità separate, potrai proteggere le tue applicazioni dai guasti di una singola posizione. Per ulteriori informazioni, consulta Regioni, zone di disponibilità e Local Zones.

Grazie a un'opzione nota come implementazione Multi-AZ, è possibile eseguire l'istanza database in varie zone di disponibilità. Quando scegli questa opzione, Amazon effettua automaticamente il provisioning e la gestione di una o più istanzee database in standby secondarie situate in una zona di disponibilità diversa. L'istanza database principale viene replicata tra le zone di disponibilità in ogni istanza database secondaria. Questo approccio consente di fornire ridondanza dei dati e supporto per il failover, eliminare blocchi I/O e ridurre al minimo i picchi di latenza durante i backup di sistema. In un'implementazione di cluster di database Multi-AZ, anche le istanze database secondarie possono gestire il traffico di lettura. Per ulteriori informazioni, consulta Configurazione e gestione di un'implementazione multi-AZ.

Sicurezza

Un gruppo di sicurezza controlla l'accesso a un'istanza database, consentendo l'accesso agli intervalli di indirizzi IP o alle istanze di Amazon EC2 specificati.

Per ulteriori informazioni sui gruppi di sicurezza, consulta Sicurezza in Amazon RDS.

Monitoraggio di Amazon RDS

Esistono vari modi per tenere traccia delle prestazioni e dello stato di un'istanza database. Puoi utilizzare il CloudWatch servizio Amazon per monitorare le prestazioni e lo stato di un'istanza DB. CloudWatch i grafici delle prestazioni sono visualizzati nella console Amazon RDS. Puoi anche eseguire la sottoscrizione agli eventi Amazon RDS per ricevere notifiche relative a modifiche apportate a un'istanza database, uno snapshot DB o un gruppo di parametri database. Per ulteriori informazioni, consulta Monitoraggio di parametri in un'istanza Amazon RDS.

Come utilizzare Amazon RDS

Esistono vari modi per interagire con Amazon RDS.

AWS Management Console

La AWS Management Console è una semplice interfaccia utente basata sul Web. La gestione delle istanze database dalla console non richiede alcuna programmazione. Per accedere alla console Amazon RDS, accedi alla AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.

Interfaccia a riga di comando

Puoi utilizzare AWS Command Line Interface (AWS CLI) per accedere all'API Amazon RDS in modo interattivo. Per installare la AWS CLI, consulta Installazione dell'interfaccia a riga di comando di AWS. Per iniziare a utilizzare la AWS CLI per RDS, consulta Riferimento AWS Command Line Interface per Amazon RDS.

API di Amazon RDS

Gli sviluppatori possono accedere ad Amazon RDS in modo programmatico tramite le API. Per ulteriori informazioni, consulta Documentazione di riferimento dell'API Amazon RDS.

Per lo sviluppo delle applicazioni, è consigliabile utilizzare uno degli SDK (Software Development Kit) AWS. Gli SDK AWS gestiscono dettagli di basso livello, ad esempio autenticazione, logica di ripetizione e gestione errori, consentendoti pertanto di concentrarti sulla logica dell'applicazione. AWS Gli SDK sono disponibili per vari linguaggi. Per ulteriori informazioni, consulta la pagina Strumenti per Amazon Web Services .

AWSIn sono inoltre disponibili librerie, codice di esempio, tutorial e altre risorse, affinché tu possa iniziare con maggiore facilità. Per ulteriori informazioni, consulta la pagina Librerie e codice di esempio.

Come vengono addebitati i costi per Amazon RDS

Quando si utilizza Amazon RDS, è possibile scegliere di utilizzare istanze database on demand o istanze database riservate. Per ulteriori informazioni, consulta Fatturazione delle istanze database per Amazon RDS.

Per informazioni sui prezzi di Amazon RDS, consulta la pagina del prodotto Amazon RDS.

Fasi successive

Nella sezione precedente viene fornita un'introduzione ai componenti dell'infrastruttura di base offerti da RDS. Cosa potrai fare dopo?

Nozioni di base

Crea un'istanza database utilizzando le istruzioni in Nozioni di base su Amazon RDS.

Argomenti specifici per i motori di database

Puoi consultare le informazioni specifiche per un determinato motore di database nelle sezioni indicate di seguito.

- Amazon RDS per Db2
- Amazon RDS for MariaDB
- Amazon RDS for Microsoft SQL Server
- Amazon RDS per MySQL
- Amazon RDS for Oracle
- Amazon RDS per PostgreSQL

Modello di responsabilità condivisa di Amazon RDS

Amazon RDS è responsabile dell'hosting dei componenti software e dell'infrastruttura delle istanze database e dei cluster di database. Tu sei responsabile dell'ottimizzazione delle query, ovvero il processo di ottimizzazione delle query SQL per migliorare le prestazioni. Le prestazioni delle query dipendono fortemente dalla progettazione del database, dalla dimensione dei dati, dalla distribuzione dei dati, dal carico di lavoro dell'applicazione e dai modelli di query, che possono variare notevolmente. Il monitoraggio e l'ottimizzazione sono processi altamente personalizzati che puoi usare per i tuoi database RDS. È possibile utilizzare Approfondimenti sulle prestazioni di Amazon RDS e altri strumenti per identificare le query problematiche.

Istanze DB Amazon RDS

Un'istanza database è un ambiente di database isolato in esecuzione nel cloud. Costituisce l'elemento di base di Amazon RDS. Un'istanza database può contenere più database creati dall'utente ed è possibile accedervi tramite le stesse applicazioni e gli stessi strumenti client utilizzati con un'istanza database standalone. Le istanze database possono facilmente essere create e modificate con gli strumenti della linea di comando AWS, le operazioni dell'API Amazon RDS o la AWS Management Console.

> **Note**
>
> Amazon RDS supporta l'accesso ai database in un'istanza database con qualsiasi applicazione client SQL standard. Amazon RDS non permette l'accesso host diretto.

È possibile avere fino a 40 istanze database Amazon RDS, con le seguenti limitazioni:

- 10 per ogni versione di SQL Server (Enterprise, Standard, Web ed Express) nel modello "license-included" (licenza inclusa)
- 10 per Oracle nel modello "license-included" (licenza inclusa)
- 40 per Db2 con il modello di licenza bring-your-own-license "" (BYOL)
- 40 per MySQL, MariaDB o PostgreSQL
- 40 per Oracle secondo il modello di licenza "bring-your-own-license" (BYOL)

> **Note**
>
> Se per l'applicazione che usi sono necessarie più istanze database, puoi richiedere ulteriori istanze database utilizzando questo modulo.

Ciascuna istanza database dispone di un identificatore istanze DB. Questo nome fornito dal cliente identifica in modo univoco l'istanza database durante l'interazione con l'API Amazon RDS e i comandi della AWS CLI. L'identificatore istanze database deve essere univoco per il cliente in una regione AWS.

L'identificatore dell'istanza database fa parte del nome host DNS allocato all'istanza da RDS. Ad esempio, se specifichi db1 come identificatore dell'istanza database, RDS allocherà

automaticamente un endpoint DNS all'istanza. Un endpoint di esempio è *db1*.*abcdefghijkl*.*us-east-1*.rds.amazonaws.com, dove *db1* è l'ID dell'istanza.

Nell'endpoint di esempio *db1*.*abcdefghijkl*.*us-east-1*.rds.amazonaws.com, la stringa *abcdefghijkl* è un identificatore univoco per una combinazione specifica di Regione AWS e Account AWS. L'identificatore *abcdefghijkl* nell'esempio è generato internamente da RDS e non cambia per la combinazione specificata di regione e account. Pertanto, tutte le istanze database in questa regione condividono lo stesso identificatore fisso. Considera le seguenti funzionalità dell'identificatore fisso:

- Se rinomini l'istanza database, l'endpoint è diverso ma l'identificatore fisso è lo stesso. Ad esempio, se rinomini *db1* in *renamed-db1*, il nuovo endpoint dell'istanza è *renamed-db1*.*abcdefghijkl*.*us-east-1*.rds.amazonaws.com.
- Se elimini e ricrei un'istanza database con lo stesso identificatore di istanza database, l'endpoint è lo stesso.
- Se utilizzi lo stesso account per creare un'istanza database in una regione diversa, l'identificatore generato internamente è diverso perché la regione è diversa, come in *db2*.*mnopqrstuvwx*.*us-west-1*.rds.amazonaws.com.

Ogni istanza database supporta un motore di database. Amazon RDS attualmente supporta i motori di database Db2, MySQL, MariaDB, PostgreSQL, Oracle, Microsoft SQL Server e Amazon Aurora.

Durante la creazione di un'istanza database, alcuni motori di database richiedono che venga specificato un nome di database. Un'istanza DB può ospitare più database, un singolo database Db2 o un singolo database Oracle con più schemi. Il valore relativo al nome del database dipende dal motore di database:

- Per il motore di database Db2, il nome del database è il nome del database ospitato nell'istanza DB. Se desideri utilizzare le stored procedure di Amazon RDS per creare o eliminare un database, non inserire un nome di database quando crei un'istanza DB.
- Per i motori di database MySQL e MariaDB, il nome del database è il nome di un database ospitato nell'istanza database. I database ospitati dalla stessa istanza database devono avere un nome univoco all'interno di essa.
- Per il motore di database Oracle, il nome del database viene utilizzato per impostare il valore di ORACLE_SID, che deve essere fornito al momento della connessione all'istanza Oracle RDS.
- Per il motore di database Microsoft SQL Server, il nome del database non è un parametro supportato.

- Per il motore di database PostgreSQL, il nome del database è il nome di un database ospitato nell'istanza database. Un nome di database non è necessario quando viene creata un'istanza database. I database ospitati dalla stessa istanza database devono avere un nome univoco all'interno di essa.

Amazon RDS crea un account utente master per l'istanza database in uso come parte del processo di creazione. Tale utente master dispone delle autorizzazioni per creare database ed eseguire operazioni di creazione, eliminazione, selezione, aggiornamento e inserimento sulle tabelle che crea. Devi impostare la password dell'utente master quando crei un'istanza database, ma puoi modificarla in qualsiasi momento tramite la AWS CLI, le operazioni API di Amazon RDS o la AWS Management Console. Poi anche utilizzare i comandi SQL standard per modificare la password dell'utente master e gestire gli utenti.

> **Note**
>
> Questa guida tratta i motori di database Amazon RDS non Aurora. Per ulteriori informazioni sull'utilizzo di Amazon Aurora, consulta la Guida per l'utente di Amazon Aurora.

Classi di istanze database

La classe di istanza database determina la capacità di calcolo e memoria di un'istanza database Amazon RDS . La classe di istanza database di cui hai bisogno dipende dalla potenza di elaborazione e dai requisiti di memoria specifici.

Una classe di istanza database è costituita dal tipo di classe di istanza database e dalla dimensione. Ad esempio, db.r6g è un tipo di classe di istanza DB ottimizzato per la memoria alimentato da processori Graviton2. AWS Nel tipo di classe di istanza db.r6g, db.r6g.2xlarge è una classe di istanza database. La dimensione di questa classe è 2xlarge.

Per ulteriori informazioni sui prezzi delle classi di istanza, consulta Prezzi di Amazon RDS.

Argomenti

Tipi di classi di istanza database

Amazon RDS supporta le classi di istanza database per i seguenti casi d'uso:

- Uso generico
- Ottimizzato per la memoria
- Ottimizzato per il calcolo
- Istanze a prestazioni espandibili
- Letture ottimizzate

Per ulteriori informazioni sui tipi di istanza Amazon EC2, consulta Tipi di istanza nella documentazione di Amazon EC2.

Tipo di classe di istanze per uso generico

Sono disponibili le classi di istanza database per uso generico seguenti:

- db.m7g — Classi di istanze DB per uso generico basate su processori Graviton3. AWS Queste classi di istanze forniscono calcolo, memoria e rete bilanciati per un'ampia gamma di carichi di lavoro per uso generico.

  È possibile modificare un'istanza DB per utilizzare una delle classi di istanze DB alimentate dai processori Graviton3. AWS Per farlo, esegui gli stessi passaggi utilizzati per qualsiasi altra modifica dell'istanza database.
- db.m6g — Classi di istanze DB generiche basate su processori Graviton2. AWS Queste istanze forniscono calcolo, memoria e rete bilanciati per un'ampia gamma di carichi di lavoro per uso generico. Le classi di istanza db.m6gd dispongono di archiviazione locale a livello di blocco SSD basato su NVMe per applicazioni che necessitano di archiviazione locale ad alta velocità e bassa latenza.

  È possibile modificare un'istanza DB per utilizzare una delle classi di istanze DB alimentate dai processori Graviton2. AWS Per farlo, esegui gli stessi passaggi utilizzati per qualsiasi altra modifica dell'istanza database.
- dbm6i: classi di istanza database per uso generico basate su processori Intel Xeon scalabili di terza generazione Queste istanze sono certificate SAP e ideali per carichi di lavoro come server di back-end che supportano applicazioni aziendali, server di gioco, parchi istanze di memorizzazione nella cache e ambienti di sviluppo di applicazioni. Le classi di istanza db.m6id e db.m6idn offrono fino a 7,6 TB di archiviazione SSD locale basata su NVMe, mentre la classe db.m6in offre solo archiviazione EBS. Le classi db.m6in e db.m6idn offrono fino a 200 Gbps di larghezza di banda della rete.
- db.m5: classi di istanza database per uso generico che forniscono un rapporto equilibrato tra calcolo, memoria e risorse di rete e rappresentano una buona scelta per numerose applicazioni. La classe di istanza db.m5d offre un'archiviazione SSD basata su NVMe che è fisicamente connessa al server host. Le classi di istanza db.m5 forniscono più capacità di calcolo delle classi di istanza db.m4 precedenti. Sono basate sul nuovo sistema AWS Nitro, una combinazione di hardware dedicato e hypervisor leggeri.
- db.m4: classi di istanza database per uso generico che forniscono più capacità di calcolo delle classi di istanza db.m3 precedenti.

Per i motori database RDS per Oracle, Amazon RDS non supporta più le classi di istanza database db.m4. Se in precedenza sono state create istanze database RDS per Oracle db.m4, Amazon RDS aggiorna automaticamente tali istanze database alle classi di istanza database db.m5 equivalenti.

- db.m3: classi di istanza database per uso generico che forniscono più capacità di calcolo delle classi di istanza db.m1 precedenti.

  Per i motori DB RDS per MariaDB, RDS per MySQL e RDS per PostgreSQL, Amazon RDS ha end-of-life avviato il processo per le classi di istanze DB db.m3 utilizzando la seguente pianificazione, che include consigli di aggiornamento. Per tutte le istanze DB RDS che utilizzano classi di istanze DB db.m3, consigliamo di eseguire l'aggiornamento a una classe di istanze DB di generazione superiore il prima possibile.

Azione o raccomandazione	Date:
Non è più possibile creare istanze database RDS che usano classi di istanza database db.m3.	Adesso
Amazon RDS ha avviato gli aggiornamenti automatici delle istanze database RDS che utilizzano le classi di istanza database db.m3 in classi di istanza database db.m5 equivalenti.	1 febbraio 2023

Tipo di classe di istanza ottimizzata per la memoria

La famiglia Z ottimizzata per la memoria supporta le seguenti classi di istanza:

- db.z1d: classi di istanze ottimizzate per applicazioni a elevato utilizzo di memoria. Queste classi di istanza offrono capacità di calcolo e memoria elevate. Le istanze z1d ad alta frequenza offrono frequenza all-core fino a 4,0 GHz.

La famiglia X ottimizzata per la memoria supporta le seguenti classi di istanza:

- db.x2g — Classi di istanze ottimizzate per applicazioni a uso intensivo di memoria e alimentate da processori Graviton2. AWS Queste classi di istanza offrono un basso costo per GiB di memoria.

È possibile modificare un'istanza DB per utilizzare una delle classi di istanze DB alimentate dai processori Graviton2. AWS Per farlo, esegui gli stessi passaggi utilizzati per qualsiasi altra modifica dell'istanza database.

- db.x2i: classi di istanza ottimizzate per applicazioni a uso intensivo di memoria. I tipi di classi di istanza db.x2iedn e db.x2idn si basano su processori Intel Xeon scalabili di terza generazione (Ice Lake). Includono fino a 3,8 TB di archiviazione SSD NVMe locale, fino a 100 Gbps di larghezza di banda della rete e fino a 4 TiB (db.x2iden) o 2 TiB (db.x2idn) di memoria. Il tipo db.x2iezn si basa su processori Intel Xeon scalabili di seconda generazione (Cascade Lake) con una frequenza turbo all-core fino a 4,5 GHz e fino a 1,5 TiB di memoria.
- db.x1 Classi di istanza di – ottimizzate per applicazioni a uso intensivo di memoria. Queste classi di istanza offrono uno dei prezzi più bassi per GiB di RAM tra le classi di istanza database e fino a 1.952 GiB di memoria di istanza basata su DRAM. Il tipo di classe di istanza db.x1e offre fino a 3.904 GiB di memoria di istanza basata su DRAM.

La famiglia R ottimizzata per la memoria supporta i seguenti tipi di classi di istanza:

- db.r7g — Classi di istanze basate su processori Graviton3. AWS Queste classi di istanza sono ideali per l'esecuzione di carichi di lavoro a uso intensivo di memoria in database open source come MySQL e PostgreSQL.

  È possibile modificare un'istanza DB per utilizzare una delle classi di istanze DB alimentate dai processori Graviton3. AWS Per farlo, esegui gli stessi passaggi utilizzati per qualsiasi altra modifica dell'istanza database.
- db.r6g — Classi di istanze basate su processori Graviton2. AWS Queste classi di istanza sono ideali per l'esecuzione di carichi di lavoro a uso intensivo di memoria in database open source come MySQL e PostgreSQL. Il tipo db.r6gd dispone di archiviazione locale a livello di blocco SSD basato su NVMe per applicazioni che necessitano di archiviazione locale ad alta velocità e bassa latenza.

  È possibile modificare un'istanza DB per utilizzare una delle classi di istanze DB alimentate dai processori Graviton2. AWS Per farlo, esegui gli stessi passaggi utilizzati per qualsiasi altra modifica dell'istanza database.
- db.r6i: classi di istanza ottimizzate basate su processori Intel Xeon scalabili di terza generazione Queste classi di istanza sono certificate SAP e ideali per l'esecuzione di carichi di lavoro a uso intensivo di memoria in database open source come MySQL e PostgreSQL. Le classi di istanze db.r6id, db.r6in e db.r6idn hanno un rapporto CPU di 8:1 e una memoria massima di 1 TiB.

memory-to-v Le classi db.r6id e db.r6idn offrono fino a 7,6 TB di archiviazione SSD basato su NVMe a collegamento diretto, mentre db.r6in offre solo archiviazione EBS. Le classi db.r6idn e db.r6in offrono fino a 200 Gbps di larghezza di banda della rete.

- db.r5b Classi di istanza di – ottimizzate per la memoria per applicazioni a uso intensivo della velocità effettiva. Basate sul sistema AWS Nitro, le istanze db.r5b offrono una larghezza di banda fino a 60 Gbps e 260.000 IOPS di prestazioni EBS. Queste sono le prestazioni di archiviazione a blocchi più veloci su EC2.
- db.r5d: classi di istanze ottimizzate per la bassa latenza, prestazioni I/O casuali molto elevate e velocità effettiva di lettura sequenziale elevata.
- db.r4: classi di istanze ottimizzate per applicazioni a uso intensivo della memoria. Queste classi di istanza offrono reti migliorate e prestazioni . Sono alimentate dal sistema AWS Nitro, una combinazione di hardware dedicato e hypervisor leggero.
- db.r4: classi di istanza che forniscono prestazioni di rete migliorate rispetto alle precedenti classi di istanza db.r3.

  Per i motori RDS per Oracle DB, Amazon RDS ha avviato il end-of-life processo per le classi di istanze DB db.r4 utilizzando la seguente pianificazione, che include consigli di aggiornamento. Per le istanze RDS per Oracle DB che utilizzano classi di istanze db.r4, consigliamo di eseguire l'aggiornamento a una classe di istanze di generazione superiore il prima possibile.

Azione o raccomandazione	Date:
Non è più possibile creare istanze database RDS per Oracle che usano classi di istanza database db.r4.	Adesso
Amazon RDS ha avviato gli aggiornamenti automatici delle istanze database RDS per Oracle che utilizzano le classi di istanza database db.r4 alle classi di istanza database db.r5 equivalenti.	17 aprile 2023

- db.r3: classi di istanze che forniscono l'ottimizzazione della memoria.

  Per i motori DB RDS per MariaDB, RDS per MySQL e RDS per PostgreSQL, Amazon RDS ha end-of-life avviato il processo per le classi di istanze DB db.r3 utilizzando la seguente pianificazione, che include consigli di aggiornamento. Per tutte le istanze DB RDS che utilizzano classi di istanze

DB db.r3, consigliamo di eseguire l'aggiornamento a una classe di istanze DB di generazione superiore il prima possibile.

Azione o raccomandazione	Date:
Non è più possibile creare istanze database RDS che usano classi di istanza database db.r3.	Adesso
Amazon RDS ha avviato gli aggiornamenti automatici delle istanze database RDS che utilizzano le classi di istanza database db.r3 in classi di istanza database db.r5 equivalenti.	1 febbraio 2023

Tipo di classe di istanza ottimizzato per il calcolo

Sono disponibili i seguenti tipi di classi di istanze ottimizzate per il calcolo:

- db.c6gd — Classi di istanze ideali per l'esecuzione di carichi di lavoro avanzati con elaborazione intensiva. Basate sui processori AWS Graviton2, queste classi di istanze offrono storage SSD locale a livello di blocco basato su NVMe per applicazioni che richiedono storage locale ad alta velocità e bassa latenza.

> **Note**
>
> Le classi di istanze c6gd sono supportate solo per le implementazioni di cluster DB Multi-AZ. Sono l'unica classe di istanza supportata per i cluster DB Multi-AZ che offrono la dimensione dell'istanza. `medium` Per ulteriori informazioni, consulta the section called “Implementazioni cluster di database multi-AZ”.

Tipi di classe di istanza a prestazioni espandibili

Sono disponibili i tipi di classe di istanza database a prestazioni espandibili seguenti:

- db.t4g — Classi di istanze generiche basate su processori Graviton2 basati su ARM. AWS Queste classi di istanza offrono prestazioni di prezzo migliori rispetto alle classi di istanza database con prestazioni espandibili della generazione precedente per un ampio set di carichi di lavoro espandibili. Le istanze Amazon RDS di tipo db.t4g sono configurate per la modalità illimitata.

Questo significa che possono espandersi oltre la linea di base in una finestra di 24 ore a un costo aggiuntivo.

È possibile modificare un'istanza DB per utilizzare una delle classi di istanze DB alimentate dai processori Graviton2. AWS Per farlo, esegui gli stessi passaggi utilizzati per qualsiasi altra modifica dell'istanza database.

- db.t3: classi di istanza che forniscono un livello di prestazioni di base, con la possibilità di burst per un utilizzo completo della CPU. Le istanze di tipo db.t3 sono configurate per la modalità illimitata. Queste classi di istanza forniscono più capacità di calcolo rispetto alle classi di istanza db.t2 precedenti. Sono basate sul nuovo sistema AWS Nitro, una combinazione di hardware dedicato e hypervisor leggeri.
- db.t2: classi di istanze che forniscono un livello di prestazioni di base, con la possibilità di burst per un utilizzo completo della CPU. Le istanze db.t2 sono configurate per la modalità Unlimited. Consigliamo di usare queste classi di istanza solo per i server di sviluppo e test o altri server non di produzione.

> **Note**
>
> Le classi di istanze DB che utilizzano il sistema AWS Nitro (db.m5, db.r5, db.t3) sono limitate dal carico di lavoro combinato di lettura e scrittura.

Per le specifiche dell'hardware della classe di istanza database, consultare Specifiche hardware per le classi di istanza database .

Tipo di classe di istanza per letture ottimizzate

I tipi di classe di istanza per letture ottimizzate disponibili sono i seguenti:

- AWS db.r6gd — Classi di istanze basate su processori Graviton2. Queste classi di istanze sono ideali per eseguire carichi di lavoro che richiedono molta memoria e offrono storage SSD locale a livello di blocco basato su NVMe per applicazioni che richiedono storage locale ad alta velocità e bassa latenza.
- db.r6id: classi di istanza ottimizzate basate su processori Intel Xeon scalabili di terza generazione. Queste classi di istanza sono certificate SAP e ideali per l'esecuzione di carichi di lavoro con elevati requisiti di memoria. Offrono una memoria massima di 1 TiB e fino a 7,6 TB di archiviazione SSD basata su NVMe a collegamento diretto.

Motori DB supportati per classi di istanza database

Di seguito vengono elencate alcune considerazioni specifiche sui motori di database per le classi di istanza database:

Db2

Il supporto delle classi di istanze DB varia a seconda della versione e dell'edizione di Db2. Per informazioni sul supporto delle classi di istanza in base a versione ed edizione, consulta RDS per classi di istanze Db2.

Microsoft SQL Server

Il supporto delle classi di istanza database varia a seconda della versione e dell'edizione di SQL Server. Per informazioni sul supporto delle classi di istanza in base a versione ed edizione, consulta Supporto classe istanza database per Microsoft SQL Server.

Oracle

Il supporto delle classi di istanza database varia a seconda della versione e dell'edizione di Oracle Database. RDS per Oracle supporta ulteriori classi di istanze ottimizzate per la memoria. Queste classi hanno i nomi del modulo db.r5.*instance_size*.tpc*threads_per_core*.mem*ratio*. Per il conteggio vCPU e l'allocazione della memoria per ogni classe ottimizzata, consulta Classi di istanza RDS per Oracle supportate.

RDS Custom

Per informazioni sulle classi di istanza database supportate in RDS Custom, consulta Supporto delle classi di istanza database per RDS Custom per Oracle e Supporto delle classi di istanza database per RDS Custom for SQL Server.

Nella tabella seguente puoi trovare i dettagli sulle classi di istanza database di Amazon RDS supportate per ciascun motore del database di Amazon RDS. La cella per ogni motore contiene uno dei seguenti valori:

Sì

La classe di istanza è supportata per tutte le versioni del motore di database.

No

La classe di istanza non è supportata per il motore di database.

specific-versions

La classe di istanza è supportata solo per le versioni di database specificate del motore di database.

Amazon RDS rende obsolete periodicamente le versioni principali e secondarie del motore DB. Non tutti Regioni AWS potrebbero supportare le versioni precedenti del motore. Per informazioni sulle versioni attualmente supportate, consulta gli argomenti relativi ai singoli motori di database: versioni per MariaDB, versioni per Microsoft SQL Server, versioni per MySQL, versioni per Oracle e versioni per PostgreSQL.

Argomenti

- Motori DB supportati per classi di istanze generiche
- Motori DB supportati per classi di istanze ottimizzate per la memoria
- Motori DB supportati per classi di istanze ottimizzate per il calcolo
- Motori DB supportati per classi di istanze con prestazioni espandibili
- Motori DB supportati per le classi di istanze Optimized Reads

Motori DB supportati per classi di istanze generiche

Le tabelle seguenti mostrano i database e le versioni dei database supportati per le classi di istanze generiche.

db.m7g: classi di istanza per uso generico con tecnologia basata su processori AWS Graviton3

Classe di istanza	Db2	MariaDB	Micros SQL Server	MySQL	Orac	PostgreSQL
db.m7g.1(xlarge	No	Versioni MariaDB 10.11, 10.6.10 e versioni successive 10.6, 10.5.17 e versioni 10.5 successiv e e 10.4.26 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successive	No	Tutte le versioni di PostgreSQL 16 e 15, 14.5 e successive 14 versioni e 13.4 e versioni successiv e 13

Classe di istanza	Db2	MariaDB	Micros SQL Server	MySQL	Orac	PostgreSQL
db.m7g.1 xlarge	No	Versioni MariaDB 10.11, 10.6.10 e versioni successive 10.6, 10.5.17 e versioni 10.5 successiv e e 10.4.26 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successive	No	Tutte le versioni di PostgreSQL 16 e 15, 14.5 e successive 14 versioni e 13.4 e versioni successiv e 13
db.m7g.8 large	No	Versioni MariaDB 10.11, 10.6.10 e versioni successive 10.6, 10.5.17 e versioni 10.5 successiv e e 10.4.26 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successive	No	Tutte le versioni di PostgreSQL 16 e 15, 14.5 e successive 14 versioni e 13.4 e versioni successiv e 13
db.m7g.4 large	No	Versioni MariaDB 10.11, 10.6.10 e versioni successive 10.6, 10.5.17 e versioni 10.5 successiv e e 10.4.26 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successive	No	Tutte le versioni di PostgreSQL 16 e 15, 14.5 e successive 14 versioni e 13.4 e versioni successiv e 13
db.m7g.2 large	No	Versioni MariaDB 10.11, 10.6.10 e versioni successive 10.6, 10.5.17 e versioni 10.5 successiv e e 10.4.26 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successive	No	Tutte le versioni di PostgreSQL 16 e 15, 14.5 e successive 14 versioni e 13.4 e versioni successiv e 13

Classe di istanza	Db2	MariaDB	Micros SQL Server	MySQL	Orac	PostgreSQL
db.m7g.xl arge	No	Versioni MariaDB 10.11, 10.6.10 e versioni successive 10.6, 10.5.17 e versioni 10.5 successiv e e 10.4.26 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successive	No	Tutte le versioni di PostgreSQL 16 e 15, 14.5 e successive 14 versioni e 13.4 e versioni successiv e 13
db.m7g.la rge	No	Versioni MariaDB 10.11, 10.6.10 e versioni successive 10.6, 10.5.17 e versioni 10.5 successiv e e 10.4.26 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successive	No	Tutte le versioni di PostgreSQL 16 e 15, 14.5 e successive 14 versioni e 13.4 e versioni successiv e 13

db.m6g: classi di istanze per uso generico basate su processori AWS Graviton2

Classe di istanza	Db2	MariaDB	Micros SQL Server	MySQL	Ora	PostgreSQL
db.m6g.1(xlarge	No	Tutte le versioni di MariadB 10.11, 10.6, 10.5 e 10.4	No	MySQL 8.0.23 e versioni successive	No	Tutte le versioni di PostgreSQL 16, 15, 14 e 13; 12 versioni 12.7 e successive
db.m6g.1: xlarge	No	Tutte le versioni di MariadB 10.11, 10.6, 10.5 e 10.4	No	MySQL 8.0.23 e versioni successive	No	Tutte le versioni di PostgreSQL 16, 15, 14 e 13; 12 versioni 12.7 e successive

Classe di istanza	Db2	MariaDB	Micros SQL Server	MySQL	Ora	PostgreSQL
db.m6g.8 large	No	Tutte le versioni di MariadB 10.11, 10.6, 10.5 e 10.4	No	MySQL 8.0.23 e versioni successive	No	Tutte le versioni di PostgreSQL 16, 15, 14 e 13; 12 versioni 12.7 e successive
db.m6g.4 large	No	Tutte le versioni di MariadB 10.11, 10.6, 10.5 e 10.4	No	MySQL 8.0.23 e versioni successive	No	Tutte le versioni di PostgreSQL 16, 15, 14 e 13; 12 versioni 12.7 e successive
db.m6g.2 large	No	Tutte le versioni di MariadB 10.11, 10.6, 10.5 e 10.4	No	MySQL 8.0.23 e versioni successive	No	Tutte le versioni di PostgreSQL 16, 15, 14 e 13; 12 versioni 12.7 e successive
db.m6g.xl arge	No	Tutte le versioni di MariadB 10.11, 10.6, 10.5 e 10.4	No	MySQL 8.0.23 e versioni successive	No	Tutte le versioni di PostgreSQL 16, 15, 14 e 13; 12 versioni 12.7 e successive
db.m6g.la rge	No	Tutte le versioni di MariadB 10.11, 10.6, 10.5 e 10.4	No	MySQL 8.0.23 e versioni successive	No	Tutte le versioni di PostgreSQL 16, 15, 14 e 13; 12 versioni 12.7 e successive

db.m6gd: classi di istanze generiche basate su processori Graviton2 e storage SSD AWS

Classe di istanza	Db2	MariaDB	Micros SQL Server	MySQL	Ora	PostgreSQL
db.m6gd.1 6xlarge	No	Versioni MariaDB 10.11, 10.6.7 e versioni	No	MySQL 8.0.28 e	No	Tutte le versioni di PostgreSQL 16, 15

Classe di istanza	Db2	MariaDB	Micros SQL Server	MySQL	Ora	PostgreSQL
		successive 10.6, 10.5.16 e versioni 10.5 successiv e e 10.4.25 e versioni 10.4 successive		versioni successive		e 14; 13 versioni 13.7 e successive; e 13.4
db.m6gd.12xlarge	No	Versioni MariaDB 10.11, 10.6.7 e versioni successive 10.6, 10.5.16 e versioni 10.5 successiv e e 10.4.25 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successive	No	Tutte le versioni di PostgreSQL 16, 15 e 14; 13 versioni 13.7 e successive; e 13.4
db.m6gd.8xlarge	No	Versioni MariaDB 10.11, 10.6.7 e versioni successive 10.6, 10.5.16 e versioni 10.5 successiv e e 10.4.25 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successive	No	Tutte le versioni di PostgreSQL 16, 15 e 14; 13 versioni 13.7 e successive; e 13.4
db.m6gd.4xlarge	No	Versioni MariaDB 10.11, 10.6.7 e versioni successive 10.6, 10.5.16 e versioni 10.5 successiv e e 10.4.25 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successive	No	Tutte le versioni di PostgreSQL 16, 15 e 14; 13 versioni 13.7 e successive; e 13.4
db.m6gd.2xlarge	No	Versioni MariaDB 10.11, 10.6.7 e versioni successive 10.6, 10.5.16 e versioni 10.5 successiv e e 10.4.25 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successive	No	Tutte le versioni di PostgreSQL 16, 15 e 14; 13 versioni 13.7 e successive; e 13.4

Classe di istanza	Db2	MariaDB	Micros SQL Server	MySQL	Ora	PostgreSQL
db.m6gd.xlarge	No	Versioni MariaDB 10.11, 10.6.7 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successive	No	Tutte le versioni di PostgreSQL 16, 15 e 14; 13 versioni 13.7 e successive; e 13.4
db.m6gd.large	No	Versioni MariaDB 10.11, 10.6.7 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successive	No	Tutte le versioni di PostgreSQL 16, 15 e 14; 13 versioni 13.7 e successive; e 13.4

db.m6id: classi di istanze generiche basate su processori scalabili Intel Xeon di terza generazione e storage SSD

Classe di istanza	Db2	MariaDB	Micros SQL Server	MySQL	Ora	PostgreSQL
db.m6id.32xlarge	No	MariaDB 10.6.10 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successive	No	Tutte le versioni di PostgreSQL 16 e 15, 14.5 e successive 14 versioni e 13.7 e versioni successive 13
db.m6id.24xlarge	No	MariaDB 10.6.10 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successive	No	Tutte le versioni di PostgreSQL 16 e 15, 14.5 e successive 14 versioni e 13.7 e versioni successive 13

Classe di istanza	Db2	MariaDB	Micros SQL Server	MySQL	Ora	PostgreSQL
db.m6id.16xlarge	No	MariaDB 10.6.10 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successive	No	Tutte le versioni di PostgreSQL 16 e 15, 14.5 e successive 14 versioni e 13.7 e versioni successive 13
db.m6id.12xlarge	No	MariaDB 10.6.10 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successive	No	Tutte le versioni di PostgreSQL 16 e 15, 14.5 e successive 14 versioni e 13.7 e versioni successive 13
db.m6id.8xlarge	No	MariaDB 10.6.10 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successive	No	Tutte le versioni di PostgreSQL 16 e 15, 14.5 e successive 14 versioni e 13.7 e versioni successive 13
db.m6id.4xlarge	No	MariaDB 10.6.10 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successive	No	Tutte le versioni di PostgreSQL 16 e 15, 14.5 e successive 14 versioni e 13.7 e versioni successive 13
db.m6id.2xlarge	No	MariaDB 10.6.10 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successive	No	Tutte le versioni di PostgreSQL 16 e 15, 14.5 e successive 14 versioni e 13.7 e versioni successive 13

Classe di istanza	Db2	MariaDB	Micros SQL Server	MySQL	Ora	PostgreSQL
db.m6id.xlarge	No	MariaDB 10.6.10 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successive	No	Tutte le versioni di PostgreSQL 16 e 15, 14.5 e successive 14 versioni e 13.7 e versioni successive 13
db.m6id.large	No	MariaDB 10.6.10 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successive	No	Tutte le versioni di PostgreSQL 16 e 15, 14.5 e successive 14 versioni e 13.7 e versioni successive 13

db.m6idn: classi di istanza per uso generico con processori scalabili Intel Xeon di terza generazione, archiviazione SSD e ottimizzazione di rete

Classe di istanza	Db	MariaDB	Micros SQL Server	MySQL	Orac	PostgreSQL
db.m6idn.32xlarge	No	MariaDB versione 10.6.8 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successive	No	Tutte le versioni di PostgreSQL 16 e 15, 14.5 e successive 14 versioni e 13.7 e versioni successive 13
db.m6idn.24xlarge	No	MariaDB versione 10.6.8 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successive	No	Tutte le versioni di PostgreSQL 16 e 15, 14.5 e successive 14 versioni e 13.7 e versioni successive 13
db.m6idn.16xlarge	No	MariaDB versione 10.6.8 e versioni successive	No	MySQL 8.0.28 e	No	Tutte le versioni di PostgreSQL 16 e 15,

Classe di istanza	Db	MariaDB	Micros SQL Server	MySQL	Orac	PostgreSQL
		10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive		versioni successive		14.5 e successive 14 versioni e 13.7 e versioni successive 13
db.m6idn. 12xlarge	No	MariaDB versione 10.6.8 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successive	No	Tutte le versioni di PostgreSQL 16 e 15, 14.5 e successive 14 versioni e 13.7 e versioni successive 13
db.m6idn. 8xlarge	Sì	MariaDB versione 10.6.8 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successive	No	Tutte le versioni di PostgreSQL 16 e 15, 14.5 e successive 14 versioni e 13.7 e versioni successive 13
db.m6idn. 4xlarge	Sì	MariaDB versione 10.6.8 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successive	No	Tutte le versioni di PostgreSQL 16 e 15, 14.5 e successive 14 versioni e 13.7 e versioni successive 13
db.m6idn. 2xlarge	Sì	MariaDB versione 10.6.8 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successive	No	Tutte le versioni di PostgreSQL 16 e 15, 14.5 e successive 14 versioni e 13.7 e versioni successive 13
db.m6idn. xlarge	Sì	MariaDB versione 10.6.8 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successive	No	Tutte le versioni di PostgreSQL 16 e 15, 14.5 e successive 14 versioni e 13.7 e versioni successive 13

Classe di istanza	Db	MariaDB	Micros SQL Server	MySQL	Orac	PostgreSQL
db.m6idn.large	Sì	MariaDB versione 10.6.8 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successive	No	Tutte le versioni di PostgreSQL 16 e 15, 14.5 e successive 14 versioni e 13.7 e versioni successive 13

db.m6in: classi di istanze generiche basate su processori scalabili Intel Xeon di terza generazione e ottimizzazione della rete

Classe di istanza	Db	MariaDB	Micros SQL Server	MySQL	Orac	PostgreSQL
db.m6in.32xlarge	No	MariaDB versione 10.6.8 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successive	No	Tutte le versioni di PostgreSQL 16 e 15, 14.3 e successive 14 versioni, 13.7 e successive 13 versioni, 12.11 e successive 12 versioni e 11.16 e versioni 11 successive
db.m6in.24xlarge	No	MariaDB versione 10.6.8 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successive	No	Tutte le versioni di PostgreSQL 16 e 15, 14.3 e successive 14 versioni, 13.7 e successive 13 versioni, 12.11 e successive 12 versioni e 11.16 e versioni 11 successive
db.m6in.16xlarge	No	MariaDB versione 10.6.8 e versioni	No	MySQL 8.0.28 e	No	Tutte le versioni di PostgreSQL 16 e 15,

Classe di istanza	Db	MariaDB	Micros SQL Serve	MySQL	Ora	PostgreSQL
		successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive		versioni successive		14.3 e successive 14 versioni, 13.7 e successive 13 versioni, 12.11 e successive 12 versioni e 11.16 e versioni 11 successive
db.m6in.12xlarge	No	MariaDB versione 10.6.8 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successive	No	Tutte le versioni di PostgreSQL 16 e 15, 14.3 e successive 14 versioni, 13.7 e successive 13 versioni, 12.11 e successive 12 versioni e 11.16 e versioni 11 successive
db.m6in.8xlarge	Sì	MariaDB versione 10.6.8 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successive	No	Tutte le versioni di PostgreSQL 16 e 15, 14.3 e successive 14 versioni, 13.7 e successive 13 versioni, 12.11 e successive 12 versioni e 11.16 e versioni 11 successive
db.m6in.4xlarge	Sì	MariaDB versione 10.6.8 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successive	No	Tutte le versioni di PostgreSQL 16 e 15, 14.3 e successive 14 versioni, 13.7 e successive 13 versioni, 12.11 e successive 12 versioni e 11.16 e versioni 11 successive

Classe di istanza	Db	MariaDB	Micros SQL Serve	MySQL	Ora	PostgreSQL
db.m6in.2xlarge	Sì	MariaDB versione 10.6.8 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successive	No	Tutte le versioni di PostgreSQL 16 e 15, 14.3 e successive 14 versioni, 13.7 e successive 13 versioni, 12.11 e successive 12 versioni e 11.16 e versioni 11 successive
db.m6in.xlarge	Sì	MariaDB versione 10.6.8 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successive	No	Tutte le versioni di PostgreSQL 16 e 15, 14.3 e successive 14 versioni, 13.7 e successive 13 versioni, 12.11 e successive 12 versioni e 11.16 e versioni 11 successive
db.m6in.large	Sì	MariaDB versione 10.6.8 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successive	No	Tutte le versioni di PostgreSQL 16 e 15, 14.3 e successive 14 versioni, 13.7 e successive 13 versioni, 12.11 e successive 12 versioni e 11.16 e versioni 11 successive

db.m6i: classi di istanze generiche basate su processori scalabili Intel Xeon di terza generazione

Classe di istanza	Db	MariaDB	Micros SQL Server	MySQL	Oracle	PostgreSQL
db.m6i.32 xlarge	Sì	Versioni MariaDB 10.11, 10.6.7 e versioni successive 10.6, 10.5.15 e versioni 10.5 successive e 10.4.24 e versioni 10.4 successive	Sì	MySQL 8.0.28 e versioni successive	Oracle Databa 19c	Tutte le versioni di PostgreSQL 16, 15 e 14; 13.4, 12.8 e 11.13 e versioni successive 11
db.m6i.24 xlarge	Sì	Versioni MariaDB 10.11, 10.6.7 e versioni successive 10.6, 10.5.15 e versioni 10.5 successive e 10.4.24 e versioni 10.4 successive	Sì	MySQL 8.0.28 e versioni successive	Oracle Databa 19c	Tutte le versioni di PostgreSQL 16, 15 e 14; 13.4, 12.8 e 11.13 e versioni successive 11
db.m6i.16 xlarge	Sì	Versioni MariaDB 10.11, 10.6.7 e versioni successive 10.6, 10.5.15 e versioni 10.5 successive e 10.4.24 e versioni 10.4 successive	Sì	MySQL 8.0.28 e versioni successive	Oracle Databa 19c	Tutte le versioni di PostgreSQL 16, 15 e 14; 13.4, 12.8 e 11.13 e versioni successive 11
db.m6i.12 xlarge	Sì	Versioni MariaDB 10.11, 10.6.7 e versioni successive 10.6, 10.5.15 e versioni 10.5 successive e 10.4.24 e versioni 10.4 successive	Sì	MySQL 8.0.28 e versioni successive	Oracle Databa 19c	Tutte le versioni di PostgreSQL 16, 15 e 14; 13.4, 12.8 e 11.13 e versioni successive 11

Classe di istanza	Db	MariaDB	Micros SQL Server	MySQL	Oracle	PostgreSQL
db.m6i.8x large	Sì	Versioni MariaDB 10.11, 10.6.7 e versioni successive 10.6, 10.5.15 e versioni 10.5 successive e 10.4.24 e versioni 10.4 successive	Sì	MySQL 8.0.28 e versioni successive	Oracle Databa 19c	Tutte le versioni di PostgreSQL 16, 15 e 14; 13.4, 12.8 e 11.13 e versioni successive 11
db.m6i.4x large	Sì	Versioni MariaDB 10.11, 10.6.7 e versioni successive 10.6, 10.5.15 e versioni 10.5 successive e 10.4.24 e versioni 10.4 successive	Sì	MySQL 8.0.28 e versioni successive	Oracle Databa 19c	Tutte le versioni di PostgreSQL 16, 15 e 14; 13.4, 12.8 e 11.13 e versioni successive 11
db.m6i.2x large	Sì	Versioni MariaDB 10.11, 10.6.7 e versioni successive 10.6, 10.5.15 e versioni 10.5 successive e 10.4.24 e versioni 10.4 successive	Sì	MySQL 8.0.28 e versioni successive	Oracle Databa 19c	Tutte le versioni di PostgreSQL 16, 15 e 14; 13.4, 12.8 e 11.13 e versioni successive 11
db.m6i.xl arge	Sì	Versioni MariaDB 10.11, 10.6.7 e versioni successive 10.6, 10.5.15 e versioni 10.5 successive e 10.4.24 e versioni 10.4 successive	Sì	MySQL 8.0.28 e versioni successive	Oracle Databa 19c	Tutte le versioni di PostgreSQL 16, 15 e 14; 13.4, 12.8 e 11.13 e versioni successive 11

Classe di istanza	Db	MariaDB	Micros SQL Server	MySQL	Oracle	PostgreSQL
db.m6i.large	Sì	Versioni MariaDB 10.11, 10.6.7 e versioni successive 10.6, 10.5.15 e versioni 10.5 successive e 10.4.24 e versioni 10.4 successive	Sì	MySQL 8.0.28 e versioni successive	Oracle Databa 19c	Tutte le versioni di PostgreSQL 16, 15 e 14; 13.4, 12.8 e 11.13 e versioni successive 11

db.m5d: classi di istanze generiche basate su processori Intel Xeon Platinum e storage SSD

Classe di istanza	Db	MariaDB	Micros SQL Server	MySQL	Ora	PostgreSQL
db.m5d.24 xlarge	No	Versioni MariaDB 10.11, 10.6.7 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	Sì	MySQL 8.0.28 e versioni successive	Sì	Tutte le versioni di PostgreSQL 16 e 15, 14.5 e successive 14 versioni, 13.7 e successive 13 versioni e 13.4
db.m5d.16 xlarge	No	Versioni MariaDB 10.11, 10.6.7 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	Sì	MySQL 8.0.28 e versioni successive	Sì	Tutte le versioni di PostgreSQL 16 e 15, 14.5 e successive 14 versioni, 13.7 e successive 13 versioni e 13.4
db.m5d.12 xlarge	No	Versioni MariaDB 10.11, 10.6.7 e versioni successive 10.6, 10.5.16 e versioni 10.5 successiv	Sì	MySQL 8.0.28 e versioni successive	Sì	Tutte le versioni di PostgreSQL 16 e 15, 14.5 e successive 14 versioni, 13.7 e successive 13 versioni e 13.4

Classe di istanza	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
		e e 10.4.25 e versioni 10.4 successive				
db.m5d.8xlarge	No	Versioni MariaDB 10.11, 10.6.7 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	Sì	MySQL 8.0.28 e versioni successive	Sì	Tutte le versioni di PostgreSQL 16 e 15, 14.5 e successive 14 versioni, 13.7 e successive 13 versioni e 13.4
db.m5d.4xlarge	No	Versioni MariaDB 10.11, 10.6.7 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	Sì	MySQL 8.0.28 e versioni successive	Sì	Tutte le versioni di PostgreSQL 16 e 15, 14.5 e successive 14 versioni, 13.7 e successive 13 versioni e 13.4
db.m5d.2xlarge	No	Versioni MariaDB 10.11, 10.6.7 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	Sì	MySQL 8.0.28 e versioni successive	Sì	Tutte le versioni di PostgreSQL 16 e 15, 14.5 e successive 14 versioni, 13.7 e successive 13 versioni e 13.4
db.m5d.xlarge	No	Versioni MariaDB 10.11, 10.6.7 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	Sì	MySQL 8.0.28 e versioni successive	Sì	Tutte le versioni di PostgreSQL 16 e 15, 14.5 e successive 14 versioni, 13.7 e successive 13 versioni e 13.4

Classe di istanza	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.m5d.large	No	Versioni MariaDB 10.11, 10.6.7 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	Sì	MySQL 8.0.28 e versioni successive	Sì	Tutte le versioni di PostgreSQL 16 e 15, 14.5 e successive 14 versioni, 13.7 e successive 13 versioni e 13.4

db.m5 — classi di istanze per uso generico, processori Intel Xeon Platinum da 2,5 GHz

Classe di istanza	Db2	Maria	Microsoft SQL Server	MyS	Oracl	PostgreSQL
db.m5.24x large	No	Sì	Sì	Sì	Sì	Tutte le versioni di PostgreSQL 16, 15, 14, 13, 12 e 11; 10 versioni 10.17 e successive; e 9.6.22 e versioni successive 9
db.m5.16x large	No	Sì	Sì	Sì	Sì	Tutte le versioni di PostgreSQL 16, 15, 14, 13, 12 e 11; 10 versioni 10.17 e successive; e 9.6.22 e versioni successive 9
db.m5.12x large	No	Sì	Sì	Sì	Sì	Tutte le versioni di PostgreSQL 16, 15, 14, 13, 12 e 11; 10 versioni 10.17 e successive; e 9.6.22 e versioni successive 9
db.m5.8xlarge	No	Sì	Sì	Sì	Sì	Tutte le versioni di PostgreSQL 16, 15, 14, 13, 12 e 11; 10 versioni 10.17 e successive; e 9.6.22 e versioni successive 9
db.m5.4xlarge	No	Sì	Sì	Sì	Sì	Tutte le versioni di PostgreSQL 16, 15, 14, 13, 12 e 11; 10 versioni 10.17 e successive; e 9.6.22 e versioni successive 9

Classe di istanza	Db2	Maria	Microsoft SQL Server	MyS	Oracl	PostgreSQL
db.m5.2xlarge	No	Sì	Sì	Sì	Sì	Tutte le versioni di PostgreSQL 16, 15, 14, 13, 12 e 11; 10 versioni 10.17 e successive; e 9.6.22 e versioni successive 9
db.m5.xlarge	No	Sì	Sì	Sì	Sì	Tutte le versioni di PostgreSQL 16, 15, 14, 13, 12 e 11; 10 versioni 10.17 e successive; e 9.6.22 e versioni successive 9
db.m5.large	No	Sì	Sì	Sì	Sì	Tutte le versioni di PostgreSQL 16, 15, 14, 13, 12 e 11; 10 versioni 10.17 e successive; e 9.6.22 e versioni successive 9

db.m4: classi di istanze generiche con processori Intel Xeon

Classe di istanza	Db	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.m4.16large	No	Deprecated	Sì	Obsoleta	Obsol	Inferiore a PostgreSQL 13
db.m4.10large	No	Deprecated	Sì	Obsoleta	Obsol	Inferiore a PostgreSQL 13
db.m4.4xarge	No	Deprecated	Sì	Obsoleta	Obsol	Inferiore a PostgreSQL 13
db.m4.2xarge	No	Deprecated	Sì	Obsoleta	Obsol	Inferiore a PostgreSQL 13

Classe di istanza	Db	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.m4.xlarge	No	Deprecated	Sì	Obsoleta	Obsol	Inferiore a PostgreSQL 13
db.m4.large	No	Deprecated	Sì	Obsoleta	Obsol	Inferiore a PostgreSQL 13

db.m3: classi di istanze per uso generico

Classe di istanza	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.m3.2xlarge	No	No	Sì	Sì	Obsoleta	Deprecated
db.m3.xlarge	No	No	Sì	Sì	Obsoleta	Deprecated
db.m3.large	No	No	Sì	Sì	Obsoleta	Deprecated
db.m3.medium	No	No	Sì	Sì	Obsoleta	Deprecated

Motori DB supportati per classi di istanze ottimizzate per la memoria

Le tabelle seguenti mostrano i database e le versioni dei database supportati per le classi di istanze ottimizzate per la memoria.

db.z1d: classi di istanze ottimizzate per la memoria

Classe di istanza	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.z1d.1 xlarge	No	No	Sì	No	Sì	No
db.z1d.6 large	No	No	Sì	No	Sì	No
db.z1d.3 large	No	No	Sì	No	Sì	No
db.z1d.2 large	No	No	Sì	No	Sì	No
db.z1d.xl arge	No	No	Sì	No	Sì	No
db.z1d.la rge	No	No	Sì	No	Sì	No

db.x2g — classi di istanze ottimizzate per la memoria basate su processori Graviton2 AWS

Classe di istanza	Db	MariaDB	Micros SQL Serve	MySQL	Ora	PostgreSQL
db.x2g.1 xlarge	No	Tutte le versioni di MariadB 10.11, 10.6, 10.5 e 10.4	No	MySQL 8.0.25 e versioni successive	No	Tutte le versioni di PostgreSQL 16, 15, 14 e 13; 12 versioni 12.7 e successive
db.x2g.1 xlarge	No	Tutte le versioni di MariadB 10.11, 10.6, 10.5 e 10.4	No	MySQL 8.0.25 e versioni successive	No	Tutte le versioni di PostgreSQL 16, 15, 14 e 13; 12 versioni 12.7 e successive

Classe di istanza	Db	MariaDB	Micros SQL Serve	MySQL	Ora	PostgreSQL
db.x2g.8 large	No	Tutte le versioni di MariadB 10.11, 10.6, 10.5 e 10.4	No	MySQL 8.0.25 e versioni successive	No	Tutte le versioni di PostgreSQL 16, 15, 14 e 13; 12 versioni 12.7 e successive
db.x2g.4 large	No	Tutte le versioni di MariadB 10.11, 10.6, 10.5 e 10.4	No	MySQL 8.0.25 e versioni successive	No	Tutte le versioni di PostgreSQL 16, 15, 14 e 13; 12 versioni 12.7 e successive
db.x2g.2 large	No	Tutte le versioni di MariadB 10.11, 10.6, 10.5 e 10.4	No	MySQL 8.0.25 e versioni successive	No	Tutte le versioni di PostgreSQL 16, 15, 14 e 13; 12 versioni 12.7 e successive
db.x2g.x arge	No	Tutte le versioni di MariadB 10.11, 10.6, 10.5 e 10.4	No	MySQL 8.0.25 e versioni successive	No	Tutte le versioni di PostgreSQL 16, 15, 14 e 13; 12 versioni 12.7 e successive
db.x2g.la rge	No	Tutte le versioni di MariadB 10.11, 10.6, 10.5 e 10.4	No	MySQL 8.0.25 e versioni successive	No	Tutte le versioni di PostgreSQL 16, 15, 14 e 13; 12 versioni 12.7 e successive

db.x2idn: classi di istanza ottimizzate per la memoria basate su processori Intel Xeon scalabili di terza generazione

Classe di istanza	Db	MariaDB	Micros SQL Server	MySQL	Oracle	PostgreSQL
db.x2idn.32xlarge	No	Tutte le versioni di MariaDB 10.11, 10.6.7 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successe	Solo Enterp e Edition	Versioni PostgreSQL 15, 14.6 e 13.9
db.x2idn.24xlarge	No	Tutte le versioni di MariaDB 10.11, 10.6.7 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successe	Solo Enterp e Edition	Versioni PostgreSQL 15, 14.6 e 13.9
db.x2idn.16xlarge	No	Tutte le versioni di MariaDB 10.11, 10.6.7 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successe	Solo Enterp e Edition	Versioni PostgreSQL 15, 14.6 e 13.9

db.x2iedn: classi di istanza ottimizzate per la memoria con SSD locali basati su NVMe, basate su processori Intel Xeon scalabili di terza generazione

Classe di istanza	DI	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.x2iedn.32xlarge	Sì	Tutte le versioni di MariaDB 10.11, 10.6.7 e versioni successive 10.6, 10.5.16 e versioni 10.5	Utilizza solo Standard Edition ed	MySQL 8.0.28 e versioni	Solo Enterp e Edition	Tutte le versioni di PostgreSQL 16 e 15, 14.5 e successive 14 versioni, 13.7 e

Classe di istanza	DI	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
		successive e 10.4.25 e versioni 10.4 successive	Enterprise Edition, SQL Server 2014 versione 12.00 e versioni successive	success e		successive 13 versioni e 13.4
db.x2iedn .24xlarge	Sì	Tutte le versioni di MariaDB 10.11, 10.6.7 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	Utilizza solo Standard Edition ed Enterprise Edition, SQL Server 2014 versione 12.00 e versioni successive	MySQL 8.0.28 e versioni success e	Solo Enterp e Editior	Tutte le versioni di PostgreSQL 16 e 15, 14.5 e successive 14 versioni, 13.7 e successive 13 versioni e 13.4

Classe di istanza	DI	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.x2iedn .16xlarge	Sì	Tutte le versioni di MariaDB 10.11, 10.6.7 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	Utilizza solo Standard Edition ed Enterpris e Edition, SQL Server 2014 versione 12.00 e versioni successive	MySQL 8.0.28 e versioni success e	Solo Enterp e Editior	Tutte le versioni di PostgreSQL 16 e 15, 14.5 e successive 14 versioni, 13.7 e successive 13 versioni e 13.4
db.x2iedn .8xlarge	Sì	Tutte le versioni di MariaDB 10.11, 10.6.7 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	Utilizza solo Standard Edition ed Enterpris e Edition, SQL Server 2014 versione 12.00 e versioni successive	MySQL 8.0.28 e versioni success e	Solo Enterp e Editior	Tutte le versioni di PostgreSQL 16 e 15, 14.5 e successive 14 versioni, 13.7 e successive 13 versioni e 13.4

Classe di istanza	DI	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.x2iedn .4xlarge	Sì	Tutte le versioni di MariaDB 10.11, 10.6.7 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	Utilizza solo Standard Edition ed Enterprise Edition, SQL Server 2014 versione 12.00 e versioni successive	MySQL 8.0.28 e versioni successe	Enterp e Edition e Standa Edition 2 (SE2)	Tutte le versioni di PostgreSQL 16 e 15, 14.5 e successive 14 versioni, 13.7 e successive 13 versioni e 13.4
db.x2iedn .2xlarge	Sì	Tutte le versioni di MariaDB 10.11, 10.6.7 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	Utilizza solo Standard Edition ed Enterprise Edition, SQL Server 2014 versione 12.00 e versioni successive	MySQL 8.0.28 e versioni successe	Enterp e Edition e Standa Edition 2 (SE2)	Tutte le versioni di PostgreSQL 16 e 15, 14.5 e successive 14 versioni, 13.7 e successive 13 versioni e 13.4

Classe di istanza	DI	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.x2iedn .xlarge	Sì	Tutte le versioni di MariaDB 10.11, 10.6.7 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	Utilizza solo Standard Edition ed Enterpris e Edition, SQL Server 2014 versione 12.00 e versioni successive	MySQL 8.0.28 e versioni success e	Enterp e Edition e Standa Edition 2 (SE2)	Tutte le versioni di PostgreSQL 16 e 15, 14.5 e successive 14 versioni, 13.7 e successive 13 versioni e 13.4

db.x2iezn: classi di istanza ottimizzate per la memoria basate su processori Intel Xeon scalabili di seconda generazione

Classe di istanza	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.x2iezn .8xlarge	No	No	No	No	Solo Enterprise Edition	No
db.x2iezn .6xlarge	No	No	No	No	Solo Enterprise Edition	No
db.x2iezn .4xlarge	No	No	No	No	Enterprise Edition e Standard Edition 2 (SE2)	No
db.x2iezn .2xlarge	No	No	No	No	Enterprise Edition e Standard Edition 2 (SE2)	No

db.x1e: classi di istanze ottimizzate per la memoria

Classe di istanza	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.x1e.32xlarge	No	No	Sì	No	Sì	No
db.x1e.16xlarge	No	No	Sì	No	Sì	No
db.x1e.8xlarge	No	No	Sì	No	Sì	No
db.x1e.4xlarge	No	No	Sì	No	Sì	No
db.x1e.2xlarge	No	No	Sì	No	Sì	No
db.x1e.xlarge	No	No	Sì	No	Sì	No

db.x1: classi di istanze ottimizzate per la memoria

Classe di istanza	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.x1.32xlarge	No	No	Sì	No	Sì	No
db.x1.16xlarge	No	No	Sì	No	Sì	No

db.r7g — classi di istanze ottimizzate per la memoria basate su processori Graviton3 AWS

Classe di istanza	DI	MariaDB	Micros SQL Server	MySQL	Ora	PostgreSQL
db.r7g.1 xlarge	N	Versioni MariaDB 10.11, 10.6.10 e versioni successiv e 10.6, 10.5.17 e versioni 10.5 successive e 10.4.26 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successive	No	Tutte le versioni di PostgreSQL 16 e 15, 14.5 e successive 14 versioni e 13.4 e versioni successive 13
db.r7g.1 xlarge	N	Versioni MariaDB 10.11, 10.6.10 e versioni successiv e 10.6, 10.5.17 e versioni 10.5 successive e 10.4.26 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successive	No	Tutte le versioni di PostgreSQL 16 e 15, 14.5 e successive 14 versioni e 13.4 e versioni successive 13
db.r7g.8 large	N	Versioni MariaDB 10.11, 10.6.10 e versioni successiv e 10.6, 10.5.17 e versioni 10.5 successive e 10.4.26 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successive	No	Tutte le versioni di PostgreSQL 16 e 15, 14.5 e successive 14 versioni e 13.4 e versioni successive 13
db.r7g.4 large	N	Versioni MariaDB 10.11, 10.6.10 e versioni successiv e 10.6, 10.5.17 e versioni 10.5 successive e 10.4.26 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successive	No	Tutte le versioni di PostgreSQL 16 e 15, 14.5 e successive 14 versioni e 13.4 e versioni successive 13
db.r7g.2 large	N	Versioni MariaDB 10.11, 10.6.10 e versioni successiv e 10.6, 10.5.17 e versioni 10.5 successive e 10.4.26 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successive	No	Tutte le versioni di PostgreSQL 16 e 15, 14.5 e successive 14 versioni e 13.4 e versioni successive 13
db.r7g.xl arge	N	Versioni MariaDB 10.11, 10.6.10 e versioni successiv e 10.6, 10.5.17 e versioni	No	MySQL 8.0.28 e versioni successive	No	Tutte le versioni di PostgreSQL 16 e 15, 14.5 e successive

Classe di istanza	Db	MariaDB	Micros SQL Server	MySQL	Ora	PostgreSQL
		10.5 successive e 10.4.26 e versioni 10.4 successive				14 versioni e 13.4 e versioni successive 13
db.r7g.la rge	No	Versioni MariaDB 10.11, 10.6.10 e versioni successiv e 10.6, 10.5.17 e versioni 10.5 successive e 10.4.26 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successive	No	Tutte le versioni di PostgreSQL 16 e 15, 14.5 e successive 14 versioni e 13.4 e versioni successive 13

db.r6g — classi di istanze ottimizzate per la memoria alimentate da processori Graviton2 AWS

Classe di istanza	Db2	MariaDB	Micros SQL Server	MySQL	Ora	PostgreSQL
db.r6g.16 xlarge	No	Tutte le versioni di MariadB 10.11, 10.6, 10.5 e 10.4	No	MySQL 8.0.23 e versioni successive	No	Tutte le versioni di PostgreSQL 16, 15, 14 e 13; 12 versioni 12.7 e successive
db.r6g.12 xlarge	No	Tutte le versioni di MariadB 10.11, 10.6, 10.5 e 10.4	No	MySQL 8.0.23 e versioni successive	No	Tutte le versioni di PostgreSQL 16, 15, 14 e 13; 12 versioni 12.7 e successive
db.r6g.8x large	No	Tutte le versioni di MariadB 10.11, 10.6, 10.5 e 10.4	No	MySQL 8.0.23 e versioni successive	No	Tutte le versioni di PostgreSQL 16, 15, 14 e 13; 12 versioni 12.7 e successive
db.r6g.4x large	No	Tutte le versioni di MariadB 10.11, 10.6, 10.5 e 10.4	No	MySQL 8.0.23 e	No	Tutte le versioni di PostgreSQL 16, 15, 14

Classe di istanza	Db2	MariaDB	Micros SQL Server	MySQL	Ora	PostgreSQL
				versioni successive		e 13; 12 versioni 12.7 e successive
db.r6g.2x large	No	Tutte le versioni di MariadB 10.11, 10.6, 10.5 e 10.4	No	MySQL 8.0.23 e versioni successive	No	Tutte le versioni di PostgreSQL 16, 15, 14 e 13; 12 versioni 12.7 e successive
db.r6g.xl arge	No	Tutte le versioni di MariadB 10.11, 10.6, 10.5 e 10.4	No	MySQL 8.0.23 e versioni successive	No	Tutte le versioni di PostgreSQL 16, 15, 14 e 13; 12 versioni 12.7 e successive
db.r6g.la rge	No	Tutte le versioni di MariadB 10.11, 10.6, 10.5 e 10.4	No	MySQL 8.0.23 e versioni successive	No	Tutte le versioni di PostgreSQL 16, 15, 14 e 13; 12 versioni 12.7 e successive

db.r6gd — classi di istanze ottimizzate per la memoria alimentate da processori Graviton2 AWS

Classe di istanza	Db	MariaDB	Micros SQL Server	MySQL	Ora	PostgreSQL
db.r6gd.1 6xlarge	No	Versioni MariaDB 10.11, 10.6.7 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successiv e	No	Tutte le versioni di PostgreSQL 16 e 15, 14.5 e successive 14 versioni, 13.7 e successive 13 versioni e 13.4
db.r6gd.1 2xlarge	No	Versioni MariaDB 10.11, 10.6.7 e versioni successive	No	MySQL 8.0.28 e	No	Tutte le versioni di PostgreSQL 16 e 15,

Classe di istanza	DI	MariaDB	Micros SQL Server	MySQL	Ora	PostgreSQL
		10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive		versioni successiv e		14.5 e successive 14 versioni, 13.7 e successive 13 versioni e 13.4
db.r6gd.8 xlarge	N	Versioni MariaDB 10.11, 10.6.7 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successiv e	No	Tutte le versioni di PostgreSQL 16 e 15, 14.5 e successive 14 versioni, 13.7 e successive 13 versioni e 13.4
db.r6gd.4 xlarge	N	Versioni MariaDB 10.11, 10.6.7 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successiv e	No	Tutte le versioni di PostgreSQL 16 e 15, 14.5 e successive 14 versioni, 13.7 e successive 13 versioni e 13.4
db.r6gd.2 xlarge	N	Versioni MariaDB 10.11, 10.6.7 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successiv e	No	Tutte le versioni di PostgreSQL 16 e 15, 14.5 e successive 14 versioni, 13.7 e successive 13 versioni e 13.4
db.r6gd.x large	N	Versioni MariaDB 10.11, 10.6.7 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successiv e	No	Tutte le versioni di PostgreSQL 16 e 15, 14.5 e successive 14 versioni, 13.7 e successive 13 versioni e 13.4

Classe di istanza	Dl	MariaDB	Micros SQL Server	MySQL	Ora	PostgreSQL
db.r6gd.l arge	N	Versioni MariaDB 10.11, 10.6.7 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successiv e	No	Tutte le versioni di PostgreSQL 16 e 15, 14.5 e successive 14 versioni, 13.7 e successive 13 versioni e 13.4

db.r6id: classi di istanza ottimizzata per la memoria basate su processori Intel Xeon scalabili di terza generazione

Classe di istanza	Db	MariaDB	Micros SQL Server	MySQL	Ora	PostgreSQL
db.r6id.3 2xlarge	No	MariaDB 10.6.10 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successive	No	Tutte le versioni di PostgreSQL 16 e 15, 14.5 e successive 14 versioni e 13.7 e versioni successive 13
db.r6id.2 4xlarge	No	MariaDB 10.6.10 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successive	No	Tutte le versioni di PostgreSQL 16 e 15, 14.5 e successive 14 versioni e 13.7 e versioni successive 13
db.r6id.1 6xlarge	No	MariaDB 10.6.10 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successive	No	Tutte le versioni di PostgreSQL 16 e 15, 14.5 e successive 14 versioni e 13.7 e versioni successive 13

Classe di istanza	Db	MariaDB	Micros SQL Server	MySQL	Ora	PostgreSQL
db.r6id.12xlarge	No	MariaDB 10.6.10 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successive	No	Tutte le versioni di PostgreSQL 16 e 15, 14.5 e successive 14 versioni e 13.7 e versioni successive 13
db.r6id.8xlarge	No	MariaDB 10.6.10 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successive	No	Tutte le versioni di PostgreSQL 16 e 15, 14.5 e successive 14 versioni e 13.7 e versioni successive 13
db.r6id.4xlarge	No	MariaDB 10.6.10 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successive	No	Tutte le versioni di PostgreSQL 16 e 15, 14.5 e successive 14 versioni e 13.7 e versioni successive 13
db.r6id.2xlarge	No	MariaDB 10.6.10 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successive	No	Tutte le versioni di PostgreSQL 16 e 15, 14.5 e successive 14 versioni e 13.7 e versioni successive 13
db.r6id.xlarge	No	MariaDB 10.6.10 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successive	No	Tutte le versioni di PostgreSQL 16 e 15, 14.5 e successive 14 versioni e 13.7 e versioni successive 13

Classe di istanza	Db	MariaDB	Micros SQL Server	MySQL	Ora	PostgreSQL
db.r6id.l arge	No	MariaDB 10.6.10 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successive	No	Tutte le versioni di PostgreSQL 16 e 15, 14.5 e successive 14 versioni e 13.7 e versioni successive 13

db.r6idn: classi di istanza ottimizzata per la memoria basate su processori Intel Xeon scalabili di terza generazione

Classe di istanza	Db	MariaDB	Micros SQL Server	MySQL	Ora	PostgreSQL
db.r6idn. 32xlarge	Sì	MariaDB versione 10.6.8 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successiv e	No	Tutte le versioni di PostgreSQL 16 e 15, 14.5 e successive 14 versioni e 13.7 e versioni successive 13
db.r6idn. 24xlarge	Sì	MariaDB versione 10.6.8 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successiv e	No	Tutte le versioni di PostgreSQL 16 e 15, 14.5 e successive 14 versioni e 13.7 e versioni successive 13
db.r6idn. 16xlarge	Sì	MariaDB versione 10.6.8 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successiv e	No	Tutte le versioni di PostgreSQL 16 e 15, 14.5 e successive 14 versioni e 13.7 e versioni successive 13
db.r6idn. 12xlarge	Sì	MariaDB versione 10.6.8 e versioni successive 10.6,	No	MySQL 8.0.28 e	No	Tutte le versioni di PostgreSQL 16 e 15,

Classe di istanza	Db2	MariaDB	Micros SQL Server	MySQL	Ora	PostgreSQL
		10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive		versioni successiv e		14.5 e successive 14 versioni e 13.7 e versioni successive 13
db.r6idn. 8xlarge	Sì	MariaDB versione 10.6.8 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successiv e	No	Tutte le versioni di PostgreSQL 16 e 15, 14.5 e successive 14 versioni e 13.7 e versioni successive 13
db.r6idn. 4xlarge	Sì	MariaDB versione 10.6.8 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successiv e	No	Tutte le versioni di PostgreSQL 16 e 15, 14.5 e successive 14 versioni e 13.7 e versioni successive 13
db.r6idn. 2xlarge	Sì	MariaDB versione 10.6.8 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successiv e	No	Tutte le versioni di PostgreSQL 16 e 15, 14.5 e successive 14 versioni e 13.7 e versioni successive 13
db.r6idn. xlarge	Sì	MariaDB versione 10.6.8 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successiv e	No	Tutte le versioni di PostgreSQL 16 e 15, 14.5 e successive 14 versioni e 13.7 e versioni successive 13

db.r6in: classi di istanza ottimizzata per la memoria basate su processori Intel Xeon scalabili di terza generazione

Classe di istanza	Db2	MariaDB	Micros SQL Server	MySQL	Ora	PostgreSQL
db.r6in.32xlarge	Sì	MariaDB versione 10.6.8 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successive	No	Tutte le versioni di PostgreSQL 16 e 15, 14.3 e successive 14 versioni, 13.7 e successive 13 versioni, 12.11 e successive 12 versioni e 11.16 e versioni 11 successive
db.r6in.24xlarge	Sì	MariaDB versione 10.6.8 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successive	No	Tutte le versioni di PostgreSQL 16 e 15, 14.3 e successive 14 versioni, 13.7 e successive 13 versioni, 12.11 e successive 12 versioni e 11.16 e versioni 11 successive
db.r6in.16xlarge	Sì	MariaDB versione 10.6.8 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successive	No	Tutte le versioni di PostgreSQL 16 e 15, 14.3 e successive 14 versioni, 13.7 e successive 13 versioni, 12.11 e successive 12 versioni e 11.16 e versioni 11 successive
db.r6in.12xlarge	Sì	MariaDB versione 10.6.8 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successive	No	Tutte le versioni di PostgreSQL 16 e 15, 14.3 e successive 14 versioni, 13.7 e successive 13 versioni, 12.11 e successive 12 versioni e 11.16 e versioni 11 successive

Classe di istanza	Db	MariaDB	Micros SQL Server	MySQL	Ora	PostgreSQL
db.r6in.8xlarge	Sì	MariaDB versione 10.6.8 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successive	No	Tutte le versioni di PostgreSQL 16 e 15, 14.3 e successive 14 versioni, 13.7 e successive 13 versioni, 12.11 e successive 12 versioni e 11.16 e versioni 11 successive
db.r6in.4xlarge	Sì	MariaDB versione 10.6.8 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successive	No	Tutte le versioni di PostgreSQL 16 e 15, 14.3 e successive 14 versioni, 13.7 e successive 13 versioni, 12.11 e successive 12 versioni e 11.16 e versioni 11 successive
db.r6in.2xlarge	Sì	MariaDB versione 10.6.8 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successive	No	Tutte le versioni di PostgreSQL 16 e 15, 14.3 e successive 14 versioni, 13.7 e successive 13 versioni, 12.11 e successive 12 versioni e 11.16 e versioni 11 successive
db.r6in.xlarge	Sì	MariaDB versione 10.6.8 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successive	No	Tutte le versioni di PostgreSQL 16 e 15, 14.3 e successive 14 versioni, 13.7 e successive 13 versioni, 12.11 e successive 12 versioni e 11.16 e versioni 11 successive

Classe di istanza	Db:	MariaDB	Micros SQL Server	MySQL	Ora	PostgreSQL
db.r6in.l arge	Sì	MariaDB versione 10.6.8 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successiv e	No	Tutte le versioni di PostgreSQL 16 e 15, 14.3 e successive 14 versioni, 13.7 e successive 13 versioni, 12.11 e successive 12 versioni e 11.16 e versioni 11 successive

db.r6i: classi di istanze ottimizzate per la memoria

Classe di istanza	Dl	MariaDB	Micros SQL Serve	MySQL	Ora	PostgreSQL
db.r6i.3: xlarge	Sì	Versioni MariaDB 10.11, 10.6.7 e versioni successiv e 10.6, 10.5.15 e versioni 10.5 successive e 10.4.24 e versioni 10.4 successive	Sì	MySQL 8.0.28 e versioni successiv e	Sì	Tutte le versioni di PostgreSQL 16, 15 e 14, 13.4 e successive 13 versioni, 12.8 e successiv e 12 versioni, 11.13 e successive 11 versioni e 10.21 e versioni successive 10
db.r6i.2- xlarge	Sì	Versioni MariaDB 10.11, 10.6.7 e versioni successiv e 10.6, 10.5.15 e versioni 10.5 successive e 10.4.24 e versioni 10.4 successive	Sì	MySQL 8.0.28 e versioni successiv e	Sì	Tutte le versioni di PostgreSQL 16, 15 e 14, 13.4 e successive 13 versioni, 12.8 e successiv e 12 versioni, 11.13 e successive 11 versioni e 10.21 e versioni successive 10

Classe di istanza	DI	MariaDB	Micros SQL Serve	MySQL	Ora	PostgreSQL
db.r6i.1(xlarge	Sì	Versioni MariaDB 10.11, 10.6.7 e versioni successiv e 10.6, 10.5.15 e versioni 10.5 successive e 10.4.24 e versioni 10.4 successive	Sì	MySQL 8.0.28 e versioni successiv e	Sì	Tutte le versioni di PostgreSQL 16, 15 e 14, 13.4 e successive 13 versioni, 12.8 e successiv e 12 versioni, 11.13 e successive 11 versioni e 10.21 e versioni successive 10
db.r6i.1: xlarge	Sì	Versioni MariaDB 10.11, 10.6.7 e versioni successiv e 10.6, 10.5.15 e versioni 10.5 successive e 10.4.24 e versioni 10.4 successive	Sì	MySQL 8.0.28 e versioni successiv e	Sì	Tutte le versioni di PostgreSQL 16, 15 e 14, 13.4 e successive 13 versioni, 12.8 e successiv e 12 versioni, 11.13 e successive 11 versioni e 10.21 e versioni successive 10
db.r6i.8: large	Sì	Versioni MariaDB 10.11, 10.6.7 e versioni successiv e 10.6, 10.5.15 e versioni 10.5 successive e 10.4.24 e versioni 10.4 successive	Sì	MySQL 8.0.28 e versioni successiv e	Sì	Tutte le versioni di PostgreSQL 16, 15 e 14, 13.4 e successive 13 versioni, 12.8 e successiv e 12 versioni, 11.13 e successive 11 versioni e 10.21 e versioni successive 10

Classe di istanza	DI	MariaDB	Micros SQL Serve	MySQL	Ora	PostgreSQL
db.r6i.4x large	Sì	Versioni MariaDB 10.11, 10.6.7 e versioni successive 10.6, 10.5.15 e versioni 10.5 successive e 10.4.24 e versioni 10.4 successive	Sì	MySQL 8.0.28 e versioni successive	Sì	Tutte le versioni di PostgreSQL 16, 15 e 14, 13.4 e successive 13 versioni, 12.8 e successive 12 versioni, 11.13 e successive 11 versioni e 10.21 e versioni successive 10
db.r6i.2x large	Sì	Versioni MariaDB 10.11, 10.6.7 e versioni successive 10.6, 10.5.15 e versioni 10.5 successive e 10.4.24 e versioni 10.4 successive	Sì	MySQL 8.0.28 e versioni successive	Sì	Tutte le versioni di PostgreSQL 16, 15 e 14, 13.4 e successive 13 versioni, 12.8 e successive 12 versioni, 11.13 e successive 11 versioni e 10.21 e versioni successive 10
db.r6i.xl arge	Sì	Versioni MariaDB 10.11, 10.6.7 e versioni successive 10.6, 10.5.15 e versioni 10.5 successive e 10.4.24 e versioni 10.4 successive	Sì	MySQL 8.0.28 e versioni successive	Sì	Tutte le versioni di PostgreSQL 16, 15 e 14, 13.4 e successive 13 versioni, 12.8 e successive 12 versioni, 11.13 e successive 11 versioni e 10.21 e versioni successive 10

Classe di istanza	DI	MariaDB	Micros SQL Serve	MySQL	Ora	PostgreSQL
db.r6i.la rge	Sì	Versioni MariaDB 10.11, 10.6.7 e versioni successiv e 10.6, 10.5.15 e versioni 10.5 successive e 10.4.24 e versioni 10.4 successive	Sì	MySQL 8.0.28 e versioni successiv e	Sì	Tutte le versioni di PostgreSQL 16, 15 e 14, 13.4 e successive 13 versioni, 12.8 e successiv e 12 versioni, 11.13 e successive 11 versioni e 10.21 e versioni successive 10

db.r5d: classi di istanze ottimizzate per la memoria

Classe di istanza	DI	MariaDB	Micros SQL Serve	MySQL	Ora	PostgreSQL
db.r5d.2 xlarge	N	Versioni MariaDB 10.11, 10.6.7 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	Sì	MySQL 8.0.28 e versioni success e	Sì	Tutte le versioni di PostgreSQL 16 e 15, 14.5 e successive 14 versioni, 13.7 e successive 13 versioni e 13.4
db.r5d.1 xlarge	N	Versioni MariaDB 10.11, 10.6.7 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	Sì	MySQL 8.0.28 e versioni success e	Sì	Tutte le versioni di PostgreSQL 16 e 15, 14.5 e successive 14 versioni, 13.7 e successive 13 versioni e 13.4
db.r5d.1 xlarge	N	Versioni MariaDB 10.11, 10.6.7 e versioni successive 10.6, 10.5.16 e versioni 10.5	Sì	MySQL 8.0.28 e versioni	Sì	Tutte le versioni di PostgreSQL 16 e 15, 14.5 e successive 14 versioni, 13.7

Classe di istanza	Dl	MariaDB	Micros SQL Serve	MySQL	Ora	PostgreSQL
		successive e 10.4.25 e versioni 10.4 successive		success e		e successive 13 versioni e 13.4
db.r5d.8 large	N	Versioni MariaDB 10.11, 10.6.7 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	Sì	MySQL 8.0.28 e versioni success e	Sì	Tutte le versioni di PostgreSQL 16 e 15, 14.5 e successive 14 versioni, 13.7 e successive 13 versioni e 13.4
db.r5d.4 large	N	Versioni MariaDB 10.11, 10.6.7 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	Sì	MySQL 8.0.28 e versioni success e	Sì	Tutte le versioni di PostgreSQL 16 e 15, 14.5 e successive 14 versioni, 13.7 e successive 13 versioni e 13.4
db.r5d.2 large	N	Versioni MariaDB 10.11, 10.6.7 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	Sì	MySQL 8.0.28 e versioni success e	Sì	Tutte le versioni di PostgreSQL 16 e 15, 14.5 e successive 14 versioni, 13.7 e successive 13 versioni e 13.4
db.r5d.x arge	N	Versioni MariaDB 10.11, 10.6.7 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	Sì	MySQL 8.0.28 e versioni success e	Sì	Tutte le versioni di PostgreSQL 16 e 15, 14.5 e successive 14 versioni, 13.7 e successive 13 versioni e 13.4

Classe di istanza	Di	MariaDB	Micros SQL Serve	MySQL	Ora	PostgreSQL
db.r5d.la rge	N	Versioni MariaDB 10.11, 10.6.7 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	Sì	MySQL 8.0.28 e versioni success e	Sì	Tutte le versioni di PostgreSQL 16 e 15, 14.5 e successive 14 versioni, 13.7 e successive 13 versioni e 13.4

db.r5b: classi di istanze ottimizzate per la memoria preconfigurate per uso intensivo di memoria, archiviazione e I/O

Classe di istanza	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r5b.8xlarge.tpc2.mem3x	No	No	No	No	Sì	No
db.r5b.6xlarge.tpc2.mem4x	No	No	No	No	Sì	No
db.r5b.4xlarge.tpc2.mem4x	No	No	No	No	Sì	No
db.r5b.4xlarge.tpc2.mem3x	No	No	No	No	Sì	No
db.r5b.4xlarge.tpc2.mem2x	No	No	No	No	Sì	No

Classe di istanza	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r5b.2xlarge.tpc2.mem8x	No	No	No	No	Sì	No
db.r5b.2xlarge.tpc2.mem4x	No	No	No	No	Sì	No
db.r5b.2xlarge.tpc1.mem2x	No	No	No	No	Sì	No
db.r5b.xlarge.tpc2.mem4x	No	No	No	No	Sì	No
db.r5b.xlarge.tpc2.mem2x	No	No	No	No	Sì	No
db.r5b.large.tpc1.mem2x	No	No	No	No	Sì	No

db.r5b: classi di istanze ottimizzate per la memoria

Classe di istanza	Db2	MariaDB	Micros SQL Server	MySQL	Orac	PostgreSQL
db.r5b.24xlarge	No	Versioni MariaDB 10.11, 10.6.5 e versioni successive 10.6, 10.5.12 e versioni 10.5	Sì	MySQL 8.0.25 e versioni	Sì	Tutte le versioni di PostgreSQL 16, 15, 14

Classe di istanza	Db2	MariaDB	Micros SQL Server	MySQL	Orac	PostgreSQL
		successive, 10.4.24 e versioni 10.4 successive e 10.3.34 e versioni 10.3 successive		successiv e		e 13; 12 versioni 12.7 e successive
db.r5b.16 xlarge	No	Versioni MariaDB 10.11, 10.6.5 e versioni successive 10.6, 10.5.12 e versioni 10.5 successive, 10.4.24 e versioni 10.4 successive e 10.3.34 e versioni 10.3 successive	Sì	MySQL 8.0.25 e versioni successiv e	Sì	Tutte le versioni di PostgreSQL 16, 15, 14 e 13; 12 versioni 12.7 e successive
db.r5b.12 xlarge	No	Versioni MariaDB 10.11, 10.6.5 e versioni successive 10.6, 10.5.12 e versioni 10.5 successive, 10.4.24 e versioni 10.4 successive e 10.3.34 e versioni 10.3 successive	Sì	MySQL 8.0.25 e versioni successiv e	Sì	Tutte le versioni di PostgreSQL 16, 15, 14 e 13; 12 versioni 12.7 e successive
db.r5b.8x large	No	Versioni MariaDB 10.11, 10.6.5 e versioni successive 10.6, 10.5.12 e versioni 10.5 successive, 10.4.24 e versioni 10.4 successive e 10.3.34 e versioni 10.3 successive	Sì	MySQL 8.0.25 e versioni successiv e	>Sì	Tutte le versioni di PostgreSQL 16, 15, 14 e 13; 12 versioni 12.7 e successive
db.r5b.4x large	No	Versioni MariaDB 10.11, 10.6.5 e versioni successive 10.6, 10.5.12 e versioni 10.5 successive, 10.4.24 e versioni 10.4 successive e 10.3.34 e versioni 10.3 successive	Sì	MySQL 8.0.25 e versioni successiv e	Sì	Tutte le versioni di PostgreSQL 16, 15, 14 e 13; 12 versioni 12.7 e successive

Classe di istanza	Db2	MariaDB	Micros SQL Server	MySQL	Orac	PostgreSQL
db.r5b.2xlarge	No	Versioni MariaDB 10.11, 10.6.5 e versioni successive 10.6, 10.5.12 e versioni 10.5 successive, 10.4.24 e versioni 10.4 successive e 10.3.34 e versioni 10.3 successive	Sì	MySQL 8.0.25 e versioni successive	Sì	Tutte le versioni di PostgreSQL 16, 15, 14 e 13; 12 versioni 12.7 e successive
db.r5b.xlarge	No	Versioni MariaDB 10.11, 10.6.5 e versioni successive 10.6, 10.5.12 e versioni 10.5 successive, 10.4.24 e versioni 10.4 successive e 10.3.34 e versioni 10.3 successive	Sì	MySQL 8.0.25 e versioni successive	Sì	Tutte le versioni di PostgreSQL 16, 15, 14 e 13; 12 versioni 12.7 e successive
db.r5b.large	No	Versioni MariaDB 10.11, 10.6.5 e versioni successive 10.6, 10.5.12 e versioni 10.5 successive, 10.4.24 e versioni 10.4 successive e 10.3.34 e versioni 10.3 successive	Sì	MySQL 8.0.25 e versioni successive	Sì	Tutte le versioni di PostgreSQL 16, 15, 14 e 13; 12 versioni 12.7 e successive

db.r5: classi di istanze ottimizzate per la memoria preconfigurate per uso intensivo di memoria, archiviazione e I/O

Classe di istanza	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r5.12xlarge.tpc2.mem2x	No	No	No	No	Sì	No

Classe di istanza	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r5.8xlarge.tpc2.mem3x	No	No	No	No	Sì	No
db.r5.6xlarge.tpc2.mem4x	No	No	No	No	Sì	No
db.r5.4xlarge.tpc2.mem4x	No	No	No	No	Sì	No
db.r5.4xlarge.tpc2.mem3x	No	No	No	No	Sì	No
db.r5.4xlarge.tpc2.mem2x	No	No	No	No	Sì	No
db.r5.2xlarge.tpc2.mem8x	No	No	No	No	Sì	No
db.r5.2xlarge.tpc2.mem4x	No	No	No	No	Sì	No
db.r5.2xlarge.tpc1.mem2x	No	No	No	No	Sì	No
db.r5.xlarge.tpc2.mem4x	No	No	No	No	Sì	No
db.r5.xlarge.tpc2.mem2x	No	No	No	No	Sì	No
db.r5.large.tpc1.mem2x	No	No	No	No	Sì	No

db.r5: classi di istanze ottimizzate per la memoria

Classe di istanza	Db2	MariaD	Microsoft SQL Server	MySC	Oracle	PostgreSQL
db.r5.24xlarge	No	Sì	Sì	Sì	Sì	Tutte le versioni di PostgreSQL 16, 15, 14, 13, 12 e 11; 10 versioni 10.17 e successive; e 9.6.22 e versioni successive 9
db.r5.16xlarge	No	Sì	Sì	Sì	Sì	Tutte le versioni di PostgreSQL 16, 15, 14, 13, 12 e 11; 10 versioni 10.17 e successive; e 9.6.22 e versioni successive 9
db.r5.12xlarge	No	Sì	Sì	Sì	Sì	Tutte le versioni di PostgreSQL 16, 15, 14, 13, 12 e 11; 10 versioni 10.17 e successive; e 9.6.22 e versioni successive 9
db.r5.8xlarge	No	Sì	Sì	Sì	Sì	Tutte le versioni di PostgreSQL 16, 15, 14, 13, 12 e 11; 10 versioni 10.17 e successive; e 9.6.22 e versioni successive 9
db.r5.4xlarge	No	Sì	Sì	Sì	Sì	Tutte le versioni di PostgreSQL 16, 15, 14, 13, 12 e 11; 10 versioni 10.17 e successive; e 9.6.22 e versioni successive 9
db.r5.2xlarge	No	Sì	Sì	Sì	Sì	Tutte le versioni di PostgreSQL 16, 15, 14, 13, 12 e 11; 10 versioni 10.17 e successive; e 9.6.22 e versioni successive 9
db.r5.xlarge	No	Sì	Sì	Sì	Sì	Tutte le versioni di PostgreSQL 16, 15, 14, 13, 12 e 11; 10 versioni 10.17 e successive; e 9.6.22 e versioni successive 9
db.r5.large	No	Sì	Sì	Sì	Sì	Tutte le versioni di PostgreSQL 16, 15, 14, 13, 12 e 11; 10 versioni 10.17 e successive; e 9.6.22 e versioni successive 9

db.r4: classi di istanze ottimizzate per la memoria

Classe di istanza	Db2	MariaDB	Microsoft SQL Server	MySQL	Oracle	PostgreSQL
db.r4.16xlarge	No	Deprecated	Sì	Obsoleta	Obsolet	Inferiore a PostgreSQL 13
db.r4.8xlarge	No	Deprecated	Sì	Obsoleta	Obsolet	Inferiore a PostgreSQL 13
db.r4.4xlarge	No	Deprecated	Sì	Obsoleta	Obsolet	Inferiore a PostgreSQL 13
db.r4.2xlarge	No	Deprecated	Sì	Obsoleta	Obsolet	Inferiore a PostgreSQL 13
db.r4.xlarge	No	Deprecated	Sì	Obsoleta	Obsolet	Inferiore a PostgreSQL 13
db.r4.large	No	Deprecated	Sì	Obsoleta	Obsolet	Inferiore a PostgreSQL 13

db.r3: classi di istanze ottimizzate per la memoria

Classe di istanza	Db2	MariaDB	Microsoft SQL Server	MySQ	Oracle	PostgreSQL
db.r3.8xlarge	No	Tutte le versioni di MariadB 10.6, 10.5, 10.4 e 10.3	Sì	Sì	Obsoleta	Deprecated

Classe di istanza	Db2	MariaDB	Microsoft SQL Server	MySQ	Oracle	PostgreSQL
db.r3.4xlarge	No	Tutte le versioni di MariadB 10.6, 10.5, 10.4 e 10.3	Sì	Sì	Obsoleta	Deprecated
db.r3.2xlarge	No	Tutte le versioni di MariadB 10.6, 10.5, 10.4 e 10.3	Sì	Sì	Obsoleta	Deprecated
db.r3.xlarge	No	Tutte le versioni di MariadB 10.6, 10.5, 10.4 e 10.3	Sì	Sì	Obsoleta	Deprecated
db.r3.large	No	Tutte le versioni di MariadB 10.6, 10.5, 10.4 e 10.3	Sì	Sì	Obsoleta	Deprecated

Motori DB supportati per classi di istanze ottimizzate per il calcolo

Le tabelle seguenti mostrano i database e le versioni dei database supportati per le classi di istanze ottimizzate per il calcolo.

db.c6gd — classi di istanze ottimizzate per il calcolo (solo per implementazioni di cluster DB Multi-AZ)

Classe di istanza	Db2	Maria	Microsoft SQL Server	MySQL	Orac	PostgreSQL
db.c6gd.16xlarge	No	No	No	MySQL 8.0.28 e versioni successive	No	Tutte le versioni di PostgreSQL 16 e 15; 14.5 e successive 14 versioni; 13.4 e 13.7 e versioni successive 13
db.c6gd.12xlarge	No	No	No	MySQL 8.0.28 e versioni successive	No	Tutte le versioni di PostgreSQL 16 e 15; 14.5 e successive 14 versioni; 13.4 e 13.7 e versioni successive 13

Classe di istanza	Db2	Maria	Microsoft SQL Server	MySQL	Orac	PostgreSQL
db.c6gd.8xlarge	No	No	No	MySQL 8.0.28 e versioni successive	No	Tutte le versioni di PostgreSQL 16 e 15; 14.5 e successive 14 versioni; 13.4 e 13.7 e versioni successive 13
db.c6gd.4xlarge	No	No	No	MySQL 8.0.28 e versioni successive	No	Tutte le versioni di PostgreSQL 16 e 15; 14.5 e successive 14 versioni; 13.4 e 13.7 e versioni successive 13
db.c6gd.2xlarge	No	No	No	MySQL 8.0.28 e versioni successive	No	Tutte le versioni di PostgreSQL 16 e 15; 14.5 e successive 14 versioni; 13.4 e 13.7 e versioni successive 13
db.c6gd.xlarge	No	No	No	MySQL 8.0.28 e versioni successive	No	Tutte le versioni di PostgreSQL 16 e 15; 14.5 e successive 14 versioni; 13.4 e 13.7 e versioni successive 13
db.c6gd.large	No	No	No	MySQL 8.0.28 e versioni successive	No	Tutte le versioni di PostgreSQL 16 e 15; 14.5 e successive 14 versioni; 13.4 e 13.7 e versioni successive 13
db.c6gd.medium	No	No	No	MySQL 8.0.28 e versioni successive	No	Tutte le versioni di PostgreSQL 16 e 15; 14.5 e successive 14 versioni; 13.4 e 13.7 e versioni successive 13

Motori DB supportati per classi di istanze con prestazioni espandibili

Le tabelle seguenti mostrano i database e le versioni dei database supportati per le classi di istanze burstable-performance.

db.t4g — classi di istanze burstable-performance basate su processori Graviton2 AWS

Classe di istanza	Db2	MariaDB	Microsoft SQL Server	MySQL	Orac	PostgreSQL
db.t4g — classi di istanze a prestazioni burstabili alimentate da processori AWS Graviton2						
db.t4g.2xlarge	No	Tutte le versioni di MariadB 10.11, 10.6, 10.5 e 10.4	No	MySQL 8.0.25 e versioni successive	No	Tutte le versioni di PostgreSQL 16, 15, 14 e 13; 12 versioni 12.7 e successive
db.t4g.xlarge	No	Tutte le versioni di MariadB 10.11, 10.6, 10.5 e 10.4	No	MySQL 8.0.25 e versioni successive	No	Tutte le versioni di PostgreSQL 16, 15, 14 e 13; 12 versioni 12.7 e successive
db.t4g.large	No	Tutte le versioni di MariadB 10.11, 10.6, 10.5 e 10.4	No	MySQL 8.0.25 e versioni successive	No	Tutte le versioni di PostgreSQL 16, 15, 14 e 13; 12 versioni 12.7 e successive
db.t4g.medium	No	Tutte le versioni di MariadB 10.11, 10.6, 10.5 e 10.4	No	MySQL 8.0.25 e versioni successive	No	Tutte le versioni di PostgreSQL 16, 15, 14 e 13; 12 versioni 12.7 e successive

Classe di istanza	Db2	MariaDB	Microsoft SQL Server	MySQL	Orac	PostgreSQL
db.t4g.small	No	Tutte le versioni di MariadB 10.11, 10.6, 10.5 e 10.4	No	MySQL 8.0.25 e versioni successive	No	Tutte le versioni di PostgreSQL 16, 15, 14 e 13; 12 versioni 12.7 e successive
db.t4g.micro	No	Tutte le versioni di MariadB 10.11, 10.6, 10.5 e 10.4	No	MySQL 8.0.25 e versioni successive	No	Tutte le versioni di PostgreSQL 16, 15, 14 e 13; 12 versioni 12.7 e successive

db.t3: classi di istanza a prestazioni espandibili

Classe di istanza	Db2	Maria	Microsoft SQL Server	MySC	Oracle	PostgreSQL
db.t3.2xlarge	Sì	Sì	Sì	Sì	Sì	Tutte le versioni di PostgreSQL 16, 15, 14, 13, 12, 11 e 10; e 9.6.22 e versioni successive 9
db.t3.xlarge	Sì	Sì	Sì	Sì	Sì	Tutte le versioni di PostgreSQL 16, 15, 14, 13, 12, 11 e 10; e 9.6.22 e versioni successive 9
db.t3.large	Sì	Sì	Sì	Sì	Sì	Tutte le versioni di PostgreSQL 16, 15, 14, 13, 12, 11 e 10; e 9.6.22 e versioni successive 9

Classe di istanza	Db2	Maria	Microsoft SQL Server	MySC	Oracle	PostgreSQL
db.t3.mec ium	Sì	Sì	Sì	Sì	Sì	Tutte le versioni di PostgreSQL 16, 15, 14, 13, 12, 11 e 10; e 9.6.22 e versioni successive 9
db.t3.sma ll	Sì	Sì	Sì	Sì	Sì	Tutte le versioni di PostgreSQL 16, 15, 14, 13, 12, 11 e 10; e 9.6.22 e versioni successive 9
db.t3.mic ro	No	Sì	No	Sì	Solo su Oracle Database 12c Release 1 (12.1.0.2), che è obsoleta	Tutte le versioni di PostgreSQL 16, 15, 14, 13, 12, 11 e 10; e 9.6.22 e versioni successive 9

db.t2: classi di istanza a prestazioni espandibili

Classe di istanza	Db2	MariaDB	Microsc SQL Server	MySQL	Oracle	PostgreSQL
db.t2.2xl arge	No	Deprecated	No	Deprecate d	Obsolet	Inferiore a PostgreSQL 13
db.t2.xla rge	No	Deprecated	No	Deprecate d	Obsolet	Inferiore a PostgreSQL 13
db.t2.lar ge	No	Deprecated	Sì	Obsoleta	Obsolet	Inferiore a PostgreSQL 13
db.t2.med ium	No	Deprecated	Sì	Obsoleta	Obsolet	Inferiore a PostgreSQL 13

Classe di istanza	Db2	MariaDB	Microsc SQL Server	MySQL	Oracle	PostgreSQL
db.t2.small	No	Deprecated	Sì	Obsoleta	Obsolet	Inferiore a PostgreSQL 13
db.t2.micro	No	Deprecated	Sì	Obsoleta	Obsolet	Inferiore a PostgreSQL 13

Motori DB supportati per le classi di istanze Optimized Reads

Le tabelle seguenti mostrano i database e le versioni dei database supportati per le classi di istanze Optimized Reads.

db.r6gd — classi di istanze ottimizzate per la memoria che supportano Optimized Reads e sono alimentate da processori Graviton2 AWS

Classe di istanza	Dl	MariaDB	Micros SQL Server	MySQL	Ora	PostgreSQL
db.r6gd.16xlarge	N	Versioni MariaDB 10.11, 10.6.7 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successive	No	Tutte le versioni di PostgreSQL 16 e 15, 14.5 e successive 14 versioni, 13.7 e successive 13 versioni e 13.4
db.r6gd.12xlarge	N	Versioni MariaDB 10.11, 10.6.7 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successive	No	Tutte le versioni di PostgreSQL 16 e 15, 14.5 e successive 14 versioni, 13.7 e successive 13 versioni e 13.4
db.r6gd.8xlarge	N	Versioni MariaDB 10.11, 10.6.7 e versioni successive	No	MySQL 8.0.28 e	No	Tutte le versioni di PostgreSQL 16 e 15,

Classe di istanza	DI	MariaDB	Micros SQL Server	MySQL	Ora	PostgreSQL
		10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive		versioni successiv e		14.5 e successive 14 versioni, 13.7 e successive 13 versioni e 13.4
db.r6gd.4 xlarge	N	Versioni MariaDB 10.11, 10.6.7 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successiv e	No	Tutte le versioni di PostgreSQL 16 e 15, 14.5 e successive 14 versioni, 13.7 e successive 13 versioni e 13.4
db.r6gd.2 xlarge	N	Versioni MariaDB 10.11, 10.6.7 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successiv e	No	Tutte le versioni di PostgreSQL 16 e 15, 14.5 e successive 14 versioni, 13.7 e successive 13 versioni e 13.4
db.r6gd.x large	N	Versioni MariaDB 10.11, 10.6.7 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successiv e	No	Tutte le versioni di PostgreSQL 16 e 15, 14.5 e successive 14 versioni, 13.7 e successive 13 versioni e 13.4
db.r6gd.l arge	N	Versioni MariaDB 10.11, 10.6.7 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successiv e	No	Tutte le versioni di PostgreSQL 16 e 15, 14.5 e successive 14 versioni, 13.7 e successive 13 versioni e 13.4

db.r6id: classi di istanze ottimizzate per la memoria che supportano Optimized Reads e sono alimentate da processori scalabili Intel Xeon di terza generazione

Classe di istanza	Db	MariaDB	Micros SQL Server	MySQL	Ora	PostgreSQL
db.r6id.32xlarge	No	MariaDB 10.6.10 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successive	No	Tutte le versioni di PostgreSQL 16 e 15, 14.5 e successive 14 versioni e 13.7 e versioni successive 13
db.r6id.24xlarge	No	MariaDB 10.6.10 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successive	No	Tutte le versioni di PostgreSQL 16 e 15, 14.5 e successive 14 versioni e 13.7 e versioni successive 13
db.r6id.16xlarge	No	MariaDB 10.6.10 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successive	No	Tutte le versioni di PostgreSQL 16 e 15, 14.5 e successive 14 versioni e 13.7 e versioni successive 13
db.r6id.12xlarge	No	MariaDB 10.6.10 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successive	No	Tutte le versioni di PostgreSQL 16 e 15, 14.5 e successive 14 versioni e 13.7 e versioni successive 13
db.r6id.8xlarge	No	MariaDB 10.6.10 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successive	No	Tutte le versioni di PostgreSQL 16 e 15, 14.5 e successive 14 versioni e 13.7 e versioni successive 13

Classe di istanza	Db	MariaDB	Micros SQL Server	MySQL	Ora	PostgreSQL
db.r6id.4xlarge	No	MariaDB 10.6.10 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successive	No	Tutte le versioni di PostgreSQL 16 e 15, 14.5 e successive 14 versioni e 13.7 e versioni successive 13
db.r6id.2xlarge	No	MariaDB 10.6.10 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successive	No	Tutte le versioni di PostgreSQL 16 e 15, 14.5 e successive 14 versioni e 13.7 e versioni successive 13
db.r6id.xlarge	No	MariaDB 10.6.10 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successive	No	Tutte le versioni di PostgreSQL 16 e 15, 14.5 e successive 14 versioni e 13.7 e versioni successive 13
db.r6id.large	No	MariaDB 10.6.10 e versioni successive 10.6, 10.5.16 e versioni 10.5 successive e 10.4.25 e versioni 10.4 successive	No	MySQL 8.0.28 e versioni successive	No	Tutte le versioni di PostgreSQL 16 e 15, 14.5 e successive 14 versioni e 13.7 e versioni successive 13

Determinazione del supporto delle classi di istanze DB in Regioni AWS

Per determinare le classi di istanza database supportate da ciascun motore di database in una specifica Regione AWS, puoi adottare uno di diversi approcci. Puoi utilizzare la AWS Management Console pagina dei prezzi di Amazon RDS o il comando describe-orderable-db-instance-options per AWS Command Line Interface ()AWS CLI.

> **Note**
>
> Quando esegui operazioni con AWS Management Console, mostra automaticamente le classi di istanze DB supportate per uno specifico motore DB, una versione del motore DB e. Regione AWS Esempi di operazioni che è possibile eseguire includono la creazione e la modifica di un'istanza database.

Indice

Utilizzo della pagina dei prezzi di Amazon RDS per determinare il supporto della classe di istanze DB in Regioni AWS

Puoi utilizzare la pagina Prezzi di Amazon RDS per determinare le classi di istanza database supportate da ciascun motore DB in una Regione AWS specifica.

Per utilizzare la pagina dei prezzi per determinare le classi di istanza database supportate da ciascun modulo di gestione in una regione

1. Vai a Prezzi di Amazon RDS.
2. Nella sezione Calcolatore dei prezzi AWS per Amazon RDS, scegli Crea subito il preventivo personalizzato.
3. In Scegli una regione, scegli una Regione AWS.
4. In Trova un servizio, inserisci **`Amazon RDS`**.
5. Scegli Configura per l'opzione di configurazione e il motore del database.
6. Utilizza la sezione dedicata alle istanze compatibili per visualizzare le classi di istanze database supportate.

7. (Facoltativo) Scegli altre opzioni nella calcolatrice, quindi scegli Salva e visualizza riepilogo o Salva e aggiungi servizio.

Utilizzo di AWS CLI per determinare il supporto delle classi di istanze DB in Regioni AWS

È possibile utilizzare il AWS CLI per determinare quali classi di istanze DB sono supportate per motori DB specifici e versioni del motore DB in un Regione AWS. Nella tabella seguente vengono illustrati i valori validi del motore DB.

Nomi del motore	Valori del motore nei comandi CLI	Ulteriori informazioni sulle versioni
Db2	`db2-ae` `db2-se`	Db2 nelle versioni Amazon RDS
MariaDB	`mariadb`	Versioni di MariaDB in Amazon RDS
Microsoft SQL Server	`sqlserver-ee` `sqlserver-se` `sqlserver-ex` `sqlserver-web`	Versioni di Microsoft SQL Server su Amazon RDS
MySQL	`mysql`	Versioni di MySQL in Amazon RDS
Oracle	`oracle-ee` `oracle-se2`	Note di rilascio di Amazon RDS for Oracle
PostgreSQL	`postgres`	Versioni del database PostgreSQL disponibili

Per informazioni sui Regione AWS nomi, vedereAWS Regioni.

Gli esempi seguenti mostrano come determinare il supporto della classe di istanze DB in un Regione AWS utilizzando il AWS CLI comando describe-orderable-db-instance-options.

> **Note**
>
> Per limitare l'output, questi esempi mostrano i risultati solo per il tipo di archiviazione SSD (gp2) per uso generico. Se necessario, nei comandi è possibile modificare il tipo di archiviazione in SSD per uso generico (gp3), capacità di IOPS allocata (io1) o magnetico (standard).

Argomenti

- Elenco delle classi di istanza database supportate da una versione specifica del motore database in una Regione AWS
- Elenco delle versioni del motore DB che supportano una classe di istanza database specifica in una Regione AWS

Elenco delle classi di istanza database supportate da una versione specifica del motore database in una Regione AWS

Per elencare le classi di istanze DB supportate da una versione specifica del motore DB in un Regione AWS, esegui il comando seguente.

Per LinuxmacOS, oUnix:

```
aws rds describe-orderable-db-instance-options --engine engine --engine-version version
 \
    --query "*[].{DBInstanceClass:DBInstanceClass,StorageType:StorageType}|[?
StorageType=='gp2']|[].{DBInstanceClass:DBInstanceClass}" \
    --output text \
    --region region
```

Per Windows:

```
aws rds describe-orderable-db-instance-options --engine engine --engine-version version
 ^
    --query "*[].{DBInstanceClass:DBInstanceClass,StorageType:StorageType}|[?
StorageType=='gp2']|[].{DBInstanceClass:DBInstanceClass}" ^
    --output text ^
    --region region
```

Ad esempio, il comando seguente elenca le classi di istanza database supportate per la versione 13.6 del motore DB RDS per PostgreSQL in Stati Uniti orientali (Virginia settentrionale).

Per LinuxmacOS, oUnix:

```
aws rds describe-orderable-db-instance-options --engine postgres --engine-version 15.4
 \
    --query "*[].{DBInstanceClass:DBInstanceClass,StorageType:StorageType}|[?
StorageType=='gp2']|[].{DBInstanceClass:DBInstanceClass}" \
    --output text \
    --region us-east-1
```

Per Windows:

```
aws rds describe-orderable-db-instance-options --engine postgres --engine-version 15.4
 ^
    --query "*[].{DBInstanceClass:DBInstanceClass,StorageType:StorageType}|[?
StorageType=='gp2']|[].{DBInstanceClass:DBInstanceClass}" ^
    --output text ^
    --region us-east-1
```

Elenco delle versioni del motore DB che supportano una classe di istanza database specifica in una Regione AWS

Per elencare le versioni del motore DB che supportano una classe di istanza database specifica in una Regione AWS, emettere il comando seguente.

Per LinuxmacOS, oUnix:

```
aws rds describe-orderable-db-instance-options --engine engine --db-instance-
class DB_instance_class \
    --query "*[].{EngineVersion:EngineVersion,StorageType:StorageType}|[?
StorageType=='gp2']|[].{EngineVersion:EngineVersion}" \
    --output text \
    --region region
```

Per Windows:

```
aws rds describe-orderable-db-instance-options --engine engine --db-instance-
class DB_instance_class ^
    --query "*[].{EngineVersion:EngineVersion,StorageType:StorageType}|[?
StorageType=='gp2']|[].{EngineVersion:EngineVersion}" ^
```

```
    --output text ^
    --region region
```

Ad esempio, il comando seguente riporta le versioni del motore DB del motore di RDS per PostgreSQL DB che supportano la classe di istanza database db.r5.large in Stati Uniti orientali (Virginia settentrionale).

Per LinuxmacOS, oUnix:

```
aws rds describe-orderable-db-instance-options --engine postgres --db-instance-class
 db.m7g.large \
    --query "*[].{EngineVersion:EngineVersion,StorageType:StorageType}|[?
StorageType=='gp2']|[].{EngineVersion:EngineVersion}" \
    --output text \
    --region us-east-1
```

Per Windows:

```
aws rds describe-orderable-db-instance-options --engine postgres --db-instance-class
 db.m7g.large ^
    --query "*[].{EngineVersion:EngineVersion,StorageType:StorageType}|[?
StorageType=='gp2']|[].{EngineVersion:EngineVersion}" ^
    --output text ^
    --region us-east-1
```

Modifica della classe di istanza database

Puoi modificare la quantità di CPU e memoria disponibile per un'istanza database modificando la relativa classe di istanza database. Per modificare la classe di istanza database, modifica l'istanza database seguendo le istruzioni in Modifica di un'istanza database Amazon RDS.

Configurazione del processore per una classe di istanza database in RDS per Oracle

Le classi di istanze database Amazon RDS supportano la tecnologia Intel Hyper-Threading, che permette l'esecuzione simultanea di più thread su un singolo core CPU Intel Xeon. Ciascun thread è rappresentato come CPU virtuale (vCPU) nell'istanza database. Un'istanza database ha un numero predefinito di core CPU, che varia in base alla classe di istanza database. Ad esempio, un tipo di istanza database db.m4.xlarge ha due core CPU e due thread per core per impostazione predefinita, per un totale di quattro vCPU.

> **Note**
>
> Ciascuna vCPU è un hyperthread di un core CPU Intel Xeon.

Argomenti

Panoramica della configurazione del processore per RDS per Oracle

In genere, in caso di utilizzo di RDS per Oracle, è disponibile una classe di istanza database con una combinazione di memoria e numero di vCPU adatta ai carichi di lavoro specifici. Tuttavia, puoi anche specificare le caratteristiche del processore seguenti per ottimizzare l'istanza database RDS per Oracle per carichi di lavoro specifici o determinate esigenze aziendali:

- Numero di core CPU – È possibile personalizzare il numero di core CPU per l'istanza database. Questo ti offre la possibilità di ottimizzare i costi di licenza del software con un'istanza database dotata di una quantità sufficiente di RAM per carichi di lavoro a elevato utilizzo di memoria, ma di un numero minore di core CPU.
- Thread per core – È possibile disabilitare la tecnologia Intel Hyper-Threading specificando un singolo thread per core CPU. Potresti scegliere questa opzione per determinati carichi di lavoro, come quelli HPC (High Performance Computing).

Puoi controllare il numero di core CPU e quello di thread per ogni core separatamente. In una richiesta puoi impostare uno di questi valori o entrambi. Dopo che un'impostazione viene associata a un'istanza database, viene conservata fino a quando non la modifichi.

Le impostazioni del processore per un'istanza database sono associate agli snapshot dell'istanza database. Quando uno snapshot viene ripristinato, la relativa istanza database ripristinata usa le impostazioni delle caratteristiche del processore in uso al momento dell'acquisizione dello snapshot.

Se modifichi la classe di un'istanza database con impostazioni del processore non predefinite, specifica le impostazioni del processore predefinite o specifica in modo esplicito le impostazioni del processore all'esecuzione della modifica. Ciò garantisce che tu sia consapevole dei costi di licenza di terze parti che potrebbero venire addebitati in caso di modifica dell'istanza database.

Non sono previsti costi aggiuntivi o ridotti quando vengono specificate le caratteristiche del processore in un'istanza database RDS per Oracle. Ti vengono addebitati gli stessi costi delle istanze database avviate con le configurazioni CPU predefinite.

Classi di istanza database che supportano la configurazione del processore

È possibile configurare il numero di core e thread della CPU per core solo quando vengono soddisfatte le seguenti condizioni:

- Stai configurando un'istanza database RDS per Oracle. Per informazioni sulle classi di istanze database supportate dalle diverse edizioni di Oracle Database, consulta Classi di istanza RDS for Oracle.
- L'istanza database utilizza l'opzione di licenza Uso di licenze proprie (BYOL) di RDS per Oracle. Per ulteriori informazioni sulle opzioni di licenza Oracle, consulta Opzioni di licenza per RDS per Oracle.
- L'istanza database non appartiene a una delle classi di istanza db.r5 o db.r5b con configurazioni predefinite del processore. Queste classi di istanza hanno nomi nel formato db.r5.*instance_size*.tpc*threads_per_core*.mem*ratio* o db.r5b.*instance_size*.tpc*threads_per_core*.mem*ratio*. Ad esempio, db.r5b.xlarge.tpc2.mem4x è preconfigurato con 2 thread per core (tpc2) e 4 volte (4x) più memoria della classe di istanza standard db.r5b.xlarge. Non è possibile configurare le funzionalità del processore di queste classi di istanza ottimizzate. Per ulteriori informazioni, consulta Classi di istanza RDS per Oracle supportate.

Nella tabella seguente sono indicate le classi di istanze database che supportano l'impostazione di un numero di core CPU e di thread CPU per core. Sono indicati anche il valore predefinito e i valori validi per il numero di core CPU e di thread CPU per core per ogni classe di istanza database.

DB instance class (Classe istanza database)	vCPU predefinite	Core CPU predefiniti	Thread per core predefiniti	Numero valido di core CPU	Numero valido di thread per core
db.m6i: classi di istanza ottimizzata per la memoria					
db.m6i.large	2	1	2	1	1, 2

DB instance class (Classe istanza database)	vCPU predefinite	Core CPU predefiniti	Thread per core predefiniti	Numero valido di core CPU	Numero valido di thread per core
db.m6i.xlarge	4	2	2	2	1, 2
db.m6i.2xlarge	8	4	2	2, 4	1, 2
db.m6i.4xlarge	16	8	2	2, 4, 6, 8	1, 2
db.m6i.4xlarge	16	8	2	2, 4, 6, 8	1, 2
db.m6i.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
db.m6i.12xlarge	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2
db.m6i.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
db.m6i.24xlarge	96	48	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2

DB instance class (Classe istanza database)	vCPU predefinite	Core CPU predefiniti	Thread per core predefiniti	Numero valido di core CPU	Numero valido di thread per core
db.m6i.32xlarge	128	64	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64	1, 2
db.m5: classi di istanze per uso generico					
db.m5.large	2	1	2	1	1, 2
db.m5.xlarge	4	2	2	2	1, 2
db.m5.2xlarge	8	4	2	2, 4	1, 2
db.m5.4xlarge	16	8	2	2, 4, 6, 8	1, 2
db.m5.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
db.m5.12xlarge	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2

DB instance class (Classe istanza database)	vCPU predefinite	Core CPU predefiniti	Thread per core predefiniti	Numero valido di core CPU	Numero valido di thread per core
db.m5.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
db.m5.24xlarge	96	48	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2
db.m5d: classi di istanze per uso generico					
db.m5d.large	2	1	2	1	1, 2
db.m5d.xlarge	4	2	2	2	1, 2
db.m5d.2xlarge	8	4	2	2, 4	1, 2
db.m5d.4xlarge	16	8	2	2, 4, 6, 8	1, 2
db.m5d.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
db.m5d.12xlarge	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2

DB instance class (Classe istanza database)	vCPU predefinite	Core CPU predefiniti	Thread per core predefiniti	Numero valido di core CPU	Numero valido di thread per core
db.m5d.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
db.m5d.24xlarge	96	48	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2
db.m4: classi di istanze per uso generico					
db.m4.10xlarge	40	20	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20	1, 2
db.m4.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
db.r6i: classi di istanze ottimizzate per la memoria					
db.r6i.large	2	1	2	1	1, 2
db.r6i.xlarge	4	2	2	1, 2	1, 2
db.r6i.2xlarge	8	4	2	2, 4	1, 2

DB instance class (Classe istanza database)	vCPU predefinite	Core CPU predefiniti	Thread per core predefiniti	Numero valido di core CPU	Numero valido di thread per core
db.r6i.4xlarge	16	8	2	2, 4, 6, 8	1, 2
db.r6i.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
db.r6i.12xlarge	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2
db.r6i.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
db.r6i.24xlarge	96	48	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2

DB instance class (Classe istanza database)	vCPU predefinite	Core CPU predefiniti	Thread per core predefiniti	Numero valido di core CPU	Numero valido di thread per core
db.r6i.32xlarge	128	64	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64	1, 2
db.r5: classi di istanze ottimizzate per la memoria					
db.r5.large	2	1	2	1	1, 2
db.r5.xlarge	4	2	2	2	1, 2
db.r5.2xlarge	8	4	2	2, 4	1, 2
db.r5.4xlarge	16	8	2	2, 4, 6, 8	1, 2
db.r5.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
db.r5.12xlarge	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2

DB instance class (Classe istanza database)	vCPU predefinite	Core CPU predefiniti	Thread per core predefiniti	Numero valido di core CPU	Numero valido di thread per core
db.r5.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
db.r5.24xlarge	96	48	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2
db.r5: classi di istanze ottimizzate per la memoria					
db.r5b.large	2	1	2	1	1, 2
db.r5b.xlarge	4	2	2	2	1, 2
db.r5b.2xlarge	8	4	2	2, 4	1, 2
db.r5b.4xlarge	16	8	2	2, 4, 6, 8	1, 2
db.r5b.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
db.r5b.12xlarge	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2

DB instance class (Classe istanza database)	vCPU predefinite	Core CPU predefiniti	Thread per core predefiniti	Numero valido di core CPU	Numero valido di thread per core
db.r5b.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
db.r5b.24xlarge	96	48	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2
db.r5d: classi di istanze ottimizzate per la memoria					
db.r5d.large	2	1	2	1	1, 2
db.r5d.xlarge	4	2	2	2	1, 2
db.r5d.2xlarge	8	4	2	2, 4	1, 2
db.r5d.4xlarge	16	8	2	2, 4, 6, 8	1, 2
db.r5d.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
db.r5d.12xlarge	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2

DB instance class (Classe istanza database)	vCPU predefinite	Core CPU predefiniti	Thread per core predefiniti	Numero valido di core CPU	Numero valido di thread per core
db.r5d.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
db.r5d.24xlarge	96	48	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2
db.r4: classi di istanze ottimizzate per la memoria					
db.r4.large	2	1	2	1	1, 2
db.r4.xlarge	4	2	2	1, 2	1, 2
db.r4.2xlarge	8	4	2	1, 2, 3, 4	1, 2
db.r4.4xlarge	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
db.r4.8xlarge	32	16	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1, 2

DB instance class (Classe istanza database)	vCPU predefinite	Core CPU predefiniti	Thread per core predefiniti	Numero valido di core CPU	Numero valido di thread per core
db.r4.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
db.r3: classi di istanze ottimizzate per la memoria					
db.r3.large	2	1	2	1	1, 2
db.r3.xlarge	4	2	2	1, 2	1, 2
db.r3.2xlarge	8	4	2	1, 2, 3, 4	1, 2
db.r3.4xlarge	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2
db.r3.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
db.x2idn: classi di istanza ottimizzata per la memoria					
db.x2idn.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2

DB instance class (Classe istanza database)	vCPU predefinite	Core CPU predefiniti	Thread per core predefiniti	Numero valido di core CPU	Numero valido di thread per core
db.x2idn.24xlarge	96	48	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2
db.x2idn.32xlarge	128	64	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64	1, 2
db.x2iedn: classi di istanza ottimizzata per la memoria					
db.x2iedn.xlarge	4	2	2	1, 2	1, 2
db.x2iedn.2xlarge	8	4	2	2, 4	1, 2
db.x2iedn.4xlarge	16	8	2	2, 4, 6, 8	1, 2
db.x2iedn.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2

DB instance class (Classe istanza database)	vCPU predefinite	Core CPU predefiniti	Thread per core predefiniti	Numero valido di core CPU	Numero valido di thread per core
db.x2iedn.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
db.x2iedn.24xlarge	96	48	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48	1, 2
db.x2iedn.32xlarge	128	64	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62, 64	1, 2
db.x2iezn: classi di istanza ottimizzata per la memoria					
db.x2iezn.2xlarge	8	4	2	2, 4	1, 2
db.x2iezn.4xlarge	16	8	2	2, 4, 6, 8	1, 2

DB instance class (Classe istanza database)	vCPU predefinite	Core CPU predefiniti	Thread per core predefiniti	Numero valido di core CPU	Numero valido di thread per core
db.x2iezn.6xlarge	24	12	2	2, 4, 6, 8, 10, 12	1, 2
db.x2iezn.8xlarge	32	16	2	2, 4, 6, 8, 10, 12, 14, 16	1, 2
db.x2iezn.12xlarge	48	24	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2
db.x1: classi di istanze ottimizzate per la memoria					
db.x1.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
db.x1.32xlarge	128	64	2	4, 8, 12, 16, 20, 24, 28, 32, 36, 40, 44, 48, 52, 56, 60, 64	1, 2
db.x1e: classi di istanze ottimizzate per la memoria					
db.x1e.xlarge	4	2	2	1, 2	1, 2
db.x1e.2xlarge	8	4	2	1, 2, 3, 4	1, 2
db.x1e.4xlarge	16	8	2	1, 2, 3, 4, 5, 6, 7, 8	1, 2

DB instance class (Classe istanza database)	vCPU predefinite	Core CPU predefiniti	Thread per core predefiniti	Numero valido di core CPU	Numero valido di thread per core
db.x1e.8xlarge	32	16	2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16	1, 2
db.x1e.16xlarge	64	32	2	2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32	1, 2
db.x1e.32xlarge	128	64	2	4, 8, 12, 16, 20, 24, 28, 32, 36, 40, 44, 48, 52, 56, 60, 64	1, 2
db.z1d: classi di istanze ottimizzate per la memoria					
db.z1d.large	2	1	2	1	1, 2
db.z1d.xlarge	4	2	2	2	1, 2
db.z1d.2xlarge	8	4	2	2, 4	1, 2
db.z1d.3xlarge	12	6	2	2, 4, 6	1, 2
db.z1d.6xlarge	24	12	2	2, 4, 6, 8, 10, 12	1, 2

DB instance class (Classe istanza database)	vCPU predefinite	Core CPU predefiniti	Thread per core predefiniti	Numero valido di core CPU	Numero valido di thread per core
db.z1d.12xlarge	48	24	2	4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24	1, 2

> **Note**
>
> Puoi utilizzarlo AWS CloudTrail per monitorare e controllare le modifiche alla configurazione del processo delle istanze DB di Amazon RDS for Oracle. Per ulteriori informazioni sull'utilizzo CloudTrail, consulta. Monitoraggio delle chamate API di Amazon RDS in AWS CloudTrail

Impostazione dei core CPU e dei thread per core CPU per una classe di istanza database

Puoi configurare il numero di core CPU e di thread per core per la classe di istanza database quando esegui le operazioni seguenti:

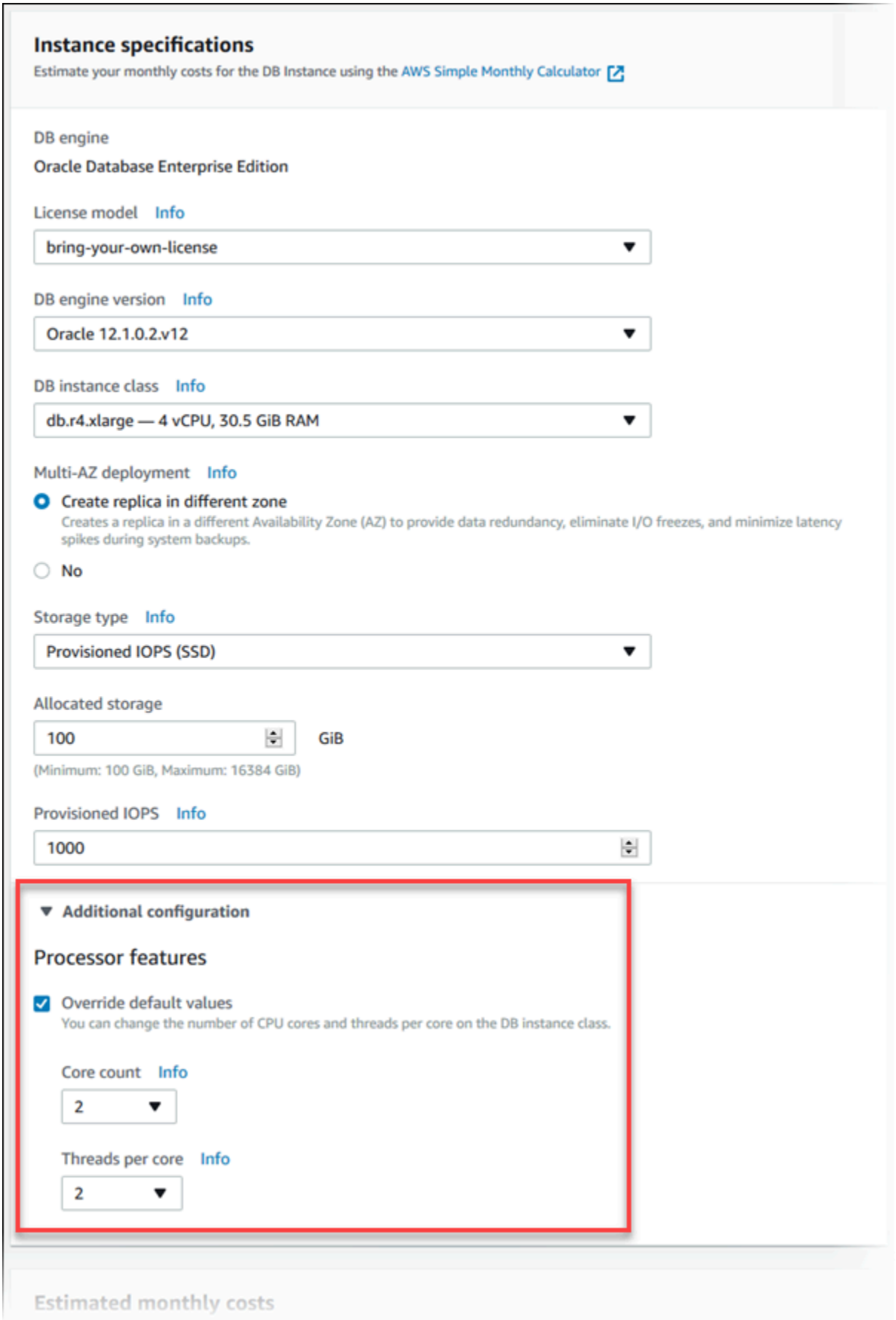
- Creazione di un'istanza database Amazon RDS
- Modifica di un'istanza database Amazon RDS
- Ripristino da uno snapshot database
- Ripristino a un'ora specifica per un'istanza database

> **Note**
>
> Quando modifichi un'istanza database per configurare il numero di core CPU o di thread per core, si verifica una breve interruzione dell'istanza database.

È possibile impostare i core della CPU e i thread per core della CPU per una classe di istanza DB utilizzando l' AWS Management Console API AWS CLI, the o RDS.

Console

Quando crei, modifichi o ripristini un'istanza database, devi impostare la classe dell'istanza database nella AWS Management Console. Nella sezione Instance specifications (Specifiche dell'istanza) sono visualizzate le opzioni per il processore. Nella figura seguente sono illustrate le opzioni relative alle caratteristiche del processore.

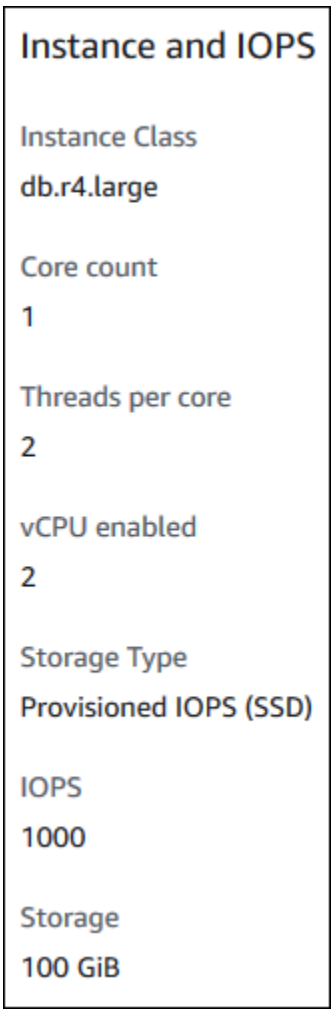

Instance specifications
Estimate your monthly costs for the DB Instance using the AWS Simple Monthly Calculator
DB engine
Oracle Database Enterprise Edition
License model Info
bring-your-own-license
DB engine version Info
Oracle 12.1.0.2.v12
DB instance class Info
db.r4.xlarge — 4 vCPU, 30.5 GiB RAM
Multi-AZ deployment Info
Create replica in different zone
Creates a replica in a different Availability Zone (AZ) to provide data redundancy, eliminate I/O freezes, and minimize latency spikes during system backups.
No
Storage type Info
Provisioned IOPS (SSD)
Allocated storage
100
GiB
(Minimum: 100 GiB, Maximum: 16384 GiB)
Provisioned IOPS Info
1000
Additional configuration
Processor features
Override default values
You can change the number of CPU cores and threads per core on the DB instance class.
Core count Info
2
Threads per core Info
2
Estimated monthly costs

Imposta le opzioni seguenti sui valori appropriati per la classe di istanza database in Processor features (Caratteristiche processore):

- Core count (Numero di core) – Imposta il numero di core CPU usando questa opzione. Questo valore deve essere minore o uguale al numero massimo di core CPU per la classe di istanza database.
- Threads per core (Thread per core) – Specifica 2 per abilitare i thread multipli per core oppure 1 per disabilitarli.

Quando modifichi o ripristini un'istanza database, puoi anche impostare i core CPU e i thread per core CPU sul valore predefinito per la classe di istanza.

Quando visualizzi i dettagli di un'istanza database nella console, puoi visualizzare le informazioni del processore per la relativa classe di istanza database nella scheda Configuration (Configurazione). Nella figura seguente è illustrata una classe di istanza database con un core CPU e più thread per core abilitati.



Per le istanze database Oracle, le informazioni sul processore vengono visualizzate solo per le istanze database BYOL (Bring Your Own License).

AWS CLI

Puoi impostare le caratteristiche del processore per un'istanza database quando esegui uno dei comandi di AWS CLI seguenti:

- create-db-instance
- modify-db-instance
- restore-db-instance-from-db-snapshot
- restore-db-instance-from-s3
- restore-db-instance-to-point-in-time

Per configurare il processore di una classe di istanza DB per un'istanza DB utilizzando AWS CLI, includi l'`--processor-features`opzione nel comando. Specifica il numero di core CPU con il nome di caratteristica `coreCount` e indica se sono abilitati i thread multipli per core con il nome di caratteristica `threadsPerCore`.

L'opzione ha la sintassi seguente.

```
--processor-features "Name=coreCount,Value=<value>" "Name=threadsPerCore,Value=<value>"
```

Di seguito vengono illustrati esempi di configurazione del processore:

Esempi

- Impostazione del numero di core CPU per un'istanza database
- Impostazione del numero di core CPU e disabilitazione dei thread multipli per un'istanza database
- Visualizzazione dei valori del processore validi per una classe di istanza database
- Ripristino delle impostazioni del processore predefinite per un'istanza database
- Ripristino del numero predefinito di core CPU per un'istanza database
- Ripristino del numero predefinito di thread per core per un'istanza database

Impostazione del numero di core CPU per un'istanza database

Example

L'esempio seguente modifica `mydbinstance` impostando il numero di core CPU su 4. Le modifiche vengono applicate immediatamente tramite `--apply-immediately`. Se desideri applicare le modifiche durante la successiva finestra di manutenzione pianificata, ometti l'opzione `--apply-immediately`.

Per LinuxmacOS, oUnix:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --processor-features "Name=coreCount,Value=4" \
    --apply-immediately
```

Per Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --processor-features "Name=coreCount,Value=4" ^
    --apply-immediately
```

Impostazione del numero di core CPU e disabilitazione dei thread multipli per un'istanza database

Example

L'esempio seguente modifica `mydbinstance` impostando il numero di core CPU su 4 e disabilitando i thread multipli per core. Le modifiche vengono applicate immediatamente tramite `--apply-immediately`. Se desideri applicare le modifiche durante la successiva finestra di manutenzione pianificata, ometti l'opzione `--apply-immediately`.

Per LinuxmacOS, oUnix:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --processor-features "Name=coreCount,Value=4" "Name=threadsPerCore,Value=1" \
    --apply-immediately
```

Per Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --processor-features "Name=coreCount,Value=4" "Name=threadsPerCore,Value=1" ^
    --apply-immediately
```

Visualizzazione dei valori del processore validi per una classe di istanza database

Example

È possibile visualizzare i valori validi del processore per una particolare classe di istanza DB eseguendo il comando describe-orderable-db-instance-options e specificando la classe di istanza per l'`--db-instance-class`opzione. Ad esempio, l'output del comando seguente mostra le opzioni del processore per la classe di istanza db.r3.large.

```
aws rds describe-orderable-db-instance-options --engine oracle-ee --db-instance-class
 db.r3.large
```

Di seguito è illustrato l'output di esempio per il comando in formato JSON.


```
    {
                "SupportsIops": true,
                "MaxIopsPerGib": 50.0,
                "LicenseModel": "bring-your-own-license",
                "DBInstanceClass": "db.r3.large",
                "SupportsIAMDatabaseAuthentication": false,
                "MinStorageSize": 100,
                "AvailabilityZones": [
                    {
                        "Name": "us-west-2a"
                    },
                    {
                        "Name": "us-west-2b"
                    },
                    {
                        "Name": "us-west-2c"
                    }
                ],
                "EngineVersion": "12.1.0.2.v2",
                "MaxStorageSize": 32768,
                "MinIopsPerGib": 1.0,
                "MaxIopsPerDbInstance": 40000,
                "ReadReplicaCapable": false,
                "AvailableProcessorFeatures": [
                    {
                        "Name": "coreCount",
                        "DefaultValue": "1",
                        "AllowedValues": "1"
                    },
                    {
                        "Name": "threadsPerCore",
                        "DefaultValue": "2",
                        "AllowedValues": "1,2"
                    }
                ],
                "SupportsEnhancedMonitoring": true,
                "SupportsPerformanceInsights": false,
                "MinIopsPerDbInstance": 1000,
                "StorageType": "io1",
                "Vpc": false,
                "SupportsStorageEncryption": true,
                "Engine": "oracle-ee",
```

```
                "MultiAZCapable": true
    }
```

Puoi inoltre eseguire i comandi seguenti per ottenere informazioni sul processore della classe di istanza database:

- describe-db-instances— Mostra le informazioni sul processore per l'istanza DB specificata.
- describe-db-snapshots— Mostra le informazioni sul processore per l'istantanea del DB specificata.
- describe-valid-db-instance-modifications: mostra le modifiche valide al processore per l'istanza DB specificata.

Nell'output dei comandi precedenti, i valori per le funzionalità del processore non sono null solo se sono soddisfatte le seguenti condizioni:

- Si sta utilizzando un'istanza database RDS per Oracle.
- L'istanza database RDS per Oracle supporta la modifica dei valori del processore.
- Le impostazioni correnti del core e del thread della CPU sono impostate su valori non predefiniti.

Se le condizioni precedenti non sono soddisfatte, puoi ottenere il tipo di istanza utilizzando. describe-db-instances È possibile ottenere le informazioni sul processore per questo tipo di istanza eseguendo l'operazione EC2. describe-instance-types

Ripristino delle impostazioni del processore predefinite per un'istanza database

Example

L'esempio seguente modifica `mydbinstance` ripristinando i valori del processore predefiniti per la classe di istanza database. Le modifiche vengono applicate immediatamente tramite `--apply-immediately`. Se desideri applicare le modifiche durante la successiva finestra di manutenzione pianificata, ometti l'opzione `--apply-immediately`.

Per LinuxmacOS, oUnix:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --use-default-processor-features \
    --apply-immediately
```

Per Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --use-default-processor-features ^
    --apply-immediately
```

Ripristino del numero predefinito di core CPU per un'istanza database

Example

L'esempio seguente modifica `mydbinstance` ripristinando il numero predefinito di core CPU per la classe di istanza database. L'impostazione relativa ai thread per core non viene modificata. Le modifiche vengono applicate immediatamente tramite `--apply-immediately`. Se desideri applicare le modifiche durante la successiva finestra di manutenzione pianificata, ometti l'opzione `--apply-immediately`.

Per LinuxmacOS, oUnix:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --processor-features "Name=coreCount,Value=DEFAULT" \
    --apply-immediately
```

Per Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --processor-features "Name=coreCount,Value=DEFAULT" ^
    --apply-immediately
```

Ripristino del numero predefinito di thread per core per un'istanza database

Example

L'esempio seguente modifica `mydbinstance` ripristinando il numero predefinito di thread per core per la classe di istanza database. L'impostazione relativa al numero di core CPU non viene modificata. Le modifiche vengono applicate immediatamente tramite `--apply-immediately`. Se desideri applicare le modifiche durante la successiva finestra di manutenzione pianificata, ometti l'opzione `--apply-immediately`.

Per LinuxmacOS, oUnix:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --processor-features "Name=threadsPerCore,Value=DEFAULT" \
    --apply-immediately
```

Per Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --processor-features "Name=threadsPerCore,Value=DEFAULT" ^
    --apply-immediately
```

API RDS

Puoi impostare le caratteristiche del processore per un'istanza database quando chiami una delle operazioni API Amazon RDS seguenti:

- CreateDBInstance
- ModifyDBInstance
- Restore DB DB InstanceFrom Snapshot
- Ripristina InstanceFrom DB S3
- Restore DB InstanceToPointInTime

Per configurare le caratteristiche del processore di una classe di istanza database usando l'API Amazon RDS, includi il parametro `ProcessFeatures` nella chiamata.

Il parametro ha la sintassi seguente.

```
ProcessFeatures "Name=coreCount,Value=<value>" "Name=threadsPerCore,Value=<value>"
```

Specifica il numero di core CPU con il nome di caratteristica `coreCount` e indica se sono abilitati i thread multipli per core con il nome di caratteristica `threadsPerCore`.

È possibile visualizzare i valori validi del processore per una particolare classe di istanza DB eseguendo l'InstanceOptionsoperazione DescribeOrderableDB e specificando la classe di istanza per il parametro. `DBInstanceClass` È inoltre possibile utilizzare le seguenti operazioni:

- DescribeDBInstances: mostra le informazioni sul processore per l'istanza database specificata.
- DescribeDBSnapshots: mostra le informazioni sul processore per lo snapshot DB specificato.
- DescribeValidDB InstanceModifications: mostra le modifiche valide al processore per l'istanza DB specificata.

Nell'output delle operazioni precedenti, i valori per le funzionalità del processore non sono null solo se sono soddisfatte le seguenti condizioni:

- Si sta utilizzando un'istanza database RDS per Oracle.
- L'istanza database RDS per Oracle supporta la modifica dei valori del processore.
- Le impostazioni correnti del core e del thread della CPU sono impostate su valori non predefiniti.

Se le condizioni precedenti non vengono soddisfatte, è possibile ottenere il tipo di istanza utilizzando DescribedBInstances. È possibile ottenere le informazioni sul processore per questo tipo di istanza eseguendo l'operazione EC2. DescribeInstanceTypes

Specifiche hardware per le classi di istanza database

La terminologia seguente viene utilizzata per descrivere le specifiche dell'hardware per le classi di istanza database:

VPCU

Numero di unità centrali di elaborazione (CPU). Una CPU virtuale è un'unità di capacità che puoi usare per confrontare le classi di istanza database. Invece di acquistare o affittare un determinato processore da utilizzare per vari mesi o anni, si affitta la capacità su base oraria. L'obiettivo è quello di rendere disponibile una quantità coerente e specifica di capacità di CPU, entro i limiti dell'hardware effettivo sottostante.

ECU

Misura relativa della potenza di elaborazione intera di un'istanza Amazon EC2. Per permettere agli sviluppatori di confrontare in modo semplice la capacità della CPU tra diverse classi di istanza, abbiamo definito un'unità di elaborazione Amazon EC2. La quantità di CPU allocata in una determinata istanza viene espressa in unità di calcolo o unità di elaborazione EC2. Un'unità ECU attualmente fornisce una capacità di CPU equivalente a un processore Opteron 2007 o Xeon 2007 da 1,0 – 1,2 GHz.

Memoria (GiB)

La RAM, in gibibyte, allocata all'istanza database. Spesso c'è un rapporto costante tra memoria e vCPU. A titolo esemplificativo, prendi la classe di istanza db.r4, che ha una memoria in rapporto vCPU simile alla classe di istanza db.r5. Tuttavia, per la maggior parte dei casi d'uso la classe di istanza db.r5 fornisce prestazioni migliori e più costanti rispetto alla classe di istanza db.r4.

Ottimizzato per EBS

L'istanza database utilizza uno stack di configurazione ottimizzato e offre capacità aggiuntiva dedicata per l'I/O di Amazon EBS. Questa ottimizzazione offre prestazioni ottimali ai volumi EBS, riducendo al minimo i conflitti tra l'I/O di Amazon EBS e altro traffico proveniente dall'istanza. Per ulteriori informazioni sulle istanze ottimizzate per Amazon EBS, consulta Istanze ottimizzate per Amazon EBS nella Guida per l'utente di Amazon EC2 per istanze Linux.

Le istanze ottimizzate per EBS hanno un numero di IOPS di base e massimo. Il numero di IOPS massimo viene applicato a livello di istanza database. Un set di volumi EBS combinati per avere un numero di IOPS superiore al massimo, non potrà comunque avere un valore maggiore della soglia a livello di istanza. Ad esempio, se il numero massimo di IOPS per una classe di istanza database specifica è 40.000 e si collegano quattro volumi EBS da 64.000 IOPS, il numero massimo di IOPS è 40.000 e non 256.000. Per il numero di IOPS per ciascun tipo di istanza EC2, consulta Tipi di istanza supportati nella Guida per l'utente per istanze Linux di Amazon EC2.

Quantità max Larghezza di banda EBS (Mbps)

La larghezza di banda EBS massima in megabit al secondo. Dividendo il valore per 8, puoi ottenere il throughput previsto in megabyte al secondo.

> **Important**
>
> I volumi SSD per scopi generici (gp2) per istanze database di Amazon RDS hanno un limite di throughput di 250 MiB/s nella maggior parte dei casi. Tuttavia, questo limite può variare in base alla dimensione del volume. Per ulteriori informazioni, consulta Tipi di volume Amazon EBS nella Guida per l'utente di Amazon EC2 per le istanze Linux.

Larghezza di banda di rete

La velocità di rete relativa ad altre classi di istanza database.

Nella tabella seguente, sono riportati i dettagli hardware relativi alle classi di istanza database Amazon RDS .

Per informazioni sul supporto del motore del database di Amazon RDS per ciascuna classe di istanza database, consulta Motori DB supportati per classi di istanza database.

Classe istanza	VPCU	ECU	Memoria (GiB)	Archiviazione delle istanze (GiB)	Quantità max Larghezza di banda EBS (Mbps)	Larghezza di banda della rete (Gbps)
db.m7g — classi di istanze generiche con processori Graviton3 AWS						
db.m7g.16xlarge	64	—	256	Solo ottimizzata per EBS	20.000	30
db.m7g.12xlarge	48	—	192	Solo ottimizzata per EBS	15.000	22.5
db.m7g.8xlarge	32	—	128	Solo ottimizzata per EBS	10.000	15
db.m7g.4xlarge	16	—	64	Solo ottimizzata per EBS	Fino a 10.000	Fino a 15
db.m7g.2xlarge*	8	—	32	Solo ottimizzata per EBS	Fino a 10.000	Fino a 15
db.m7g.xlarge*	4	—	16	Solo ottimizzata per EBS	Fino a 10.000	Fino a 12,5
db.m7g.large*	2	—	8	Solo ottimizzata per EBS	Fino a 10.000	Fino a 12,5
db.m6g — classi di istanze generiche con processori Graviton2 AWS						
db.m6g.16xlarge	64	—	256	Solo ottimizzata per EBS	19.000	25

Classe istanza	VPCU	ECU	Memoria (GiB)	Archiviazione delle istanze (GiB)	Quantità max Larghezza di banda EBS (Mbps)	Larghezza di banda della rete (Gbps)
db.m6g.12xlarge	48	—	192	Solo ottimizzata per EBS	13.500	20
db.m6g.8xlarge	32	—	128	Solo ottimizzata per EBS	9.500	12
db.m6g.4xlarge	16	—	64	Solo ottimizzata per EBS	6.800	Fino a 10
db.m6g.2xlarge*	8	—	32	Solo ottimizzata per EBS	Fino a 4.750	Fino a 10
db.m6g.xlarge*	4	—	16	Solo ottimizzata per EBS	Fino a 4.750	Fino a 10
db.m6g.large*	2	—	8	Solo ottimizzata per EBS	Fino a 4.750	Fino a 10
db.m6gd — classi di istanze generiche con processori Graviton2 e storage SSD AWS						
db.m6gd.16xlarge	64	—	256	2 x 1900 SSD NVMe	19.000	25
db.m6gd.12xlarge	48	—	192	2 x 1425 SSD NVMe	13.500	20
db.m6gd.8xlarge	32	—	128	1 x 1900 SSD NVMe	9.000	12
db.m6gd.4xlarge	16	—	64	1 x 950 SSD NVMe	4.750	Fino a 10
db.m6gd.2xlarge	8	—	32	1 x 474 SSD NVMe	Fino a 4.750	Fino a 10

Classe istanza	VPCU	ECU	Memoria (GiB)	Archiviazione delle istanze (GiB)	Quantità max Larghezza di banda EBS (Mbps)	Larghezza di banda della rete (Gbps)
db.m6gd.xlarge	4	—	16	1 x 237 SSD NVMe	Fino a 4.750	Fino a 10
db.m6gd.large	2	—	8	1 x 118 SSD NVMe	Fino a 4.750	Fino a 10
db.m6id: classi di istanza per uso generico con processori Intel Xeon scalabili di terza generazione e archiviazione SSD						
db.m6id.32xlarge	128	—	512	4 x 1900 SSD NVMe	40.000	50
db.m6id.24xlarge	96	—	384	4 x 1425 SSD NVMe	30.000	37,5
db.m6id.16xlarge	64	—	256	2 x 1900 SSD NVMe	20.000	25
db.m6id.12xlarge	48	—	192	2 x 1425 SSD NVMe	15.000	18.75
db.m6id.8xlarge	32	—	128	1 x 1900 SSD NVMe	10.000	12,5
db.m6id.4xlarge*	16	—	64	1 x 950 SSD NVMe	Fino a 10.000	Fino a 12,5
db.m6id.2xlarge*	8	—	32	1 x 474 SSD NVMe	Fino a 10.000	Fino a 12,5
db.m6id.xlarge*	4	—	16	1 x 237 SSD NVMe	Fino a 10.000	Fino a 12,5

Classe istanza	VPCU	ECU	Memoria (GiB)	Archiviazione delle istanze (GiB)	Quantità max Larghezza di banda EBS (Mbps)	Larghezza di banda della rete (Gbps)
db.m6id.large*	2	—	8	1 x 118 SSD NVMe	Fino a 10.000	Fino a 12,5
db.m6idn: classi di istanza per uso generico con processori scalabili Intel Xeon di terza generazione, archiviazione SSD e ottimizzazione di rete						
db.m6idn.32xlarge	128	—	512	4 x 1900 SSD NVMe	80.000	200
db.m6idn.24xlarge	96	—	384	4 x 1425 SSD NVMe	60.000	150
db.m6idn.16xlarge	64	—	256	2 x 1900 SSD NVMe	40.000	100
db.m6idn.12xlarge	48	—	192	2 x 1425 SSD NVMe	30.000	75
db.m6idn.8xlarge	32	—	128	1 x 1900 SSD NVMe	20.000	50
db.m6idn.4xlarge*	16	—	64	1 x 950 SSD NVMe	Fino a 20.000	Fino a 50
db.m6idn.2xlarge*	8	—	32	1 x 474 SSD NVMe	Fino a 20.000	Fino a 40
db.m6idn.xlarge*	4	—	16	1 x 237 SSD NVMe	Fino a 20.000	Fino a 30
db.m6idn.large*	2	—	8	1 x 118 SSD NVMe	Fino a 20.000	Fino a 25

Classe istanza	VPCU	ECU	Memoria (GiB)	Archiviazione delle istanze (GiB)	Quantità max Larghezza di banda EBS (Mbps)	Larghezza di banda della rete (Gbps)
db.m6idn: classi di istanza per uso generico con processori scalabili Intel Xeon di terza generazione e ottimizzazione di rete						
db.m6in.32xlarge	128	—	512	Solo ottimizzata per EBS	80.000	200
db.m6in.24xlarge	96	—	384	Solo ottimizzata per EBS	60.000	150
db.m6in.16xlarge	64	—	256	Solo ottimizzata per EBS	40.000	100
db.m6in.12xlarge	48	—	192	Solo ottimizzata per EBS	30.000	75
db.m6in.8xlarge	32	—	128	Solo ottimizzata per EBS	20.000	50
db.m6in.4xlarge*	16	—	64	Solo ottimizzata per EBS	Fino a 20.000	Fino a 50
db.m6in.2xlarge*	8	—	32	Solo ottimizzata per EBS	Fino a 20.000	Fino a 40
db.m6in.xlarge*	4	—	16	Solo ottimizzata per EBS	Fino a 20.000	Fino a 30
db.m6in.large*	2	—	8	Solo ottimizzata per EBS	Fino a 20.000	Fino a 25
db.m6i: classi di istanza per uso generico con processori Intel Xeon scalabili di terza generazione						
db.m6i.32xlarge	128	—	512	Solo ottimizzata per EBS	40.000	50

Classe istanza	VPCU	ECU	Memoria (GiB)	Archiviazione delle istanze (GiB)	Quantità max Larghezza di banda EBS (Mbps)	Larghezza di banda della rete (Gbps)
db.m6i.24xlarge	96	—	384	Solo ottimizzata per EBS	30.000	37,5
db.m6i.16xlarge	64	—	256	Solo ottimizzata per EBS	20.000	25
db.m6i.12xlarge	48	—	192	Solo ottimizzata per EBS	15.000	18,75
db.m6i.8xlarge	32	—	128	Solo ottimizzata per EBS	10.000	12,5
db.m6i.4xlarge*	16	—	64	Solo ottimizzata per EBS	Fino a 10.000	Fino a 12,5
db.m6i.2xlarge*	8	—	32	Solo ottimizzata per EBS	Fino a 10.000	Fino a 12,5
db.m6i.xlarge*	4	—	16	Solo ottimizzata per EBS	Fino a 10.000	Fino a 12,5
db.m6i.large*	2	—	8	Solo ottimizzata per EBS	Fino a 10.000	Fino a 12,5
db.m5d: classi di istanza per uso generico con processori Intel Xeon Platinum e archiviazione SSD						
db.m5d.24xlarge	96	345	384	4 x 900 SSD NVMe	19.000	25
db.m5d.16xlarge	64	262	256	4 x 600 SSD NVMe	13.600	20
db.m5d.12xlarge	48	173	192	2 x 900 SSD NVMe	9.500	10

Classe istanza	VPCU	ECU	Memoria (GiB)	Archiviazione delle istanze (GiB)	Quantità max Larghezza di banda EBS (Mbps)	Larghezza di banda della rete (Gbps)
db.m5d.8xlarge	32	131	128	2 x 600 SSD NVMe	6.800	10
db.m5d.4xlarge	16	61	64	2 x 300 SSD NVMe	4.750	Fino a 10
db.m5d.2xlarge	8	31	32	1 x 300 SSD NVMe	Fino a 4.750	Fino a 10
db.m5d.xlarge	4	15	16	1 x 150 SSD NVMe	Fino a 4.750	Fino a 10
db.m5d.large	2	10	8	1 x 75 SSD NVMe	Fino a 4.750	Fino a 10
db.m5: classi di istanza per uso generico con processori Intel Xeon Platinum						
db.m5.24xlarge	96	345	384	Solo ottimizzata per EBS	19.000	25
db.m5.16xlarge	64	262	256	Solo ottimizzata per EBS	13.600	20
db.m5.12xlarge	48	173	192	Solo ottimizzata per EBS	9.500	10
db.m5.8xlarge	32	131	128	Solo ottimizzata per EBS	6.800	10
db.m5.4xlarge	16	61	64	Solo ottimizzata per EBS	4.750	Fino a 10
db.m5.2xlarge*	8	31	32	Solo ottimizzata per EBS	Fino a 4.750	Fino a 10

Classe istanza	VPCU	ECU	Memoria (GiB)	Archiviazione delle istanze (GiB)	Quantità max Larghezza di banda EBS (Mbps)	Larghezza di banda della rete (Gbps)
db.m5.xlarge*	4	15	16	Solo ottimizzata per EBS	Fino a 4.750	Fino a 10
db.m5.large*	2	10	8	Solo ottimizzata per EBS	Fino a 4.750	Fino a 10
v: classi di istanza per uso generico con processori Intel Xeon scalabili						
db.m4.16xlarge	64	188	256	Solo ottimizzata per EBS	10.000	25
db.m4.10xlarge	40	124.5	160	Solo ottimizzata per EBS	4.000	10
db.m4.4xlarge	16	53.5	64	Solo ottimizzata per EBS	2.000	Elevate
db.m4.2xlarge	8	25.5	32	Solo ottimizzata per EBS	1.000	Elevate
db.m4.xlarge	4	13	16	Solo ottimizzata per EBS	750	Elevate
db.m4.large	2	6,5	8	Solo ottimizzata per EBS	450	Moderate
db.m3: classi di istanze per uso generico						
db.m3.2xlarge	8	26	30	Solo ottimizzata per EBS	1.000	Elevate
db.m3.xlarge	4	13	15	Solo ottimizzata per EBS	500	Elevate
db.m3.large	2	6,5	7,5	Solo EBS	—	Moderata

Classe istanza	VPCU	ECU	Memoria (GiB)	Archiviazione delle istanze (GiB)	Quantità max Larghezza di banda EBS (Mbps)	Larghezza di banda della rete (Gbps)
db.m3.medium	1	3	3,75	Solo EBS	—	Moderata
db.m1: classi di istanze per uso generico						
db.m1.xlarge	4	4	15	Solo ottimizzata per EBS	450	Elevate
db.m1.large	2	2	7,5	Solo ottimizzata per EBS	450	Moderate
db.m1.medium	1	1	3,75	Solo EBS	—	Moderata
db.m1.small	1	1	1,7	Solo EBS	—	Molto basse
db.x2iezn: classi di istanza ottimizzata per la memoria						
db.x2iezn.12xlarge	>48	—	1.536	Solo ottimizzata per EBS	19.000	100
db.x2iezn.8xlarge	32	—	1,024	Solo ottimizzata per EBS	12.000	75
db.x2iezn.6xlarge	24	—	768	Solo ottimizzata per EBS	Fino a 9.500	50
db.x2iezn.4xlarge	16	—	512	Solo ottimizzata per EBS	Fino a 4.750	Fino a 25
db.x2iezn.2xlarge	8	—	256	Solo ottimizzata per EBS	Fino a 3.170	Fino a 25
db.x2iedn: classi di istanza ottimizzata per la memoria con archiviazione SSD e ottimizzazione di rete						

Classe istanza	VPCU	ECU	Memoria (GiB)	Archiviazione delle istanze (GiB)	Quantità max Larghezza di banda EBS (Mbps)	Larghezza di banda della rete (Gbps)
db.x2iedn.32xlarge	128	—	4,096	2 x 1900 SSD NVMe	80.000	100
db.x2iedn.24xlarge	96	—	3.072	2 x 1425 SSD NVMe	60.000	75
db.x2iedn.16xlarge	64	—	2.048	1 x 1900 SSD NVMe	40.000	50
db.x2iedn.8xlarge	32	—	1,024	1 x 950 SSD NVMe	20.000	25
db.x2iedn.4xlarge	16	—	512	1 x 475 SSD NVMe	Fino a 20.000	Fino a 25
db.x2iedn.2xlarge	8	—	256	1 x 237 SSD NVMe	Fino a 20.000	Fino a 25
db.x2iedn.xlarge	4	—	128	1 x 118 SSD NVMe	Fino a 20.000	Fino a 25
db.x2idn: classi di istanza ottimizzata per la memoria con archiviazione SSD e ottimizzazione di rete						
db.x2idn.32xlarge	128	—	2.048	2 x 1900 SSD NVMe	80.000	100
db.x2idn.24xlarge	96	—	1.536	2 x 1425 SSD NVMe	60.000	75
db.x2idn.16xlarge	64	—	1,024	1 x 1900 SSD NVMe	40.000	50
db.x2g: classi di istanza ottimizzata per la memoria						

Classe istanza	VPCU	ECU	Memoria (GiB)	Archiviazione delle istanze (GiB)	Quantità max Larghezza di banda EBS (Mbps)	Larghezza di banda della rete (Gbps)
db.x2g.16xlarge	64	—	1.024	Solo ottimizzata per EBS	19.000	25
db.x2g.12xlarge	48	—	768	Solo ottimizzata per EBS	14.250	20
db.x2g.8xlarge	32	—	512	Solo ottimizzata per EBS	9.500	12
db.x2g.4xlarge	16	—	256	Solo ottimizzata per EBS	4.750	Fino a 10
db.x2g.2xlarge	8	—	128	Solo ottimizzata per EBS	Fino a 4.750	Fino a 10
db.x2g.xlarge	4	—	64	Solo ottimizzata per EBS	Fino a 4.750	Fino a 10
db.x2g.large	2	—	32	Solo ottimizzata per EBS	Fino a 4.750	Fino a 10
db.z1d: classi di istanza ottimizzata per la memoria con archiviazione SSD						
db.z1d.12xlarge	48	271	384	2 x 900 SSD NVMe	14.000	25
db.z1d.6xlarge	24	134	192	1 x 900 SSD NVMe	7,000	10
db.z1d.3xlarge	12	75	96	1 x 450 SSD NVMe	3,500	Fino a 10
db.z1d.2xlarge	8	53	64	1 x 300 SSD NVMe	2.333	Fino a 10

Classe istanza	VPCU	ECU	Memoria (GiB)	Archiviazione delle istanze (GiB)	Quantità max Larghezza di banda EBS (Mbps)	Larghezza di banda della rete (Gbps)
db.z1d.xlarge*	4	28	32	1 x 150 SSD NVMe	Fino a 2.333	Fino a 10
db.z1d.large*	2	15	16	1 x 75 SSD NVMe	Fino a 2.333	Fino a 10
db.x1e: classi di istanze ottimizzate per la memoria						
db.x1e.32xlarge	128	340	3,904	Solo ottimizzata per EBS	14.000	25
db.x1e.16xlarge	64	179	1,952	Solo ottimizzata per EBS	7,000	10
db.x1e.8xlarge	32	91	976	Solo ottimizzata per EBS	3,500	Fino a 10
db.x1e.4xlarge	16	47	488	Solo ottimizzata per EBS	1,750	Fino a 10
db.x1e.2xlarge	8	23	244	Solo ottimizzata per EBS	1.000	Fino a 10
db.x1e.xlarge	4	12	122	Solo ottimizzata per EBS	500	Fino a 10
db.x1: classi di istanze ottimizzate per la memoria						
db.x1.32xlarge	128	349	1,952	Solo ottimizzata per EBS	14.000	25
db.x1.16xlarge	64	174,5	976	Solo ottimizzata per EBS	7,000	10
db.r7g — classi di istanze ottimizzate per la memoria con processori Graviton3 AWS						

Classe istanza	VPCU	ECU	Memoria (GiB)	Archiviazione delle istanze (GiB)	Quantità max Larghezza di banda EBS (Mbps)	Larghezza di banda della rete (Gbps)
db.r7g.16xlarge	64	—	512	Solo ottimizzata per EBS	20.000	30
db.r7g.12xlarge	48	—	384	Solo ottimizzata per EBS	15.000	22.5
db.r7g.8xlarge	32	—	256	Solo ottimizzata per EBS	10.000	15
db.r7g.4xlarge	16	—	128	Solo ottimizzata per EBS	Fino a 10.000	Fino a 15
db.r7g.2xlarge*	8	—	64	Solo ottimizzata per EBS	Fino a 10.000	Fino a 15
db.r7g.xlarge*	4	—	32	Solo ottimizzata per EBS	Fino a 10.000	Fino a 12,5
db.r7g.large*	2	—	16	Solo ottimizzata per EBS	Fino a 10.000	Fino a 12,5
db.r6g — classi di istanze ottimizzate per la memoria con processori Graviton2 AWS						
db.r6g.16xlarge	64	—	512	Solo ottimizzata per EBS	19.000	25
db.r6g.12xlarge	48	—	384	Solo ottimizzata per EBS	13.500	20
db.r6g.8xlarge	32	—	256	Solo ottimizzata per EBS	9.000	12
db.r6g.4xlarge	16	—	128	Solo ottimizzata per EBS	4.750	Fino a 10

Classe istanza	VPCU	ECU	Memoria (GiB)	Archiviazione delle istanze (GiB)	Quantità max Larghezza di banda EBS (Mbps)	Larghezza di banda della rete (Gbps)
db.r6g.2xlarge*	8	—	64	Solo ottimizzata per EBS	Fino a 4.750	Fino a 10
db.r6g.xlarge*	4	—	32	Solo ottimizzata per EBS	Fino a 4.750	Fino a 10
db.r6g.large*	2	—	16	Solo ottimizzata per EBS	Fino a 4.750	Fino a 10
db.r6gd — classi di istanze ottimizzate per la memoria con processori Graviton2 e storage SSD AWS						
db.r6gd.16xlarge	64	—	512	2 x 1900 SSD NVMe	19.000	25
db.r6gd.12xlarge	48	—	384	2 x 1425 SSD NVMe	13.500	20
db.r6gd.8xlarge	32	—	256	1 x 1900 SSD NVMe	9.000	12
db.r6gd.4xlarge	16	—	128	1 x 950 SSD NVMe	4.750	Fino a 10
db.r6gd.2xlarge	8	—	64	1 x 474 SSD NVMe	Fino a 4.750	Fino a 10
db.r6gd.xlarge	4	—	32	1 x 237 SSD NVMe	Fino a 4.750	Fino a 10
db.r6gd.large	2	—	16	1 x 118 SSD NVMe	Fino a 4.750	Fino a 10

Classe istanza	VPCU	ECU	Memoria (GiB)	Archiviazione delle istanze (GiB)	Quantità max Larghezza di banda EBS (Mbps)	Larghezza di banda della rete (Gbps)
db.r6id: classi di istanza per uso generico con processori Intel Xeon scalabili di terza generazione e archiviazione SSD						
db.r6id.32xlarge	128	—	1,024	4 x 1900 SSD NVMe	40.000	50
db.r6id.24xlarge	96	—	768	4 x 1425 SSD NVMe	30.000	37,5
db.r6id.16xlarge	64	—	512	2 x 1900 SSD NVMe	20.000	25
db.r6id.12xlarge	48	—	384	2 x 1425 SSD NVMe	15.000	18,75
db.r6id.8xlarge	32	—	256	1 x 1900 SSD NVMe	10.000	12,5
db.r6id.4xlarge*	16	—	128	1 x 950 SSD NVMe	Fino a 10.000	Fino a 12,5
db.r6id.2xlarge*	8	—	64	1 x 474 SSD NVMe	Fino a 10.000	Fino a 12,5
db.r6id.xlarge*	4	—	32	1 x 237 SSD NVMe	Fino a 10.000	Fino a 12,5
db.r6id.large*	2	—	16	1 x 118 SSD NVMe	Fino a 10.000	Fino a 12,5
db.r6idn: classi di istanza ottimizzata per la memoria con processori scalabili Intel Xeon di terza generazione, archiviazione SSD e ottimizzazione di rete						

Classe istanza	VPCU	ECU	Memoria (GiB)	Archiviazione delle istanze (GiB)	Quantità max Larghezza di banda EBS (Mbps)	Larghezza di banda della rete (Gbps)
db.r6idn.32xlarge	128	—	1,024	4 x 1900 SSD NVMe	80.000	200
db.r6idn.24xlarge	96	—	768	4 x 1425 SSD NVMe	60.000	150
db.r6idn.16xlarge	64	—	512	2 x 1900 SSD NVMe	40.000	100
db.r6idn.12xlarge	48	—	384	2 x 1425 SSD NVMe	30.000	75
db.r6idn.8xlarge	32	—	256	1 x 1900 SSD NVMe	20.000	50
db.r6idn.4xlarge*	16	—	128	1 x 950 SSD NVMe	Fino a 20.000	Fino a 50
db.r6idn.2xlarge*	8	—	64	1 x 474 SSD NVMe	Fino a 20.000	Fino a 40
db.r6idn.xlarge*	4	—	32	1 x 237 SSD NVMe	Fino a 20.000	Fino a 30
db.r6idn.large*	2	—	16	1 x 118 SSD NVMe	Fino a 20.000	Fino a 25
db.r6in: classi di istanza ottimizzata per la memoria con processori scalabili Intel Xeon di terza generazione e ottimizzazione di rete						
db.r6in.32xlarge	128	—	1,024	Solo ottimizzata per EBS	80.000	200

Classe istanza	VPCU	ECU	Memoria (GiB)	Archiviazione delle istanze (GiB)	Quantità max Larghezza di banda EBS (Mbps)	Larghezza di banda della rete (Gbps)
db.r6in.24xlarge	96	—	768	Solo ottimizzata per EBS	60.000	150
db.r6in.16xlarge	64	—	512	Solo ottimizzata per EBS	40.000	100
db.r6in.12xlarge	48	—	384	Solo ottimizzata per EBS	30.000	75
db.r6in.8xlarge	32	—	256	Solo ottimizzata per EBS	20.000	50
db.r6in.4xlarge*	16	—	128	Solo ottimizzata per EBS	Fino a 20.000	Fino a 50
db.r6in.2xlarge*	8	—	64	Solo ottimizzata per EBS	Fino a 20.000	Fino a 40
db.r6in.xlarge*	4	—	32	Solo ottimizzata per EBS	Fino a 20.000	Fino a 30
db.r6in.large*	2	—	16	Solo ottimizzata per EBS	Fino a 20.000	Fino a 25
db.r6id: classi di istanza per uso generico con processori Intel Xeon scalabili di terza generazione e archiviazione SSD						
db.r6id.32xlarge	128	—	1,024	4 x 1900 SSD NVMe	40.000	50
db.r6id.24xlarge	96	—	768	4 x 1425 SSD NVMe	30.000	37,5

Classe istanza	VPCU	ECU	Memoria (GiB)	Archiviazione delle istanze (GiB)	Quantità max Larghezza di banda EBS (Mbps)	Larghezza di banda della rete (Gbps)
db.r6id.16xlarge	64	—	512	2 x 1900 SSD NVMe	20.000	25
db.r6id.12xlarge	48	—	384	2 x 1425 SSD NVMe	15.000	18,75
db.r6id.8xlarge	32	—	256	1 x 1900 SSD NVMe	10.000	12,5
db.r6id.4xlarge*	16	—	128	1 x 950 SSD NVMe	Fino a 10.000	Fino a 12,5
db.r6id.2xlarge*	8	—	64	1 x 474 SSD NVMe	Fino a 10.000	Fino a 12,5
db.r6id.xlarge*	4	—	32	1 x 237 SSD NVMe	Fino a 10.000	Fino a 12,5
db.r6id.large*	2	—	16	1 x 118 SSD NVMe	Fino a 10.000	Fino a 12,5
db.r6i: classi di istanza ottimizzata per la memoria con processori Intel Xeon scalabili di terza generazione						
db.r6i.32xlarge	128	—	1,024	Solo ottimizzata per EBS	40.000	50
db.r6i.24xlarge	96	—	768	Solo ottimizzata per EBS	30.000	37,5
db.r6i.16xlarge	64	—	512	Solo ottimizzata per EBS	20.000	25

Classe istanza	VPCU	ECU	Memoria (GiB)	Archiviazione delle istanze (GiB)	Quantità max Larghezza di banda EBS (Mbps)	Larghezza di banda della rete (Gbps)
db.r6i.12xlarge	48	—	384	Solo ottimizzata per EBS	15.000	18,75
db.r6i.8xlarge	32	—	256	Solo ottimizzata per EBS	10.000	12,5
db.r6i.4xlarge*	16	—	128	Solo ottimizzata per EBS	Fino a 10.000	Fino a 12,5
db.r6i.2xlarge*	8	—	64	Solo ottimizzata per EBS	Fino a 10.000	Fino a 12,5
db.r6i.xlarge*	4	—	32	Solo ottimizzata per EBS	Fino a 10.000	Fino a 12,5
db.r6i.large*	2	—	16	Solo ottimizzata per EBS	Fino a 10.000	Fino a 12,5
db.r5d: classi di istanza ottimizzata per la memoria con processori Intel Xeon Platinum e archiviazione SSD						
db.r5d.24xlarge	96	347	768	4 x 900 SSD NVMe	19.000	25
db.r5d.16xlarge	64	264	512	4 x 600 SSD NVMe	13.600	20
db.r5d.12xlarge	48	173	384	2 x 900 SSD NVMe	9.500	10
db.r5d.8xlarge	32	132	256	2 x 600 SSD NVMe	6.800	10

Classe istanza	VPCU	ECU	Memoria (GiB)	Archiviazione delle istanze (GiB)	Quantità max Larghezza di banda EBS (Mbps)	Larghezza di banda della rete (Gbps)
db.r5d.4xlarge	16	71	128	2 x 300 SSD NVMe	4.750	Fino a 10
db.r5d.2xlarge	8	38	64	1 x 300 SSD NVMe	Fino a 4.750	Fino a 10
db.r5d.xlarge	4	19	32	1 x 150 SSD NVMe	Fino a 4.750	Fino a 10
db.r5d.large	2	10	16	1 x 75 SSD NVMe	Fino a 4.750	Fino a 10
db.r5b: classi di istanza ottimizzata per la memoria con processori Intel Xeon Platinum e archiviazione SSD						
db.r5b.24xlarge	96	347	768	Solo ottimizzata per EBS	60.000	25
db.r5b.16xlarge	64	264	512	Solo ottimizzata per EBS	40.000	20
db.r5b.12xlarge	48	173	384	Solo ottimizzata per EBS	30.000	10
db.r5b.8xlarge	32	132	256	Solo ottimizzata per EBS	20.000	10
db.r5b.4xlarge	16	71	128	Solo ottimizzata per EBS	10.000	Fino a 10
db.r5b.2xlarge*	8	38	64	Solo ottimizzata per EBS	Fino a 10.000	Fino a 10

Classe istanza	VPCU	ECU	Memoria (GiB)	Archiviazione delle istanze (GiB)	Quantità max Larghezza di banda EBS (Mbps)	Larghezza di banda della rete (Gbps)
db.r5b.xlarge*	4	19	32	Solo ottimizzata per EBS	Fino a 10.000	Fino a 10
db.r5b.large*	2	10	16	Solo ottimizzata per EBS	Fino a 10.000	Fino a 10
db.r5b: classi di istanza ottimizzata per la memoria Oracle preconfigurate per uso intensivo di memoria, archiviazione e I/O						
db.r5b.8xlarge.tpc2.mem3x	32	—	768	Solo ottimizzata per EBS	60.000	25
db.r5b.6xlarge.tpc2.mem4x	24	—	768	Solo ottimizzata per EBS	60.000	25
db.r5b.4xlarge.tpc2.mem4x	16	—	512	Solo ottimizzata per EBS	40.000	20
db.r5b.4xlarge.tpc2.mem3x	16	—	384	Solo ottimizzata per EBS	30.000	10
db.r5b.4xlarge.tpc2.mem2x	16	—	256	Solo ottimizzata per EBS	20.000	10
db.r5b.2xlarge.tpc2.mem8x	8	—	512	Solo ottimizzata per EBS	40.000	20
db.r5b.2xlarge.tpc2.mem4x	8	—	256	Solo ottimizzata per EBS	20.000	10
db.r5b.2xlarge.tpc1.mem2x	8	—	128	Solo ottimizzata per EBS	10.000	Fino a 10

Classe istanza	VPCU	ECU	Memoria (GiB)	Archiviazione delle istanze (GiB)	Quantità max Larghezza di banda EBS (Mbps)	Larghezza di banda della rete (Gbps)
db.r5b.xlarge.tpc2.mem4x	4	—	128	Solo ottimizzata per EBS	10.000	Fino a 10
db.r5b.xlarge.tpc2.mem2x	4	—	64	Solo ottimizzata per EBS	Fino a 10.000	Fino a 10
db.r5b.large.tpc1.mem2x	2	—	32	Solo ottimizzata per EBS	Fino a 10.000	Fino a 10
db.r5: classi di istanza ottimizzata per la memoria con processori Intel Xeon Platinum						
db.r5.24xlarge	96	347	768	Solo ottimizzata per EBS	19.000	25
db.r5.16xlarge	64	264	512	Solo ottimizzata per EBS	13.600	20
db.r5.12xlarge	48	173	384	Solo ottimizzata per EBS	9.500	12
db.r5.8xlarge	32	132	256	Solo ottimizzata per EBS	6.800	10
db.r5.4xlarge	16	71	128	Solo ottimizzata per EBS	4.750	Fino a 10
db.r5.2xlarge*	8	38	64	Solo ottimizzata per EBS	Fino a 4.750	Fino a 10
db.r5.xlarge*	4	19	32	Solo ottimizzata per EBS	Fino a 4.750	Fino a 10
db.r5.large	2	10	16	Solo ottimizzata per EBS	Fino a 4.750	Fino a 10

Classe istanza	VPCU	ECU	Memoria (GiB)	Archiviazione delle istanze (GiB)	Quantità max Larghezza di banda EBS (Mbps)	Larghezza di banda della rete (Gbps)
db.r5: classi di istanza ottimizzata per la memoria Oracle preconfigurate per uso intensivo di memoria, archiviazione e I/O						
db.r5.12xlarge.tpc2.mem2x	48	—	768	Solo ottimizzata per EBS	19.000	25
db.r5.8xlarge.tpc2.mem3x	32	—	768	Solo ottimizzata per EBS	19.000	25
db.r5.6xlarge.tpc2.mem4x	24	—	768	Solo ottimizzata per EBS	19.000	25
db.r5.4xlarge.tpc2.mem4x	16	—	512	Solo ottimizzata per EBS	13.600	20
db.r5.4xlarge.tpc2.mem3x	16	—	384	Solo ottimizzata per EBS	9.500	10
db.r5.4xlarge.tpc2.mem2x	16	—	256	Solo ottimizzata per EBS	6.800	10
db.r5.2xlarge.tpc2.mem8x	8	—	512	Solo ottimizzata per EBS	13.600	20
db.r5.2xlarge.tpc2.mem4x	8	—	256	Solo ottimizzata per EBS	6.800	10
db.r5.2xlarge.tpc1.mem2x	8	—	128	Solo ottimizzata per EBS	4.750	Fino a 10
db.r5.xlarge.tpc2.mem4x	4	—	128	Solo ottimizzata per EBS	4.750	Fino a 10

Classe istanza	VPCU	ECU	Memoria (GiB)	Archiviazione delle istanze (GiB)	Quantità max Larghezza di banda EBS (Mbps)	Larghezza di banda della rete (Gbps)
db.r5.xlarge.tpc2.mem2x	4	—	64	Solo ottimizzata per EBS	Fino a 4.750	Fino a 10
db.r5.large.tpc1.mem2x	2	—	32	Solo ottimizzata per EBS	Fino a 4.750	Fino a 10
db.r4: classi di istanza ottimizzata per la memoria con processori Intel Xeon scalabili						
db.r4.16xlarge	64	195	488	Solo ottimizzata per EBS	14.000	25
db.r4.8xlarge	32	99	244	Solo ottimizzata per EBS	7,000	10
db.r4.4xlarge	16	53	122	Solo ottimizzata per EBS	3,500	Fino a 10
db.r4.2xlarge	8	27	61	Solo ottimizzata per EBS	1.700	Fino a 10
db.r4.xlarge	4	13,5	30,5	Solo ottimizzata per EBS	850	Fino a 10
db.r4.large	2	7	15,25	Solo ottimizzata per EBS	425	Fino a 10
db.r3: classi di istanze ottimizzate per la memoria						
db.r3.8xlarge	32	104	244	Solo EBS	—	10
db.r3.4xlarge	16	52	122	Solo ottimizzata per EBS	2.000	Elevate
db.r3.2xlarge	8	26	61	Solo ottimizzata per EBS	1.000	Elevate

Classe istanza	VPCU	ECU	Memoria (GiB)	Archiviazione delle istanze (GiB)	Quantità max Larghezza di banda EBS (Mbps)	Larghezza di banda della rete (Gbps)
db.r3.xlarge	4	13	30,5	Solo ottimizzata per EBS	500	Moderate
db.r3.large	2	6,5	15,25	Solo ottimizzata per EBS	—	Moderata
db.c6gd — classi di istanze ottimizzate per il calcolo (solo per implementazioni di cluster DB Multi-AZ)						
db.c6gd.16xlarge	64	—	128	2 x 1900 SSD NVMe	19.000	25
db.c6gd.12xlarge	48	—	96	2 x 1425 SSD NVMe	13.500	20
db.c6gd.8xlarge	32	—	64	1 x 1900 SSD NVMe	9.000	12
db.c6gd.4xlarge	16	—	32	1 x 950 SSD NVMe	4.750	Fino a 10
db.c6gd.2xlarge	8	—	16	1 x 474 SSD NVMe	Fino a 4.750	Fino a 10
db.c6gd.xlarge	4	—	8	1 x 237 SSD NVMe	Fino a 4.750	Fino a 10
db.c6gd.large	2	—	4	1 x 118 SSD NVMe	Fino a 4.750	Fino a 10
db.c6gd.medium	1	—	2	1 x SSD NVMe da 59	Fino a 4.750	Fino a 10
db.t4g: classi di istanze a prestazioni incredibili con processori Graviton2 AWS						

Classe istanza	VPCU	ECU	Memoria (GiB)	Archiviazione delle istanze (GiB)	Quantità max Larghezza di banda EBS (Mbps)	Larghezza di banda della rete (Gbps)
db.t4g.2xlarge*	8	—	32	Solo ottimizzata per EBS	Fino a 2.780	Fino a 5
db.t4g.xlarge*	4	—	16	Solo ottimizzata per EBS	Fino a 2.780	Fino a 5
db.t4g.large*	2	—	8	Solo ottimizzata per EBS	Fino a 2.780	Fino a 5
db.t4g.medium*	2	—	4	Solo ottimizzata per EBS	Fino a 2.085	Fino a 5
db.t4g.small*	2	—	2	Solo ottimizzata per EBS	Fino a 2.085	Fino a 5
db.t4g.micro*	2	—	1	Solo ottimizzata per EBS	Fino a 2.085	Fino a 5
db.t3: classi di istanze a prestazioni espandibili						
db.t3.2xlarge*	8	Variab	32	Solo ottimizzata per EBS	Fino a 2.048	Fino a 5
db.t3.xlarge*	4	Variab	16	Solo ottimizzata per EBS	Fino a 2.048	Fino a 5
db.t3.large*	2	Variab	8	Solo ottimizzata per EBS	Fino a 2.048	Fino a 5
db.t3.medium*	2	Variab	4	Solo ottimizzata per EBS	Fino a 1.536	Fino a 5
db.t3.small*	2	Variab	2	Solo ottimizzata per EBS	Fino a 1.536	Fino a 5

Classe istanza	VPCU	ECU	Memoria (GiB)	Archiviazione delle istanze (GiB)	Quantità max Larghezza di banda EBS (Mbps)	Larghezza di banda della rete (Gbps)
db.t3.micro*	2	Variab	1	Solo ottimizzata per EBS	Fino a 1.536	Fino a 5
db.t2: classi di istanza a prestazioni espandibili						
db.t2.2xlarge	8	Variab	32	Solo EBS	—	Moderata
db.t2.xlarge	4	Variab	16	Solo EBS	—	Moderata
db.t2.large	2	Variab	8	Solo EBS	—	Moderata
db.t2.medium	2	Variab	4	Solo EBS	—	Moderata
db.t2.small	1	Variab	2	Solo EBS	—	Basse
db.t2.micro	1	Variab	1	Solo EBS	—	Bassa

* Questi tipi di classi di istanze DB possono offrire prestazioni massime per 30 minuti almeno una volta ogni 24 ore. Per ulteriori informazioni sulle prestazioni di base dei tipi di istanza EC2 sottostanti, consulta Istanze ottimizzate per Amazon EBS nella Guida per l'utente di Amazon EC2 per le istanze Linux.

** La classe di istanza database r3.8xlarge non ha una larghezza di banda EBS dedicata e quindi non offre l'ottimizzazione EBS. Per questa classe di istanza, il traffico di rete e il traffico Amazon EBS condividono la stessa interfaccia di rete da 10 gigabit.

Storage delle istanze di database Amazon RDS

Le istanze DB per Amazon RDS per Db2, MariaDB, MySQL, PostgreSQL, Oracle e Microsoft SQL Server utilizzano volumi Amazon Elastic Block Store (Amazon EBS) per lo storage di database e log.

In alcuni casi, il carico di lavoro del database potrebbe non essere in grado di ottenere il 100% della capacità di IOPS di cui hai effettuato il provisioning. Per ulteriori informazioni, consulta Fattori che influenzano le prestazioni di storage.

Per ulteriori informazioni sui prezzi dell'archiviazione delle istanze, consulta Prezzi di Amazon RDS.

Tipi di storage Amazon RDS

Amazon RDS offre tre tipi di storage: SSD Provisioned IOPS (noto anche come io1 e io2 Block Express), SSD generico (noto anche come gp2 e gp3) e magnetico (noto anche come standard). Questi tipi presentano caratteristiche di prestazioni e prezzi diversi, per permetterti di personalizzare le prestazioni e i costi di storage in base alle esigenze del carico di lavoro dei database. Puoi creare istanze DB RDS Db2, MySQL, MariaDB, Oracle e PostgreSQL con un massimo di 64 tebibyte (TiB) di storage. Puoi creare istanze database RDS per SQL Server con uno storage fino a 16 TiB. RDS for Db2 non supporta i tipi di archiviazione gp3 e magnetica.

Nell'elenco seguente vengono descritti brevemente i tre tipi di storage:

- SSD con capacità di IOPS allocata – L'archiviazione con capacità di IOPS allocata è progettata per soddisfare le esigenze dei carichi di lavoro con intenso traffico di I/O, in particolare i carichi di lavoro di database, che richiedono bassa latenza di I/O e velocità di trasmissione effettiva di I/O coerente. L'archiviazione con capacità di IOPS allocata è più adatta per gli ambienti di produzione.

  Per ulteriori informazioni sull'archiviazione con capacità di IOPS allocata, inclusi gli intervalli di dimensioni dell'archiviazione, consulta Storage SSD Provisioned IOPS.
- SSD per uso generico – I volumi SSD per uso generico offrono un'archiviazione conveniente ideale per un'ampia gamma di carichi di lavoro in esecuzione su istanze database di medie dimensioni. L'archiviazione per uso generico è più adatta per gli ambienti di sviluppo e test.

  Per ulteriori informazioni sullo storage SSD per scopi generici, inclusi gli intervalli di dimensioni dello storage, consulta Storage SSD per scopi generici.
- Magnetico – Amazon RDS supporta anche lo storage magnetico per garantire la compatibilità con le versioni precedenti. Per le nuove esigenze di storage, è consigliabile usare lo storage SSD per

scopi generici o SSD Provisioned IOPS. La quantità massima di storage consentita per le istanze DB sullo storage magnetico è di 3 TiB. Per ulteriori informazioni, consulta Archiviazione magnetica (precedente, non consigliata).

Quando si seleziona l'opzione "SSD per scopo generico" o "SSD con capacità di IOPS allocata", a seconda del motore selezionato e della quantità di archiviazione richiesta, Amazon RDS esegue automaticamente lo stripping su più volumi per migliorare le prestazioni, come mostrato nella tabella seguente.

Motore del database	Dimensioni dell'archiviazione di Amazon RDS	Numero di volumi assegnati
Db2	Meno di 400 GiB	1
Db2	400—65.536 GiB	4
MariaDB, MySQL e PostgreSQL	Meno di 400 GiB	1
MariaDB, MySQL e PostgreSQL	400—65.536 GiB	4
Oracle	Meno di 200 GiB	1
Oracle	200-65.536 GiB	4
SQL Server	Qualsiasi	1

Se si modifica un volume di tipo "SSD per scopo generico" o "SSD con capacità di IOPS allocata", tale volume passa attraverso una sequenza di stati. Mentre il volume è nello `optimizing` stato, le prestazioni del volume sono comprese tra le specifiche di configurazione di origine e di destinazione. Le prestazioni di volume transitorie non saranno inferiori alla più bassa delle due specifiche.

> **Important**
>
> Quando modifichi lo storage di un'istanza in modo che passi da un volume a quattro volumi o quando modifichi un'istanza utilizzando lo storage magnetico, Amazon RDS non utilizza la

funzionalità Elastic Volumes. Amazon RDS effettua invece il provisioning dei nuovi volumi e sposta in modo trasparente i dati dal vecchio volume a quelli nuovi. Questa operazione consuma una quantità significativa di IOPS e di velocità di trasmissione effettiva sia dei volumi vecchi che di quelli nuovi. A seconda delle dimensioni del volume e della quantità di carico di lavoro del database presente durante la modifica, questa operazione può consumare una quantità elevata di IOPS, aumentare in modo significativo la latenza di I/O e richiedere diverse ore per essere completata, mentre l'istanza RDS rimane nello stato. `Modifying`

Storage SSD Provisioned IOPS

Per qualsiasi applicazione di produzione che richieda prestazioni I/O veloci e affidabili, per l'archiviazione si consiglia di utilizzare l'opzione Capacità di IOPS allocata. Lo storage Provisioned IOPS è un tipo di storage che offre prestazioni prevedibili e latenza costantemente bassa. L'archiviazione di tipo Capacità di IOPS allocata è ottimizzata per i carichi di lavoro OLTP (Online Transaction Processing, elaborazione di transazioni online) che richiedono prestazioni costanti. L'opzione Provisioned IOPS permette di ottimizzare le prestazioni per questi carichi di lavoro.

Quando si crea un'istanza database, è necessario specificare la velocità IOPS e la dimensione del volume. Amazon RDS fornisce la frequenza IOPS per l'istanza database fino a quando non viene modificata.

Amazon RDS offre due tipi di storage SSD Provisioned IOPS: e. storage io2 Block Express (consigliato) storage io1 (generazione precedente)

storage io2 Block Express (consigliato)

Per carichi di lavoro a elevata intensità di I/O e sensibili alla latenza, è possibile utilizzare lo storage Provisioned IOPS SSD io2 Block Express per ottenere fino a 256.000 operazioni di I/O al secondo (IOPS). La velocità effettiva dei volumi io2 Block Express varia in base alla quantità di IOPS fornita per volume e alla dimensione delle operazioni di I/O eseguite.

Tutti i volumi RDS io2 basati sul sistema AWS Nitro sono volumi io2 Block Express e offrono una latenza media inferiore al millisecondo. Le istanze DB non basate sul sistema Nitro sono volumi io2. AWS

La tabella seguente mostra l'intervallo di Provisioned IOPS e il throughput massimo per ogni motore di database e intervallo di dimensioni di archiviazione.

Motore del database	Intervallo di dimensione di archiviazione	Intervallo di Provisioned IOPS	Velocità di trasmissione effettiva massima
Db2, MariaDB, MySQL e PostgreSQL	100-65.536 GiB	1.000–256.000 IOPS	4.000 MiB/s
Oracle	100-199 GiB	1.000—19.000 IOPS	4.000 MiB/s
Oracle	200-65.536 GiB	1.000–256.000 IOPS	4.000 MiB/s[1]
SQL Server	20—16.384 GiB	1.000–64.000 IOPS	4.000 MiB/s

> **Note**
>
> [1] Per Oracle, in determinate condizioni, ad esempio istanze DB di dimensioni molto grandi e letture di grandi dimensioni, è possibile riscontrare un throughput massimo molto più elevato.

Agli intervalli di dimensioni di archivio e IOPS si applicano i vincoli seguenti:

- Il rapporto tra IOPS e storage allocato (in GiB) non deve essere superiore a 1000:1. Per le istanze DB non basate sul sistema AWS Nitro, il rapporto è 500:1.
- È possibile eseguire il provisioning di IOPS massime con volumi di 256 GiB di dimensioni e superiori (1.000 IOPS x 256 GiB = 256.000 IOPS). Per le istanze DB non basate sul sistema AWS Nitro, gli IOPS massimi vengono raggiunti a 512 GiB (500 IOPS x 512 GiB = 256.000 IOPS).
- La velocità di trasmissione effettiva è scalabile in modo proporzionale fino a 0,256 MiB/s per capacità di IOPS allocata. Il throughput massimo di 4.000 MiB/s può essere raggiunto a 256.000 IOPS con una dimensione di I/O di 16 KiB e 16.000 IOPS o superiore con una dimensione di I/O di 256 KiB. Per le istanze DB non basate sul sistema AWS Nitro, è possibile ottenere un throughput massimo di 2.000 MiB/s a 128.000 IOPS con una dimensione di I/O di 16 KiB.
- Se utilizzi la scalabilità automatica dell'archiviazione, si applicano anche gli stessi rapporti tra IOPS e la soglia massima di archiviazione (in GiB). Per ulteriori informazioni sulla scalabilità automatica dell'archiviazione, consulta Gestione della capacità automaticamente con Auto Scaling dello storage Amazon RDS.

I volumi Amazon RDS io2 Block Express sono disponibili nei seguenti formati: Regioni AWS

- Asia Pacifico (Hong Kong)
- Asia Pacifico (Mumbai)
- Asia Pacifico (Seoul)
- Asia Pacifico (Singapore)
- Asia Pacifico (Sydney)
- Asia Pacifico (Tokyo)
- Canada (Centrale)
- Europa (Francoforte)
- Europa (Irlanda)
- Europe (London)
- Europa (Stoccolma)
- Medio Oriente (Bahrein)
- Stati Uniti orientali (Ohio)
- Stati Uniti orientali (Virginia settentrionale)
- Stati Uniti occidentali (California settentrionale)
- US West (Oregon)

storage io1 (generazione precedente)

Per carichi di lavoro con uso intensivo di I/O, puoi utilizzare l'archivio SSD io1 di capacità di IOPS allocata e ottenere fino a 256.000 operazioni di I/O al secondo (IOPS). Il throughput dei volumi io1 varia in base alla quantità di IOPS assegnati per volume e alla dimensione delle operazioni di I/O eseguite. Consigliamo di utilizzare lo storage io2 Block Express laddove disponibile.

La tabella seguente mostra l'intervallo di Provisioned IOPS e il throughput massimo per ogni motore di database e intervallo di dimensioni di archiviazione.

Motore del database	Intervallo di dimensione di archiviazione	Intervallo di Provisioned IOPS	Velocità di trasmissione effettiva massima
Db2, MariaDB, MySQL e PostgreSQL	100—399 GiB	1.000-19.950 IOPS	500 MiB/s
Db2, MariaDB, MySQL e PostgreSQL	400—65.536 GiB	1.000–256.000 IOPS	4.000 MiB/s
Oracle	100-199 GiB	1.000-9.950 IOPS	500 MiB/s
Oracle	200-65.536 GiB	1.000-256.000 IOPS[1]	4.000 MiB/s
SQL Server	20—16.384 GiB	1.000-64.000 IOPS[2]	1.000 MiB/s

Note

[1] Per Oracle, puoi effettuare il provisioning del massimo di 256.000 IOPS solo sul tipo di istanza r5b.
[2] Per SQL Server, il massimo di 64.000 IOPS è garantito solo sulle istanze basate su Nitro che si trovano sui tipi di istanza m5*, m6i, r5*, r6i e z1d. Altri tipi di istanze garantiscono prestazioni fino a 32.000 IOPS.

Agli intervalli di dimensioni di archivio e IOPS si applicano i vincoli seguenti:

- Il rapporto tra IOPS e archiviazione allocata (in GiB) deve essere compreso tra 1 e 50 su RDS for SQL Server e tra 0,5 e 50 su altri motori DB RDS.
- Se utilizzi la scalabilità automatica dell'archiviazione, si applicano anche gli stessi rapporti tra IOPS e la soglia massima di archiviazione (in GiB).

  Per ulteriori informazioni sulla scalabilità automatica dell'archiviazione, consulta Gestione della capacità automaticamente con Auto Scaling dello storage Amazon RDS.

Combinazione di storage Provisioned IOPS con implementazioni Multi-AZ o repliche di lettura

Per i casi d'uso OLTP, è consigliabile usare implementazioni Multi-AZ per una maggiore tolleranza ai guasti con lo storage Provisioned IOPS per prestazioni veloci e prevedibili.

Puoi anche utilizzare lo storage Provisioned IOPS con repliche di lettura per MySQL, MariaDB o PostgreSQL. Il tipo di storage per una replica di lettura è indipendente da quello dell'istanza database master. Ad esempio, puoi usare lo storage SSD per scopi generici per le repliche di lettura con un'istanza database master che usa lo storage SSD Provisioned IOPS, per ridurre i costi. Tuttavia, le prestazioni della replica di lettura in questo caso potrebbero differire da quelle di una configurazione in cui sia l'istanza DB principale che le repliche di lettura utilizzano lo storage Provisioned IOPS.

Costi dello storage Provisioned IOPS

Con lo storage Provisioned IOPS, paghi per le risorse di cui viene effettuato il provisioning, indipendentemente dal fatto che vegano usate o meno in un determinato mese.

Per ulteriori informazioni sui prezzi, consulta Prezzi di Amazon RDS.

Ottenere le migliori prestazioni dallo storage IOPS di Amazon RDS Provisioned

Se il carico di lavoro è limitato all'I/O, l'utilizzo dello storage Provisioned IOPS può aumentare il numero di richieste di I/O che il sistema può elaborare contemporaneamente. Una maggiore concorrenza permette una riduzione della latenza, perché le richieste di I/O restano in coda per meno tempo. Una latenza minore permette commit più rapidi nel database, migliorando così il tempo di risposta e permettendo un throughput del database maggiore.

Lo storage Provisioned IOPS offre un modo per riservare la capacità di I/O specificando IOPS. Come per qualsiasi altro attributo di capacità di un sistema, tuttavia, il throughput massimo sotto carico è limitato dalla risorsa consumata per prima. Tale risorsa può essere rappresentata da larghezza di banda di rete, CPU, memoria o risorse interne del database.

Storage SSD per scopi generici

Lo storage generico offre uno storage conveniente, accettabile per la maggior parte dei carichi di lavoro di database che non sono sensibili alla latenza o alle prestazioni.

> **Note**
>
> Le istanze DB che utilizzano lo storage General Purpose possono presentare una latenza molto più lunga rispetto alle istanze che utilizzano lo storage Provisioned IOPS. Se è necessaria un'istanza database con latenza minima dopo queste operazioni, è consigliabile utilizzare Storage SSD Provisioned IOPS.

Amazon RDS offre due tipi di storage generico: archiviazione gp3 (consigliata) egp2 storage (generazione precedente).

archiviazione gp3 (consigliata)

Utilizzando i volumi di archiviazione gp3 per uso generico, è possibile personalizzare le prestazioni di archiviazione indipendentemente dalla capacità di archiviazione. Le prestazioni di archiviazione sono la combinazione delle operazioni I/O al secondo (IOPS) e della velocità con cui il volume di archiviazione può eseguire operazioni di lettura e scrittura (velocità di trasmissione effettiva dell'archiviazione). Per i volumi di archiviazione gp3, Amazon RDS offre prestazioni di archiviazione di base di 3000 IOPS e 125 MiB/s.

Per ogni motore RDS DB ad eccezione di RDS per SQL Server, quando la dimensione di archiviazione per i volumi gp3 raggiunge una determinata soglia, le prestazioni di storage di base aumentano. Ciò è dovuto allo striping dei volumi, in cui l'archiviazione utilizza quattro volumi anziché uno. RDS per SQL Server non supporta lo striping dei volumi e quindi non ha un valore di soglia. Per i volumi con striping, Amazon RDS offre prestazioni di storage di base di 12.000 IOPS e 500 MiB/s.

Le prestazioni di archiviazione per i volumi gp3 sui motori di database Amazon RDS, inclusa la soglia, sono mostrate nella tabella seguente.

Motore database	Dimensioni dell'archiviazione	Prestazioni di archiviazione di base	Intervallo di Provisioned IOPS	Velocità di trasmissione effettiva dell'archiviazione allocata
Db2, MariaDB, MySQL e PostgreSQL	20—399 GiB	3.000 IOPS/125 MiB/s	N/D	N/D

Motore database	Dimensioni dell'archiviazione	Prestazioni di archiviazione di base	Intervallo di Provisioned IOPS	Velocità di trasmissione effettiva dell'archiviazione allocata
Db2, MariaDB, MySQL e PostgreSQL	400—65.536 GiB	12.000 IOPS/500 MiB/s	12.000 - 64.000 IOPS	500-4.000 MiB/s
Oracle	20—199 GiB	3.000 IOPS/125 MiB/s	N/D	N/D
Oracle	200-65.536 GiB	12.000 IOPS/500 MiB/s	12.000 - 64.000 IOPS	500-4.000 MiB/s
SQL Server	20—16.384 GiB	3.000 IOPS/125 MiB/s	3.000 - 16.000 IOPS	125-1.000 MiB/s

Per ogni motore di database ad eccezione di RDS per SQL Server, è possibile fornire IOPS e velocità di trasmissione effettiva per archiviazione aggiuntivi quando le dimensioni dell'archiviazione sono pari o superiori al valore di soglia. Per RDS per SQL Server, è possibile allocare IOPS e velocità di trasmissione effettiva per archiviazione aggiuntivi per qualsiasi dimensione di archiviazione disponibile. Per tutti i motori di database, paghi solo per le prestazioni di archiviazione allocata aggiuntive. Per ulteriori informazioni, consulta Prezzi di Amazon RDS.

Sebbene la capacità di IOPS allocata e la velocità di trasmissione effettiva per archiviazione aggiunti non dipendano dalle dimensioni dello archiviazione, sono correlate tra loro. Quando aumenti gli IOPS oltre 32.000 per MariadB e MySQL, il valore del throughput di archiviazione aumenta automaticamente da 500. MiBps Ad esempio, quando si imposta l'IOPS su 40.000 su RDS per MySQL, il throughput di archiviazione deve essere almeno 625. MiBps L'aumento automatico non si verifica per le istanze DB Db2, Oracle, PostgreSQL e SQL Server.

Per i cluster DB Multi-AZ, Amazon RDS imposta automaticamente il valore di throughput in base agli IOPS forniti. Non è possibile modificare il valore del throughput.

I valori delle prestazioni di archiviazione per i volumi gp3 su RDS presentano i seguenti vincoli:

- Il rapporto massimo tra la velocità di trasmissione effettiva per archiviazione e gli IOPS è 0,25 per tutti i motori di database supportati.
- Il rapporto minimo tra IOPS e archiviazione allocata (in GiB) è 0,5 su RDS per SQL Server. Non esiste un rapporto minimo per gli altri motori di database supportati.
- Il rapporto massimo tra IOPS per l'archiviazione allocata è 500 per tutti i motori di database supportati.
- Se utilizzi la scalabilità automatica dell'archiviazione, si applicano anche gli stessi rapporti tra IOPS e la soglia massima di archiviazione (in GiB).

  Per ulteriori informazioni sulla scalabilità automatica dell'archiviazione, consulta Gestione della capacità automaticamente con Auto Scaling dello storage Amazon RDS.

gp2 storage (generazione precedente)

Quando le tue applicazioni non richiedono prestazioni di archiviazione elevate, puoi utilizzare l'archiviazione gp2 SSD per uso generico. Le prestazioni di I/O di base per l'archiviazione gp2 sono di 3 IOPS per ogni GiB, con un minimo di 100 IOPS. Pertanto volumi di dimensioni maggiori offrono prestazioni migliori. Le prestazioni di base per un volume da 100 GiB sono ad esempio pari a 300 IOPS, per un volume da 1.000 GiB sono di 3.000 IOPS,

I singoli volumi gp2 con dimensioni inferiori a 1.000 GiB permettono inoltre il burst a fino a 3.000 IOPS per periodi di tempo prolungati. Il saldo dei crediti di I/O del volume determina le prestazioni di burst. Per una descrizione più dettagliata di come le prestazioni di base e il saldo del credito di I/O influiscono sulle prestazioni, consulta il post Understanding burst vs. baseline performance with Amazon RDS and gp2 sul Database Blog. AWS

Molti carichi di lavoro non esauriscono mai il saldo di burst. Alcuni carichi di lavoro, tuttavia, possono esaurire il saldo dei crediti di burst dello storage di 3.000 IOPS, quindi è consigliabile pianificare la capacità di storage in modo da soddisfare le esigenze dei carichi di lavoro.

Per volumi gp2 superiori a 4.000 GiB, le prestazioni di base sono superiori alle prestazioni burst. Per tali volumi, la burst è irrilevante perché le prestazioni di base sono migliori delle prestazioni di burst di 3.000 IOPS. Tuttavia, per le istanze database di determinati motori e dimensioni, l'archiviazione viene sottoposta a striping su quattro volumi e ciò garantisce una velocità di trasmissione effettiva quattro volte superiore a quella di base e quattro volte l'IOPS di burst di un singolo volume.

Le prestazioni di storage per volumi gp2 di varie dimensioni di storage sui motori Amazon RDS DB sono illustrate nella tabella seguente.

Motore database	Dimensioni di archiviazione RDS	Intervallo di IOPS di base	Intervallo di throughput di base	IOPS di burst
MariaDB, MySQL e PostgreSQL	5—399 GiB[1]	100-1197 IOPS	128-250 MiB/s	3.000
MariaDB, MySQL e PostgreSQL	400-1.335 GiB	1.200-4.005 IOPS	500-1.000 MiB/s	12.000
MariaDB, MySQL e PostgreSQL	1.336—3.999 GiB	4008-11.997 IOPS	1.000 MiB/s	12.000
MariaDB, MySQL e PostgreSQL	4.000-65.536 GiB	12.000-64.000 IOPS	1.000 MiB/s	N/M[2]
Oracle	20—199 GiB	100-597 IOPS	128-250 MiB/s	3.000
Oracle	200-1.335 GiB	600-4.005 IOPS	500-1.000 MiB/s	12.000
Oracle	1.336—3.999 GiB	4008-11.997 IOPS	1.000 MiB/s	12.000
Oracle	4.000-65.536 GiB	12.000-64.000 IOPS	1.000 MiB/s	N/M[2]
SQL Server	20—333 GiB	100-999 IOPS	128-250 MiB/s	3.000
SQL Server	334-999 GiB	1.002-2.997 IOPS	250 MiB/s	3.000
SQL Server	1.000—16.384 GiB	3.000-16.000 IOPS	250 MiB/s	N/M[2]

> **Note**
>
> ¹ Utilizzando AWS Management Console, è possibile creare istanze DB con una dimensione di storage minima di 5 GiB nel livello Free per le classi di istanze DB db.t3.micro e db.t4g.micro. Altrimenti, la dimensione minima di archiviazione è di 20 GiB. Questa limitazione non si applica all'API AWS CLI e RDS.
> ² Le prestazioni di base del volume superano le prestazioni massime di raffica.

Confronto dei tipi di archiviazione unità di memoria a stato solido (SSD)

La tabella seguente mostra i casi d'uso e le caratteristiche di prestazioni dei volumi di archiviazione SSD utilizzati da Amazon RDS.

Caratteristica	IOPS fornito (io2 Block Express)	Capacità di IOPS allocata (io1)	Uso generico (gp3)	Uso generico (gp2)
Descrizione	Prestazioni più elevate all'interno del portafoglio di storage RDS (IOPS, throughput, latenza) Progettato per carichi di lavoro transazionali sensibili alla latenza	Prestazioni di archiviazione coerenti (IOPS, velocità di trasmissione effettiva, latenza) Progettato per carichi di lavoro transazionali sensibili alla latenza	Flessibilità nell'allocazione di archiviazione, IOPS e velocità di trasmissione effettiva in modo indipendente Bilancia prezzi e prestazioni per un'ampia gamma di carichi di lavoro transazionali	Fornisce IOPS espandibili Bilancia prezzi e prestazioni per un'ampia gamma di carichi di lavoro transazionali
Casi d'uso	Carichi di lavoro transazionali critici per l'azienda che richiedono una latenza inferiore	Carichi di lavoro transazionali che richiedono prestazioni IOPS sostenute fino a 256.000 IOPS	Ampia gamma di carichi di lavoro eseguiti su database relazionali di medie dimension	Ampia gamma di carichi di lavoro eseguiti su database relazionali di medie dimension

Caratteristica	IOPS fornito (io2 Block Express)	Capacità di IOPS allocata (io1)	Uso generico (gp3)	Uso generico (gp2)
	al millisecondo e prestazioni IOPS sostenute fino a 256.000 IOPS		i in ambienti di sviluppo/test	i in ambienti di sviluppo/test
Latenza	Meno di millisecondo, forniti costantemente il 99,9% delle volte	Millisecondo a una cifra singola, fornito costantemente il 99,9% delle volte	Millisecondo a una cifra singola, fornito costantemente il 99% delle volte	Millisecondo a una cifra singola, fornito costantemente il 99% delle volte
Volume size (Dimensione dei volumi)	100-65.536 GiB (16.384 GiB su RDS per SQL Server)	100—65.536 GiB (20—16.384 GiB su RDS per SQL Server)	20—65.536 GiB (16.384 GiB su RDS per SQL Server)	20—65.536 GiB (16.384 GiB su RDS per SQL Server)

Caratteristica	IOPS fornito (io2 Block Express)	Capacità di IOPS allocata (io1)	Uso generico (gp3)	Uso generico (gp2)
Numero massimo di IOPS	256.000 (64.000 su RDS per SQL Server)	256.000 (64.000 su RDS per SQL Server)	64.000 (16.000 su RDS per SQL Server)	64.000 (16.000 su RDS per SQL Server) **Note** Non è possibile impostare la capacità di IOPS allocata direttamente sull'archiviazione gp2. La capacità di IOPS varia in base alla dimensione dell'archiviazione allocata.

Caratteristica	IOPS fornito (io2 Block Express)	Capacità di IOPS allocata (io1)	Uso generico (gp3)	Uso generico (gp2)
Velocità di trasmissione effettiva massima	Scale basate sulla capacità di IOPS allocata fino a 4.000 MB/s La velocità di trasmissione effettiva è scalabile in modo proporzionale fino a 0,256 MiB/s per capacità di IOPS allocata. Il throughput massimo di 4.000 MiB/s può essere raggiunto a 256.000 IOPS con una dimensione di I/O di 16 KiB e 16.000 IOPS o superiore con una dimensione di I/O di 256 KiB. Per le istanze non basate sul sistema AWS Nitro, è possibile ottenere un throughput	Scale basate sulla capacità di IOPS allocata fino a 4.000 MB/s	Alloca ulteriore velocità di trasmissione effettiva fino a 4.000 MB/s (1000 MB/s su RDS per SQL Server)	1000 MB/s (250 MB/s su RDS per SQL Server)

Caratteristica	IOPS fornito (io2 Block Express)	Capacità di IOPS allocata (io1)	Uso generico (gp3)	Uso generico (gp2)
	massimo di 2.000 MiB/s a 128.000 IOPS con una dimensione di I/O di 16 KiB.			
AWS CLI e nome dell'API RDS	io2	io1	gp3	gp2

Archiviazione magnetica (precedente, non consigliata)

Amazon RDS supporta anche lo storage magnetico per garantire la compatibilità con le versioni precedenti. Per le nuove esigenze di storage, è consigliabile usare lo storage SSD per scopi generici o SSD Provisioned IOPS. Di seguito sono elencate le limitazioni per lo storage magnetico:

- Non permette il dimensionamento dello storage quando si usa il motore di database di SQL Server.
- Non consente la conversione in un tipo di archiviazione diverso quando si utilizza il motore di database SQL Server.
- Non supporta il dimensionamento automatico dello storage.
- Non supporta i volumi elastici.
- Prevede una dimensione massima di 3 TiB.
- Prevede un limite massimo di 1.000 IOPS.

Volume di registro dedicato (DLV)

Puoi utilizzare un volume di log dedicato (DLV) per un'istanza DB che utilizza lo storage Provisioned IOPS (PIOPS) utilizzando la console Amazon RDS AWS CLI o l'API Amazon RDS. Un DLV sposta i log delle transazioni del database PostgreSQL e i redo log e i log binari di MySQL/MariaDB su un volume di archiviazione separato dal volume contenente le tabelle del database. Un DLV rende il log di scrittura delle transazioni più efficiente e coerente. I DLV sono ideali per database con

archiviazione allocata di grandi dimensioni, requisiti di I/O al secondo (IOPS) elevati o carichi di lavoro sensibili alla latenza.

I DLV sono supportati per lo storage PIOPS (io1 e io2 Block Express) e vengono creati con una dimensione fissa di 1.000 GiB e 3.000 Provisioned IOPS.

> **Note**
>
> I DLV non sono supportati per lo storage generico (gp2 e gp3).

Amazon RDS supporta tutti i DLV Regioni AWS per le seguenti versioni:

- MariaDB 10.6.7 e versioni successive alla 10
- MySQL 8.0.28 e versioni successive alla 8
- PostgreSQL 13.10 e versioni successive 13 versioni, 14.7 e successive 14 versioni, 15.2 e successive 15 versioni e 16.1 e versioni successive 16

RDS supporta i DLV con le implementazioni multi-AZ. Quando modifichi o crei un'istanza Multi-AZ, viene creato un DLV sia per l'istanza primaria che per quella secondaria.

RDS supporta i DLV con le repliche di lettura. Se l'istanza database primaria ha un DLV abilitato, anche tutte le repliche di lettura create dopo aver abilitato il DLV avranno un DLV. Tutte le repliche di lettura create prima del passaggio al DLV non saranno abilitate a meno che non vengano modificate esplicitamente in tal senso. Si consiglia inoltre di modificare manualmente tutte le repliche di lettura collegate a un'istanza primaria prima dell'abilitazione del DLV per includere un DLV.

Dopo aver modificato l'impostazione DLV per un'istanza database, è necessario riavviare l'istanza database.

Per informazioni sull'attivazione di un DLV, consulta. Utilizzo di un volume di log dedicato (DLV)

Monitoraggio delle prestazioni di storage

Amazon RDS fornisce diversi parametri che è possibile usare per determinare le prestazioni dell'istanza database. Puoi visualizzare i parametri nella pagina di riepilogo per l'istanza nella console di gestione Amazon RDS. Puoi anche utilizzare Amazon CloudWatch per monitorare questi parametri. Per ulteriori informazioni, consulta Visualizzazione dei parametri nella console Amazon

RDS. Il monitoraggio avanzato fornisce parametri di I/O più dettagliati. Per altre informazioni, consulta Monitoraggio dei parametri del sistema operativo con il monitoraggio avanzato.

I parametri seguenti sono utili per il monitoraggio dello storage per l'istanza database:

- IOPS – Numero di operazioni di I/O completate ogni secondo. Questo parametro viene segnalato indicando la media IOPS per un determinato intervallo di tempo. I report di Amazon RDS leggono e scrivono IOPS separatamente a intervalli di 1 minuto. Il valore di IOPS totali rappresenta la somma delle quantità di IOPS di lettura e di scrittura. I valori di IOPS tipici vanno da zero a decine di migliaia al secondo.
- Latenza – Tempo trascorso tra l'invio di una richiesta di I/O e il suo completamento. Questo parametro viene segnalato indicando la latenza media per un determinato intervallo di tempo. Amazon RDS indica separatamente la latenza di lettura e scrittura, in intervalli di 1 minuto. I valori tipici della latenza sono in millisecondi (ms).
- Throughput – Numero di byte al secondo trasferiti da o verso il disco. Questo parametro viene segnalato come il throughput medio per un determinato intervallo di tempo. Amazon RDS riporta il throughput di lettura e scrittura separatamente a intervalli di 1 minuto utilizzando unità di byte al secondo (B/s). I valori tipici per il throughput vanno da zero alla larghezza di banda massima del canale di I/O.
- Profondità della coda – Numero di richieste di I/O in coda in attesa di essere elaborate. Si tratta delle richieste di I/O che sono state inviate dall'applicazione ma non sono state trasmesse al dispositivo perché il dispositivo sta elaborando altre richieste di I/O. Il tempo speso in attesa in coda è un componente della latenza e del tempo di elaborazione (non disponibile come parametro). Questo parametro viene segnalato indicando la profondità della coda media per un determinato intervallo di tempo. Amazon RDS riporta la profondità della coda a intervalli di 1 minuto. I valori tipici per la profondità della coda vanno da zero ad alcune centinaia.

I valori di IOPS misurati sono indipendenti dalla dimensione della singola operazione di I/O. Ciò significa che quando misuri le prestazioni di I/O, assicurati di considerare la velocità di trasmissione effettiva dell'istanza e non semplicemente il numero di operazioni di I/O.

Fattori che influenzano le prestazioni di storage

Le attività di sistema, il carico di lavoro del database e la classe d’istanza DB possono influenzare le prestazioni di storage.

Attività di sistema

Le attività seguenti correlate al sistema utilizzano capacità di I/O e possono ridurre le prestazioni dell'istanza DB mentre sono in esecuzione:

- Creazione della copia di standby Multi-AZ
- Creazione della replica di lettura
- Modifica dei tipi di storage

Carico di lavoro del database

In alcuni casi, la progettazione del database o dell'applicazione provoca problemi di concorrenza, blocchi o altre forme di conflitto nel database. In questi casi, potrebbe non essere possibile usare direttamente tutta la larghezza di banda di cui è stato effettuato il provisioning. Potrebbero inoltre verificarsi le situazioni seguenti correlate ai carichi di lavoro:

- Il limite di throughput del tipo di istanza sottostante viene raggiunto.
- La profondità della coda è costantemente inferiore a 1 perché l'applicazione non effettua un numero sufficiente di operazioni di I/O.
- Si verificano conflitti di query nel database anche se vi è capacità di I/O non utilizzata.

In alcuni casi, non esiste una risorsa di sistema che ha raggiunto il limite o si è avvicinata a esso e l'aggiunta di thread non aumenta la velocità delle transazioni del database. In questi casi, il collo di bottiglia è molto probabilmente contesa nel database. Le forme più comuni sono i conflitti di blocco di riga e blocco di pagina di indice, ma ci sono numerose altre possibilità. Se si verifica questa situazione, rivolgiti a un esperto di regolazione delle prestazioni del database.

DB instance class (Classe istanza database)

Per ottenere il massimo delle prestazioni dall'istanza DB Amazon RDS, scegli un tipo di istanza di generazione corrente con larghezza di banda sufficiente per supportare il tipo di storage. Puoi ad esempio scegliere istanze ottimizzate per Amazon EBS–istanze con connettività di rete a 10 gigabit.

> **Important**
>
> A seconda della classe di istanza che stai utilizzando, potresti vedere prestazioni IOPS inferiori rispetto al massimo che RDS consente di eseguire il provisioning. Per informazioni

specifiche sulle prestazioni IOPS per le classi di istanza database, consulta Istanze ottimizzate per Amazon EBS nella Guida per l'utente di Amazon EC2. Si consiglia di determinare il numero massimo di IOPS per la classe di istanza prima di impostare un valore IOPS con provisioning per l'istanza database.

Per ottenere le prestazioni migliori, ti consigliamo di usare le istanze di ultima generazione. Le istanze database della generazione precedente possono anche avere un limite di storage inferiore.

Alcuni file system a 32 bit più vecchi potrebbero avere capacità di archiviazione inferiori. Per determinare la capacità di storage della tua istanza DB, puoi utilizzare il comando -modifications. describe-valid-db-instance AWS CLI

La lista seguente mostra lo storage massimo in base a cui la maggior parte delle classi di istanza database possono essere ridimensionate per ogni motore di database:

- Db2 — 64 TiB
- MariaDB - 64 TiB
- Microsoft SQL Server - 16 TiB
- MySQL - 64 TiB
- Oracle - 64 TiB
- PostgreSQL - 64 TiB

Nella tabella riportata di seguito sono elencate alcune eccezioni relative allo spazio di archiviazione massimo (in TiB). Tutte le istanze database RDS per Microsoft SQL Server DB hanno uno spazio di archiviazione massimo di 16 TiB, quindi non ci sono voci per SQL Server.

Classe di istanza	Db2	MariaDB	MySQL	Oracle	PostgreSQL
db.m3: classi di istanza standard					
db.t4g: classi di istanza a prestazioni espandibili					
db.t4g.medium	N/D	16	16	N/D	32
db.t4g.small	N/D	16	16	N/D	16

Classe di istanza	Db2	MariaDB	MySQL	Oracle	PostgreSQL
db.t4g.micro	N/D	6	6	N/D	6
db.t3: classi di istanza a prestazioni espandibili					
db.t3.medium	32	16	16	32	32
db.t3.small	32	16	16	32	16
db.t3.micro	N/D	6	6	32	6
db.t2: classi di istanza a prestazioni espandibili					

Per ulteriori dettagli sulle classi di istanza supportate, consultare Istanze database di generazioni precedenti.

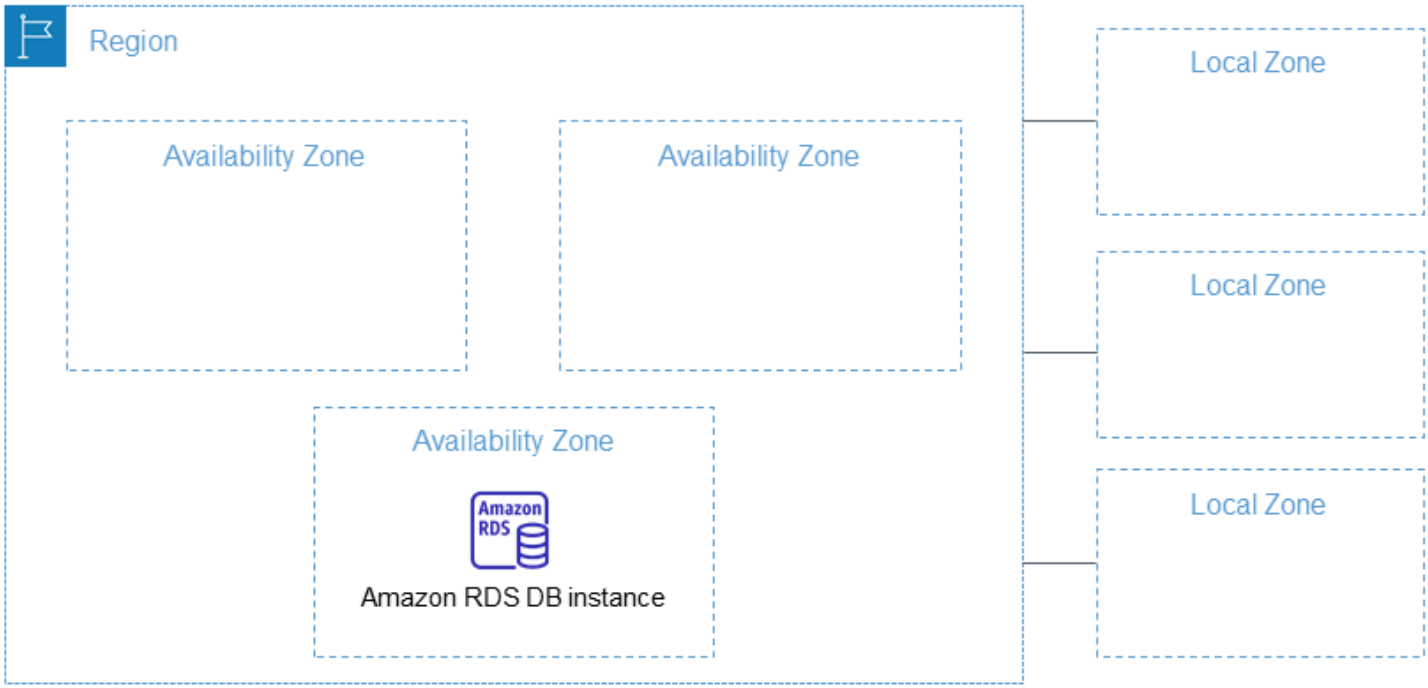
Regioni, zone di disponibilità e Local Zones

Le risorse di cloud computing Amazon sono ospitate in più ubicazioni in tutto il mondo. Queste località sono composte da AWS Regioni, Zone di disponibilità e Local Zones. Ciascuna regione AWS è un'area geografica distinta. Ogni AWS regione ha più sedi isolate note come zone di disponibilità.

> **Note**
>
> Per informazioni su come trovare le zone di disponibilità per una AWS regione, consulta Descrivi le tue zone di disponibilità nella documentazione di Amazon EC2.

Le Local Zones offrono la possibilità di collocare risorse, ad esempio calcolo e archiviazione, in più posizioni più vicine agli utenti finali. Amazon RDS consente di inserire le risorse, ad esempio le istanze database, e i dati in più località. Le risorse non vengono replicate tra le AWS regioni a meno che tu non lo faccia in modo specifico.

Amazon gestisce state-of-the-art data center ad alta disponibilità. Sebbene rari, possono verificarsi dei guasti che influiscano sulla disponibilità delle istanze database che si trovano nello stesso luogo. Se tutte le istanze database sono ospitate in un singolo luogo interessato da questo errore, nessuna delle istanze database sarà disponibile.



È importante ricordare che ogni AWS regione è completamente indipendente. Qualsiasi attività Amazon RDS avviata (ad esempio, la creazione di istanze di database o l'elenco delle istanze di database disponibili) viene eseguita solo nella regione predefinita corrente. AWS La AWS regione predefinita può essere modificata nella console o impostando la variabile di ambiente. `AWS_DEFAULT_REGION` Oppure può essere sovrascritta utilizzando il `--region` parametro con AWS Command Line Interface ()AWS CLI. Per ulteriori informazioni, consulta la pagina relativa alla configurazione dell' AWS Command Line Interface, in particolare le sezioni relative alle variabili di ambiente e opzioni della riga di comando.

Amazon RDS supporta AWS regioni speciali chiamate AWS GovCloud (US). sono progettate per consentire ai clienti e agli enti governativi degli Stati Uniti di spostare nel cloud i carichi di lavoro più sensibili. Le regioni AWS GovCloud (US) fanno riferimento ai requisiti normativi e di compliance specifici del governo degli Stati Uniti. Per ulteriori informazioni, consulta Cos'è AWS GovCloud (US)?

Per creare o utilizzare un'istanza database Amazon RDS in una AWS regione specifica, utilizza l'endpoint di servizio regionale corrispondente.

AWS Regioni

Ogni AWS regione è progettata per essere isolata dalle altre AWS regioni. Questo progetto permette di raggiungere la maggiore stabilità e tolleranza ai guasti possibile.

Quando si visualizzano le risorse, vengono visualizzate solo le risorse legate alla AWS regione specificata. Questo perché AWS le regioni sono isolate l'una dall'altra e non replichiamo automaticamente le risorse tra le AWS regioni.

Disponibilità nelle regioni

La tabella seguente mostra le AWS regioni in cui Amazon RDS è attualmente disponibile e l'endpoint per ciascuna regione.

Nome della regione	Regione	Endpoint	Protocollo	
US East (Ohio)	us-east-2	rds.us-east-2.amazonaws.com	HTTPS	
		rds-fips.us-east-2.api.aws	HTTPS	
		rds.us-east-2.api.aws	HTTPS	

Nome della regione	Regione	Endpoint	Protocollo
		rds-fips.us-east-2.amazonaws.com	HTTPS
US East (N. Virginia)	us-east-1	rds.us-east-1.amazonaws.com	HTTPS
		rds-fips.us-east-1.api.aws	HTTPS
		rds-fips.us-east-1.amazonaws.com	HTTPS
		rds.us-east-1.api.aws	HTTPS
Stati Uniti occidentali (California settentrionale)	us-west-1	rds.us-west-1.amazonaws.com	HTTPS
		rds.us-west-1.api.aws	HTTPS
		rds-fips.us-west-1.amazonaws.com	HTTPS
		rds-fips.us-west-1.api.aws	HTTPS
US West (Oregon)	us-west-2	rds.us-west-2.amazonaws.com	HTTPS
		rds-fips.us-west-2.amazonaws.com	HTTPS
		rds.us-west-2.api.aws	HTTPS
		rds-fips.us-west-2.api.aws	HTTPS
Africa (Cape Town)	af-south-1	rds.af-south-1.amazonaws.com	HTTPS
		rds.af-south-1.api.aws	HTTPS
Asia Pacifico (Hong Kong)	ap-east-1	rds.ap-east-1.amazonaws.com	HTTPS
		rds.ap-east-1.api.aws	HTTPS

Nome della regione	Regione	Endpoint	Protocollo	
Asia Pacific (Hyderabad)	ap-south-2	rds.ap-south-2.amazonaws.com rds.ap-south-2.api.aws	HTTPS HTTPS	
Asia Pacifico (Giacarta)	ap-southeast-3	rds.ap-southeast-3.amazonaws.com rds.ap-southeast-3.api.aws	HTTPS HTTPS	
Asia Pacifico (Melbourne)	ap-southeast-4	rds.ap-southeast-4.amazonaws.com rds.ap-southeast-4.api.aws	HTTPS HTTPS	
Asia Pacifico (Mumbai)	ap-south-1	rds.ap-south-1.amazonaws.com rds.ap-south-1.api.aws	HTTPS HTTPS	
Asia Pacifico (Osaka-Locale)	ap-northeast-3	rds.ap-northeast-3.amazonaws.com rds.ap-northeast-3.api.aws	HTTPS HTTPS	
Asia Pacifico (Seoul)	ap-northeast-2	rds.ap-northeast-2.amazonaws.com rds.ap-northeast-2.api.aws	HTTPS HTTPS	
Asia Pacifico (Singapore)	ap-southeast-1	rds.ap-southeast-1.amazonaws.com rds.ap-southeast-1.api.aws	HTTPS HTTPS	

Nome della regione	Regione	Endpoint	Protocollo	
Asia Pacifico (Sydney)	ap-southeast-2	rds.ap-southeast-2.amazonaws.com	HTTPS	
		rds.ap-southeast-2.api.aws	HTTPS	
Asia Pacifico (Tokyo)	ap-northeast-1	rds.ap-northeast-1.amazonaws.com	HTTPS	
		rds.ap-northeast-1.api.aws	HTTPS	
Canada (Centrale)	ca-central-1	rds.ca-central-1.amazonaws.com	HTTPS	
		rds.ca-central-1.api.aws	HTTPS	
		rds-fips.ca-central-1.api.aws	HTTPS	
		rds-fips.ca-central-1.amazonaws.com	HTTPS	
Canada occidentale (Calgary)	ca-west-1	rds.ca-west-1.amazonaws.com	HTTPS	
		rds-fips.ca-west-1.amazonaws.com	HTTPS	
Europa (Francoforte)	eu-central-1	rds.eu-central-1.amazonaws.com	HTTPS	
		rds.eu-central-1.api.aws	HTTPS	
Europa (Irlanda)	eu-west-1	rds.eu-west-1.amazonaws.com	HTTPS	
		rds.eu-west-1.api.aws	HTTPS	
Europa (Londra)	eu-west-2	rds.eu-west-2.amazonaws.com	HTTPS	
		rds.eu-west-2.api.aws	HTTPS	
Europa (Milano)	eu-south-1	rds.eu-south-1.amazonaws.com	HTTPS	
		rds.eu-south-1.api.aws	HTTPS	

Nome della regione	Regione	Endpoint	Protocollo
Europa (Parigi)	eu-west-3	rds.eu-west-3.amazonaws.com rds.eu-west-3.api.aws	HTTPS HTTPS
Europa (Spagna)	eu-south-2	rds.eu-south-2.amazonaws.com rds.eu-south-2.api.aws	HTTPS HTTPS
Europa (Stoccolma)	eu-north-1	rds.eu-north-1.amazonaws.com rds.eu-north-1.api.aws	HTTPS HTTPS
Europa (Zurigo)	eu-central-2	rds.eu-central-2.amazonaws.com rds.eu-central-2.api.aws	HTTPS HTTPS
Israele (Tel Aviv)	il-central-1	rds.il-central-1.amazonaws.com rds.il-central-1.api.aws	HTTPS HTTPS
Medio Oriente (Bahrein)	me-south-1	rds.me-south-1.amazonaws.com rds.me-south-1.api.aws	HTTPS HTTPS
Medio Oriente (Emirati Arabi Uniti)	me-central-1	rds.me-central-1.amazonaws.com rds.me-central-1.api.aws	HTTPS HTTPS
Sud America (São Paulo)	sa-east-1	rds.sa-east-1.amazonaws.com rds.sa-east-1.api.aws	HTTPS HTTPS

Nome della regione	Regione	Endpoint	Protocollo	
AWS GovCloud (Stati Uniti orientali)	us-gov-east-1	rds.us-gov-east-1.amazonaws.com rds.us-gov-east-1.api.aws	HTTPS HTTPS	
AWS GovCloud (Stati Uniti occidentali)	us-gov-west-1	rds.us-gov-west-1.amazonaws.com rds.us-gov-west-1.api.aws	HTTPS HTTPS	

Se non specifichi un endpoint in modo esplicito, l'endpoint Stati Uniti occidentali (Oregon) viene utilizzato per impostazione predefinita.

Quando lavori con un'istanza DB utilizzando le operazioni AWS CLI o API, assicurati di specificare il relativo endpoint regionale.

Zone di disponibilità

Quando crei un'istanza database, puoi scegliere una zona di disponibilità o fare in modo che Amazon RDS ne scelga una per te. Una zona di disponibilità è rappresentata da un codice AWS regionale seguito da una lettera identificativa (ad esempio,`us-east-1a`).

Utilizza il comando describe-availability-zonesAmazon EC2 come segue per descrivere le zone di disponibilità all'interno della regione specificata che sono abilitate per il tuo account.

```
aws ec2 describe-availability-zones --region region-name
```

Ad esempio, per descrivere le zone di disponibilità all'interno della regione Stati Uniti orientali (Virginia settentrionale) (us-east-1) abilitate per il tuo account, esegui il comando seguente:

```
aws ec2 describe-availability-zones --region us-east-1
```

Non è possibile scegliere le zone di disponibilità per le istanze database primarie e secondarie in una implementazione database Multi-AZ. Amazon RDS li sceglie per te in modo casuale. Per ulteriori informazioni sulle implementazioni Multi-AZ, consulta Configurazione e gestione di un'implementazione multi-AZ.

> **Note**
>
> La selezione casuale delle zone di disponibilità tramite RDS non garantisce una distribuzione uniforme delle istanze database tra le zone di disponibilità all'interno di un singolo account o gruppo di sottoreti del database. È possibile richiedere una AZ specifica quando si crea o si modifica un'istanza Single-AZ singolo ed è possibile utilizzare gruppi di sottoreti database più specifici per le istanze Multi-AZ. Per ulteriori informazioni, consulta Creazione di un'istanza database Amazon RDS e Modifica di un'istanza database Amazon RDS.

Zone locali

Una zona locale è un'estensione di una AWS regione geograficamente vicina ai tuoi utenti. Puoi estendere qualsiasi VPC della Regione AWS padre nelle zone locali. A tale scopo, crea una nuova sottorete e assegnala alla zona locale AWS . Quando si crea una sottorete in una zona locale, il VPC viene esteso anche a tale zona locale. La sottorete nell'area locale funziona allo stesso modo delle altre sottoreti nel VPC.

Quando si crea un'istanza database, è possibile scegliere una sottorete in un'area locale. Le Local Zones hanno le loro connessioni a Internet e supportano AWS Direct Connect. Pertanto, le risorse create in un'area locale possono servire gli utenti locali con comunicazioni a latenza molto bassa. Per ulteriori informazioni, consulta AWS Local Zones.

Una zona locale è rappresentata da un codice AWS regionale seguito da un identificatore che indica la posizione, ad esempio. `us-west-2-lax-1a`

> **Note**
>
> Una zona locale non può essere inclusa in una implementazione Multi-AZ.

Per utilizzare una zona locale

1. Attivare l a zona locale nella console Amazon EC2.

Per ulteriori informazioni, vedere Abilitazione delle Local Zones nella Guida per l'utente di Amazon EC2 per le istanze Linux.

2. Creare una sottorete nella zona locale.

   Per ulteriori informazioni, consulta Creazione di una sottorete nel VPC nella Guida per l'utente di Amazon VPC.

3. Creare un gruppo di sottoreti di database nella zona locale.

   Quando si crea un gruppo di sottoreti di database, scegliere il gruppo Zona di disponibilità per la zona locale.

   Per ulteriori informazioni, consulta Creazione di un'istanza database in un VPC.

4. Creare un'istanza database che utilizza il gruppo di sottoreti di database nella zona locale.

   Per ulteriori informazioni, consulta Creazione di un'istanza database Amazon RDS.

> **Important**
>
> Attualmente, l'unica zona AWS locale in cui Amazon RDS è disponibile è Los Angeles, nella regione Stati Uniti occidentali (Oregon).

Funzionalità supportate in Amazon RDS by Regione AWS e DB engine

Il supporto per le caratteristiche e le opzioni di Amazon RDS varia Regioni AWS a seconda delle versioni specifiche di ciascun motore DB. Per individuare il supporto e la disponibilità della versione del motore di database RDS in una determinata Regione AWS, puoi utilizzare le sezioni seguenti.

Le funzionalità di Amazon RDS sono diverse dalle funzionalità e dalle opzioni native del motore. Per ulteriori informazioni sulle funzionalità e sulle opzioni native del motore, consulta la pagina relativa alle funzionalità native del motore.

Regioni e motori DB supportati

- Convenzioni tabella
- Riferimento rapido alle funzionalità
- Regioni e motori DB supportati per le distribuzioni Blue/Green di Amazon RDS
- Regioni e motori DB supportati per backup automatici tra regioni in Amazon RDS
- Regioni e motori DB supportati per repliche di lettura interregionali in Amazon RDS
- Regioni e motori DB supportati per i flussi di attività del database in Amazon RDS
- Regioni e motori DB supportati per la modalità dual-stack in Amazon RDS
- Regioni e motori DB supportati per l'esportazione di snapshot in S3 in Amazon RDS
- Regioni e motori DB supportati per l'autenticazione del database IAM in Amazon RDS
- Regioni e motori DB supportati per l'autenticazione Kerberos in Amazon RDS
- Regioni e motori DB supportati per cluster DB Multi-AZ in Amazon RDS
- Regioni e motori DB supportati per Performance Insights in Amazon RDS
- Regioni e motori DB supportati per RDS Custom
- Regioni e motori DB supportati per Amazon RDS Proxy
- Regioni e motori DB supportati per l'integrazione di Secrets Manager con Amazon RDS
- Regioni e motori DB supportati per le integrazioni Zero-ETL di Amazon RDS con Amazon Redshift
- Funzionalità native del motore in Amazon RDS

Convenzioni tabella

Nelle tabelle delle sezioni relative alle funzionalità sono utilizzati i seguenti modelli per specificare i numeri di versione e il livello di disponibilità:

- Versione x.y: è disponibile solo la versione indicata.
- Versione x.y e successive: sono supportate la versione specificata e tutte le relative versioni secondarie successive. Ad esempio, "versione 10.11 e successive" significa che sono disponibili le versioni 10.11, 10.11.1 e 10.12.
- —: la funzionalità non è attualmente disponibile per il motore di database RDS selezionato o nella Regione AWS specificata.

Riferimento rapido alle funzionalità

Nella seguente tabella di riferimento rapido sono elencate tutte le funzionalità e il motore di database RDS disponibili. La disponibilità di regioni e versioni specifiche viene visualizzata nelle successive sezioni delle funzionalità.

Funzionalità	RDS per Db2	RDS per MariaDB	RDS for MySQL	RDS per Oracle	RDS per PostgreSQL.	RDS per SQL Server
Distribuzioni blu/verdi	–	Disponibilità	Disponibilità	–	Disponibilità	–
Backup automatici tra regioni	Disponibilità	Disponibilità	Disponibilità	Disponibilità	Disponibilità	Disponibilità
Replica di lettura tra	–	Disponibilità	Disponibilità	Disponibilità	Disponibilità	Disponibilità

Funzionalità	RDS per Db2	RDS per MariaDB	RDS for MySQL	RDS per Oracle	RDS per PostgreSQL.	RDS per SQL Server
region diverse						
Flussi di attività di databa	–	–	–	Disponibilità	–	Disponibilità
Modali dual-stack	–	Disponibilità	Disponibilità	Disponibilità	Disponibilità	Disponibilità
Esport one di snapsl in Amazo S3	–	Disponibilità	Disponibilità	–	Disponibilità	–
AWS Identity and Access Manag t autent zione del databa (IAM)	–	Disponibilità	Disponibilità	–	Disponibilità	–

Funzic ità	RDS per Db2	RDS per MariaDB	RDS for MySQL	RDS per Oracle	RDS per PostgreSQ L.	RDS per SQL Server
Autent zione Kerbe	Disponibilità	–	Disponibilità	Disponibilità	Disponibilità	Disponibilità
Cluste di databa Multi- AZ	–	–	Disponibilità	–	Disponibilità	–
Approf imenti sulle presta ni	–	Disponibilità	Disponibilità	Disponibilità	Disponibilità	Disponibilità
RDS Custor	–	–	–	Disponibilità	–	Disponibilità
Server proxy per RDS	–	Disponibilità	Disponibilità	–	Disponibilità	Disponibilità
Integra one di Secret Manag	Disponibilità	Disponibilità	Disponibilità	Disponibilità	Disponibilità	Disponibilità

Regioni e motori DB supportati per le distribuzioni Blue/Green di Amazon RDS

Un'implementazione blu/verde copia un ambiente di database di produzione in un ambiente di gestione temporanea separato e sincronizzato. Utilizzando le implementazioni blu/verde Amazon RDS, puoi apportare modifiche al database nell'ambiente di gestione temporanea senza influire sull'ambiente di produzione. Ad esempio, è possibile aggiornare la versione principale o secondaria del motore di database, modificare i parametri del database o apportare modifiche allo schema nell'ambiente di gestione temporanea. Quando sei pronto, puoi promuovere l'ambiente di gestione temporanea nel nuovo ambiente di database di produzione. Per ulteriori informazioni, consulta Utilizzo delle implementazioni blu/verde Amazon RDS per gli aggiornamenti del database.

La funzionalità delle implementazioni blu/verde è supportata per i seguenti motori:

- RDS per MariaDB versione 10.2 e successive
- RDS per MySQL versione 5.7 e successive
- RDS per MySQL versione 8.0.15 e successive
- RDS per PostgreSQL 11.21 e versioni successive
- RDS per PostgreSQL 12.16 e versioni successive
- RDS per PostgreSQL 13.12 e versioni successive
- RDS per PostgreSQL 14.9 e versioni successive
- RDS per PostgreSQL 15.4 e versioni successive
- RDS per PostgreSQL versione 16.1 e successive

La funzionalità delle implementazioni blu/verde non è supportata per i seguenti motori:

- RDS per Db2
- RDS per SQL Server
- RDS per Oracle

La funzionalità Blue/Green Deployments è supportata in tutti. Regioni AWS

Regioni e motori DB supportati per backup automatici tra regioni in Amazon RDS

Utilizzando la replica di backup in Amazon RDS, puoi configurare l'istanza database RDS per crere repliche di snapshot e registri delle transazioni in una regione di destinazione. Se per un'istanza database è configurata la replica di backup, RDS avvia una copia tra regioni di tutti gli snapshot e registri delle transazioni non appena sono pronti. Per ulteriori informazioni, consulta Replica dei backup automatici su un altro Regione AWS.

La replica di Backup è disponibile in tutte le versioni Regioni AWS tranne le seguenti:

- Africa (Città del Capo)
- Asia Pacifico (Hong Kong)
- Asia Pacific (Hyderabad)
- Asia Pacifico (Giacarta)
- Europa (Milano)
- Europa (Spagna)
- Europa (Zurigo)
- Medio Oriente (Bahrein)
- Medio Oriente (Emirati Arabi Uniti)

Per informazioni più dettagliate sulle limitazioni per la regione di backup di origine e destinazione, vedereReplica dei backup automatici su un altro Regione AWS.

Argomenti

- Replica di backup con RDS per Db2
- Replica di backup con RDS per MariaDB
- Replica di backup con RDS per MySQL
- Replica di backup con RDS per Oracle
- Replica di backup con RDS per PostgreSQL
- Replica di backup con RDS per SQL Server

Replica di backup con RDS per Db2

Amazon RDS supporta la replica dei backup per tutte le versioni attualmente disponibili di RDS for Db2.

Replica di backup con RDS per MariaDB

Amazon RDS supporta la replica di backup per tutte le versioni attualmente disponibili di RDS per MariaDB.

Replica di backup con RDS per MySQL

Amazon RDS supporta la replica di backup per tutte le versioni attualmente disponibili di RDS per MySQL.

Replica di backup con RDS per Oracle

Amazon RDS supporta la replica di backup per tutte le versioni attualmente disponibili di RDS per Oracle.

Replica di backup con RDS per PostgreSQL

Amazon RDS supporta la replica di backup per tutte le versioni attualmente disponibili di RDS per PostgreSQL.

Replica di backup con RDS per SQL Server

Amazon RDS supporta la replica di backup per tutte le versioni attualmente disponibili di RDS per SQL Server.

Regioni e motori DB supportati per repliche di lettura interregionali in Amazon RDS

Se in Amazon RDS vengono utilizzate le repliche di lettura tra regioni, puoi creare una replica di lettura MariaDB, MySQL, Oracle, PostgreSQL o SQL Server in una regione diversa rispetto all'istanza database di origine. Per informazioni sulle repliche di lettura tra regioni, con considerazioni sulle regioni di origine e destinazione, consulta Creazione di una replica di lettura in un altro Regione AWS.

Le repliche di lettura interregionali non sono disponibili per i seguenti motori:

- RDS per Db2

Argomenti

Repliche di lettura tra Regioni con RDS per MariaDB

Le repliche di lettura tra Regioni con RDS per MariaDB sono disponibili in tutte le Regioni per le versioni seguenti:

- RDS per MariaDB 10.11 (tutte le versioni disponibili)
- RDS per MariaDB 10.6 (tutte le versioni disponibili)
- RDS per MariaDB 10.5 (tutte le versioni disponibili)
- RDS per MariaDB 10.4 (tutte le versioni disponibili)
- RDS per MariaDB 10.3 (tutte le versioni disponibili)

Repliche di lettura tra Regioni con RDS per MySQL

Le repliche di lettura tra Regioni con RDS per MySQL sono disponibili in tutte le Regioni per le seguenti versioni:

- RDS per MySQL 8.0 (tutte le versioni disponibili)
- RDS per MySQL 5.7 (tutte le versioni disponibili)

Repliche di lettura tra Regioni con RDS per Oracle

Le repliche di lettura tra Regioni per RDS per Oracle sono disponibili in tutte le Regioni con le seguenti limitazioni di versione:

- Per RDS per Oracle 19c e 21c, le repliche di lettura tra regioni non sono disponibili nella configurazione multi-tenant dell'architettura CDB. Le repliche sono supportate nelle versioni non CDB e nella configurazione single-tenant dell'architettura CDB.

- Per RDS per Oracle 12c, sono disponibili repliche di lettura tra regioni per Oracle Enterprise Edition (EE) di Oracle Database 12c Release 1 (12.1) che utilizza 12.1.0.2.v10 e versioni successive alla 12c.

Per ulteriori informazioni sui requisiti aggiuntivi per le repliche di lettura tra regioni con RDS per Oracle, consulta Requisiti e considerazioni sulle repliche RDS per Oracle.

Repliche di lettura tra Regioni con RDS per PostgreSQL

Le repliche di lettura tra Regioni con RDS per PostgreSQL sono disponibili in tutte le Regioni per le seguenti versioni:

- RDS per PostgreSQL 16 (tutte le versioni disponibili)
- RDS per PostgreSQL 15 (tutte le versioni disponibili)
- RDS per PostgreSQL 14 (tutte le versioni disponibili)
- RDS per PostgreSQL 13 (tutte le versioni disponibili)
- RDS per PostgreSQL 12 (tutte le versioni disponibili)
- RDS per PostgreSQL 11 (tutte le versioni disponibili)
- RDS per PostgreSQL 10 (tutte le versioni disponibili)

Repliche di lettura tra regioni con RDS per SQL Server

Le repliche di lettura tra regioni con RDS per SQL Server sono disponibili in tutte le regioni eccetto le seguenti:

- Africa (Città del Capo)
- Asia Pacifico (Hong Kong)
- Asia Pacific (Hyderabad)
- Asia Pacifico (Giacarta)
- Asia Pacifico (Melbourne)
- Canada occidentale (Calgary)
- Europa (Milano)
- Europa (Spagna)
- Europa (Zurigo)

- Israele (Tel Aviv)
- Medio Oriente (Bahrein)
- Medio Oriente (Emirati Arabi Uniti)

Le repliche di lettura tra regioni con RDS per SQL Server sono disponibili per le seguenti versioni utilizzando Microsoft SQL Server Enterprise Edition:

- RDS per SQL Server 2022
- RDS per SQL Server 2.019 (versione 15.00.4073.23 e successive)
- RDS per SQL Server 2017 (versione 14.00.3281.6 e successive)
- RDS per SQL Server 2016 (versione 13.00.6300.2 e successive)

Regioni e motori DB supportati per i flussi di attività del database in Amazon RDS

Utilizzando i flussi di attività del database in Amazon RDS, puoi monitorare e impostare allarmi per l'attività di controllo nel tuo database Oracle e nel database SQL Server. Per ulteriori informazioni, consulta Panoramica dei flussi di attività di database.

I flussi di attività di database non sono disponibili con i seguenti motori:

- RDS per Db2
- RDS per MariaDB
- RDS for MySQL
- RDS per PostgreSQL

Argomenti

- Flussi di attività del database con RDS per Oracle
- Flussi di attività del database con RDS per SQL Server

Flussi di attività del database con RDS per Oracle

Di seguito sono riportate le regioni e le versioni di motore disponibili per i flussi di attività del database con RDS per Oracle.

Per ulteriori informazioni sui requisiti aggiuntivi per i flussi di attività di database con RDS per Oracle, consulta Panoramica dei flussi di attività di database.

Regione	RDS per Oracle 21c	RDS per Oracle 19c
Stati Uniti orientali (Ohio)	–	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 e versioni successive, con Enterprise Edition (EE) o Standard Edition 2 (SE2)
Stati Uniti orientali (Virginia settentrionale)	–	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 e versioni successive, con Enterprise Edition (EE) o Standard Edition 2 (SE2)
Stati Uniti occidentali (California settentrionale)	–	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 e versioni successive, con Enterprise Edition (EE) o Standard Edition 2 (SE2)
US West (Oregon)	–	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 e versioni successive, con Enterprise Edition (EE) o Standard Edition 2 (SE2)
Africa (Città del Capo)	–	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 e versioni successive, con Enterprise Edition (EE) o Standard Edition 2 (SE2)
Asia Pacifico (Hong Kong)	–	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 e versioni successive, con Enterprise Edition (EE) o Standard Edition 2 (SE2)
Asia Pacific (Hyderabad)	–	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 e versioni successive,

Regione	RDS per Oracle 21c	RDS per Oracle 19c
		con Enterprise Edition (EE) o Standard Edition 2 (SE2)
Asia Pacifico (Giacarta)	–	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 e versioni successive, con Enterprise Edition (EE) o Standard Edition 2 (SE2)
Asia Pacifico (Melbourne)	–	–
Asia Pacifico (Mumbai)	–	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 e versioni successive, con Enterprise Edition (EE) o Standard Edition 2 (SE2)
Asia Pacifico (Osaka-Locale)	–	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 e versioni successive, con Enterprise Edition (EE) o Standard Edition 2 (SE2)
Asia Pacifico (Seoul)	–	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 e versioni successive, con Enterprise Edition (EE) o Standard Edition 2 (SE2)
Asia Pacifico (Singapore)	–	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 e versioni successive, con Enterprise Edition (EE) o Standard Edition 2 (SE2)
Asia Pacifico (Sydney)	–	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 e versioni successive, con Enterprise Edition (EE) o Standard Edition 2 (SE2)

Regione	RDS per Oracle 21c	RDS per Oracle 19c
Asia Pacifico (Tokyo)	–	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 e versioni successive, con Enterprise Edition (EE) o Standard Edition 2 (SE2)
Canada (Centrale)	–	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 e versioni successive, con Enterprise Edition (EE) o Standard Edition 2 (SE2)
Canada occidentale (Calgary)	–	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 e versioni successive, con Enterprise Edition (EE) o Standard Edition 2 (SE2)
Cina (Pechino)	–	–
China (Ningxia)	–	–
Europa (Francoforte)	–	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 e versioni successive, con Enterprise Edition (EE) o Standard Edition 2 (SE2)
Europa (Irlanda)	–	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 e versioni successive, con Enterprise Edition (EE) o Standard Edition 2 (SE2)
Europa (Londra)	–	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 e versioni successive, con Enterprise Edition (EE) o Standard Edition 2 (SE2)

Regione	RDS per Oracle 21c	RDS per Oracle 19c
Europa (Milano)	–	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 e versioni successive, con Enterprise Edition (EE) o Standard Edition 2 (SE2)
Europa (Parigi)	–	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 e versioni successive, con Enterprise Edition (EE) o Standard Edition 2 (SE2)
Europa (Spagna)	–	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 e versioni successive, con Enterprise Edition (EE) o Standard Edition 2 (SE2)
Europa (Stoccolma)	–	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 e versioni successive, con Enterprise Edition (EE) o Standard Edition 2 (SE2)
Europa (Zurigo)	–	–
Asia Pacifico (Melbourne)	–	–
Medio Oriente (Bahrein)	–	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 e versioni successive, con Enterprise Edition (EE) o Standard Edition 2 (SE2)
Medio Oriente (Emirati Arabi Uniti)	–	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 e versioni successive, con Enterprise Edition (EE) o Standard Edition 2 (SE2)

Regione	RDS per Oracle 21c	RDS per Oracle 19c
Sud America (San Paolo)	–	Oracle Database 19.0.0.0.ru-2019-07.rur-2019-07.r1 e versioni successive, con Enterprise Edition (EE) o Standard Edition 2 (SE2)
AWS GovCloud (Stati Uniti orientali)	–	–
AWS GovCloud (Stati Uniti occidentali)	–	–

Flussi di attività del database con RDS per SQL Server

Di seguito sono riportate le regioni e le versioni di motore disponibili per i flussi di attività del database con RDS per SQL Server.

Per ulteriori informazioni sui requisiti aggiuntivi per i flussi di attività del database con RDS per SQL Server, consulta Panoramica dei flussi di attività di database.

Regione	RDS per SQL Server 2019	RDS per SQL Server 2017	RDS per SQL Server 2016	RDS per SQL Server 2014
Stati Uniti orientali (Ohio)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	–
Stati Uniti orientali (Virginia settentrionale)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	–
Stati Uniti occidentali (California settentrionale)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	–
US West (Oregon)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	–

Regione	RDS per SQL Server 2019	RDS per SQL Server 2017	RDS per SQL Server 2016	RDS per SQL Server 2014
Africa (Città del Capo)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	–
Asia Pacifico (Hong Kong)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	–
Asia Pacific (Hyderabad)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	–
Asia Pacifico (Giacarta)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	–
Asia Pacifico (Melbourne)	–	–	–	–
Asia Pacifico (Mumbai)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	–
Asia Pacifico (Osaka-Locale)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	–
Asia Pacifico (Seoul)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	–
Asia Pacifico (Singapore)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	–
Asia Pacifico (Sydney)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	–
Asia Pacifico (Tokyo)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	–
Canada (Centrale)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	–

Regione	RDS per SQL Server 2019	RDS per SQL Server 2017	RDS per SQL Server 2016	RDS per SQL Server 2014
Canada occidentale (Calgary)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	–
Cina (Pechino)	–	–	–	–
China (Ningxia)	–	–	–	–
Europa (Francoforte)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	–
Europa (Irlanda)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	–
Europa (Londra)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	–
Europa (Milano)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	–
Europa (Parigi)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	–
Europa (Spagna)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	–
Europa (Stoccolma)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	–
Europa (Zurigo)	–	–	–	–
Israele (Tel Aviv)	–	–	–	–
Medio Oriente (Bahrein)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	–

Regione	RDS per SQL Server 2019	RDS per SQL Server 2017	RDS per SQL Server 2016	RDS per SQL Server 2014
Medio Oriente (Emirati Arabi Uniti)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	–
Sud America (San Paolo)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	–
AWS GovCloud (Stati Uniti orientali)	–	–	–	–
AWS GovCloud (Stati Uniti occidentali)	–	–	–	–

Regioni e motori DB supportati per la modalità dual-stack in Amazon RDS

Utilizzando la modalità dual-stack in RDS, le risorse possono comunicare con l'istanza database tramite IP versione 4 (IPv4), IPv6 o entrambi i protocolli. Per ulteriori informazioni, consulta Modalità dual-stack.

Argomenti

- Modalità dual-stack con RDS per Db2
- Modalità dual-stack con RDS per MariaDB
- Modalità dual-stack con RDS per MySQL
- Modalità dual-stack con RDS per Oracle
- Modalità dual-stack con RDS per PostgreSQL
- Modalità dual-stack con RDS per SQL Server

Modalità dual-stack con RDS per Db2

Le seguenti regioni e versioni del motore sono disponibili per la modalità dual-stack con RDS per Db2.

Regione	RDS per Db2 11.5				
Stati Uniti orientali (Ohio)	Tutte le versioni disponibili				
Stati Uniti orientali (Virginia settentrionale)	Tutte le versioni disponibili				
Stati Uniti occidentali (California settentrionale)	Tutte le versioni disponibili				
US West (Oregon)	Tutte le versioni disponibili				
Africa (Città del Capo)	Tutte le versioni disponibili				
Asia Pacifico (Hong Kong)	Tutte le versioni disponibili				
Asia Pacific (Hyderabad)	Tutte le versioni disponibili				
Asia Pacifico (Giacarta)	Tutte le versioni disponibili				

Regione	RDS per Db2 11.5				
Asia Pacifico (Melbourne)	Tutte le versioni disponibili				
Asia Pacifico (Mumbai)	Tutte le versioni disponibili				
Asia Pacifico (Osaka-Locale)	Tutte le versioni disponibili				
Asia Pacifico (Seoul)	Tutte le versioni disponibili				
Asia Pacifico (Singapore)	Tutte le versioni disponibili				
Asia Pacifico (Sydney)	Tutte le versioni disponibili				
Asia Pacifico (Tokyo)	Tutte le versioni disponibili				
Canada (Centrale)	Tutte le versioni disponibili				
Canada occidentale (Calgary)	–				

Regione	RDS per Db2 11.5				
Cina (Pechino)	–				
China (Ningxia)	–				
Europa (Francoforte)	Tutte le versioni disponibili				
Europa (Irlanda)	Tutte le versioni disponibili				
Europa (Londra)	Tutte le versioni disponibili				
Europa (Milano)	Tutte le versioni disponibili				
Europa (Parigi)	Tutte le versioni disponibili				
Europa (Spagna)	Tutte le versioni disponibili				
Europa (Stoccolma)	Tutte le versioni disponibili				

Regione	RDS per Db2 11.5				
Europa (Zurigo)	Tutte le versioni disponibili				
Israele (Tel Aviv)	–				
Medio Oriente (Bahrein)	Tutte le versioni disponibili				
Medio Oriente (Emirati Arabi Uniti)	Tutte le versioni disponibili				
Sud America (San Paolo)	Tutte le versioni disponibili				
AWS GovCloud (Stati Uniti orientali)	–				
AWS GovCloud (Stati Uniti occidentali)	–				

Modalità dual-stack con RDS per MariaDB

Di seguito sono riportate le regioni e le versioni di motore disponibili per la modalità dual-stack con RDS per MariaDB.

Regione	RDS per MariaDB 10.11	RDS per MariaDB 10.6	RDS per MariaDB 10.5	RDS per MariaDB 10.4	RDS per MariaDB 10.3
Stati Uniti orientali (Ohio)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Stati Uniti orientali (Virginia settentrionale)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Stati Uniti occidentali (California settentrionale)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
US West (Oregon)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Africa (Città del Capo)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Asia Pacifico (Hong Kong)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Asia Pacific (Hyderabad)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili

Regione	RDS per MariaDB 10.11	RDS per MariaDB 10.6	RDS per MariaDB 10.5	RDS per MariaDB 10.4	RDS per MariaDB 10.3
Asia Pacifico (Giacarta)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Asia Pacifico (Melbourne)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Asia Pacifico (Mumbai)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Asia Pacifico (Osaka-Locale)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Asia Pacifico (Seoul)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Asia Pacifico (Singapore)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Asia Pacifico (Sydney)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Asia Pacifico (Tokyo)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Canada (Centrale)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili

Regione	RDS per MariaDB 10.11	RDS per MariaDB 10.6	RDS per MariaDB 10.5	RDS per MariaDB 10.4	RDS per MariaDB 10.3
Canada occidentale (Calgary)	–	–	–	–	–
Cina (Pechino)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Cina (Ningxia)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Europa (Francoforte)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Europa (Irlanda)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Europa (Londra)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Europa (Milano)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Europa (Parigi)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Europa (Spagna)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili

Regione	RDS per MariaDB 10.11	RDS per MariaDB 10.6	RDS per MariaDB 10.5	RDS per MariaDB 10.4	RDS per MariaDB 10.3
Europa (Stoccolma)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Europa (Zurigo)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Israele (Tel Aviv)	–	–	–	–	–
Medio Oriente (Bahrein)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Medio Oriente (Emirati Arabi Uniti)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Sud America (San Paolo)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
AWS GovCloud (Stati Uniti orientali)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
AWS GovCloud (Stati Uniti occidentali)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili

Modalità dual-stack con RDS per MySQL

Di seguito sono riportate le regioni e le versioni di motore disponibili per la modalità dual-stack con RDS per MySQL.

Regione	RDS per MySQL 8.0	RDS per MySQL 5.7	RDS per MySQL 5.6
Stati Uniti orientali (Ohio)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Stati Uniti orientali (Virginia settentrionale)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Stati Uniti occidentali (California settentrionale)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
US West (Oregon)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Africa (Città del Capo)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Asia Pacifico (Hong Kong)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Asia Pacific (Hyderabad)	Tutte le versioni disponibili	Tutte le versioni disponibili	–
Asia Pacifico (Giacarta)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Asia Pacifico (Melbourne)	Tutte le versioni disponibili	Tutte le versioni disponibili	–
Asia Pacifico (Mumbai)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili

Regione	RDS per MySQL 8.0	RDS per MySQL 5.7	RDS per MySQL 5.6
Asia Pacifico (Osaka-Locale)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Asia Pacifico (Seoul)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Asia Pacifico (Singapore)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Asia Pacifico (Sydney)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Asia Pacifico (Tokyo)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Canada (Centrale)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Canada occidentale (Calgary)	–	–	–
Cina (Pechino)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Cina (Ningxia)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Europa (Francoforte)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Europa (Irlanda)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Europa (Londra)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Europa (Milano)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili

Regione	RDS per MySQL 8.0	RDS per MySQL 5.7	RDS per MySQL 5.6
Europa (Parigi)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Europa (Spagna)	Tutte le versioni disponibili	Tutte le versioni disponibili	–
Europa (Stoccolma)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Europa (Zurigo)	Tutte le versioni disponibili	Tutte le versioni disponibili	–
Israele (Tel Aviv)	–	–	–
Medio Oriente (Bahrein)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Medio Oriente (Emirati Arabi Uniti)	Tutte le versioni disponibili	Tutte le versioni disponibili	–
Sud America (San Paolo)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
AWS GovCloud (Stati Uniti orientali)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
AWS GovCloud (Stati Uniti occidentali)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili

Modalità dual-stack con RDS per Oracle

Di seguito sono riportate le regioni e le versioni di motore disponibili per la modalità dual-stack con RDS per Oracle.

Regione	RDS per Oracle 21c	RDS per Oracle 19c	RDS per Oracle 12c
Stati Uniti orientali (Ohio)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Stati Uniti orientali (Virginia settentrionale)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Stati Uniti occidentali (California settentrionale)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
US West (Oregon)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Africa (Città del Capo)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Asia Pacifico (Hong Kong)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Asia Pacific (Hyderabad)	–	–	–
Asia Pacifico (Giacarta)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Asia Pacifico (Melbourne)	–	–	–
Asia Pacifico (Mumbai)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Asia Pacifico (Osaka-Locale)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Asia Pacifico (Seoul)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili

Regione	RDS per Oracle 21c	RDS per Oracle 19c	RDS per Oracle 12c
Asia Pacifico (Singapore)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Asia Pacifico (Sydney)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Asia Pacifico (Tokyo)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Canada (Centrale)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Canada occidentale (Calgary)	–	–	–
Cina (Pechino)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Cina (Ningxia)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Europa (Francoforte)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Europa (Irlanda)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Europa (Londra)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Europa (Milano)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Europa (Parigi)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Europa (Spagna)	–	–	–

Regione	RDS per Oracle 21c	RDS per Oracle 19c	RDS per Oracle 12c
Europa (Stoccolma)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Europa (Zurigo)	–	–	–
Israele (Tel Aviv)	–	–	–
Medio Oriente (Bahrein)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Medio Oriente (Emirati Arabi Uniti)	–	–	–
Sud America (San Paolo)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
AWS GovCloud (Stati Uniti orientali)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
AWS GovCloud (Stati Uniti occidentali)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili

Modalità dual-stack con RDS per PostgreSQL

Di seguito sono riportate le regioni e le versioni di motore disponibili per la modalità dual-stack con RDS per PostgreSQL.

Regione	RDS per PostgreSQL 16	RDS per PostgreSQL 15	RDS per PostgreSQL 14	RDS per PostgreSQL 13	RDS per PostgreSQL 12	RDS per PostgreSQL 11	RDS per PostgreSQL 10
Stati Uniti orientali (Ohio)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili

Regione	RDS per PostgreSQL 16	RDS per PostgreSQL 15	RDS per PostgreSQL 14	RDS per PostgreSQL 13	RDS per PostgreSQL 12	RDS per PostgreSQL 11	RDS per PostgreSQL 10
Stati Uniti orientali (Virginia settentrionale)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Stati Uniti occidentali (California settentrionale)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
US West (Oregon)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Africa (Città del Capo)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Asia Pacifico (Hong Kong)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Asia Pacific (Hyderabad)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili

Regione	RDS per PostgreSQL 16	RDS per PostgreSQL 15	RDS per PostgreSQL 14	RDS per PostgreSQL 13	RDS per PostgreSQL 12	RDS per PostgreSQL 11	RDS per PostgreSQL 10
Asia Pacifico (Melbourne)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Asia Pacifico (Giacarta)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Asia Pacifico (Mumbai)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Asia Pacifico (Osaka-Locale)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Asia Pacifico (Seoul)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Asia Pacifico (Singapore)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Asia Pacifico (Sydney)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili

Regione	RDS per PostgreSQL 16	RDS per PostgreSQL 15	RDS per PostgreSQL 14	RDS per PostgreSQL 13	RDS per PostgreSQL 12	RDS per PostgreSQL 11	RDS per PostgreSQL 10
Asia Pacifico (Tokyo)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Canada (Centrale)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Canada occidentale (Calgary)	–	–	–	–	–	–	–
Cina (Pechino)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Cina (Ningxia)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Europa (Francoforte)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Europa (Irlanda)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili

Regione	RDS per PostgreSQL 16	RDS per PostgreSQL 15	RDS per PostgreSQL 14	RDS per PostgreSQL 13	RDS per PostgreSQL 12	RDS per PostgreSQL 11	RDS per PostgreSQL 10
Europa (Londra)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Europa (Milano)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Europa (Parigi)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Europa (Spagna)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Europa (Stoccolma)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Europa (Zurigo)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Israele (Tel Aviv)	–	–	–	–	–	–	–

Regione	RDS per PostgreSQL 16	RDS per PostgreSQL 15	RDS per PostgreSQL 14	RDS per PostgreSQL 13	RDS per PostgreSQL 12	RDS per PostgreSQL 11	RDS per PostgreSQL 10
Medio Oriente (Bahrein)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Medio Oriente (Emirati Arabi Uniti)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Sud America (San Paolo)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
AWS GovCloud (Stati Uniti orientali)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
AWS GovCloud (Stati Uniti occidentali)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili

Modalità dual-stack con RDS per SQL Server

Di seguito sono riportate le regioni e le versioni di motore disponibili per la modalità dual-stack con RDS per SQL Server.

Regione	RDS per SQL Server 2019	RDS per SQL Server 2017	RDS per SQL Server 2016	RDS per SQL Server 2014
Stati Uniti orientali (Ohio)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	–
Stati Uniti orientali (Virginia settentrionale)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	–
Stati Uniti occidentali (California settentrionale)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	–
US West (Oregon)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	–
Africa (Città del Capo)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	–
Asia Pacifico (Hong Kong)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	–
Asia Pacific (Hyderabad)	–	–	–	–
Asia Pacifico (Giacarta)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	–
Asia Pacifico (Melbourne)	–	–	–	–
Asia Pacifico (Mumbai)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	–
Asia Pacifico (Osaka-Locale)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	–

Regione	RDS per SQL Server 2019	RDS per SQL Server 2017	RDS per SQL Server 2016	RDS per SQL Server 2014
Asia Pacifico (Seoul)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	–
Asia Pacifico (Singapore)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	–
Asia Pacifico (Sydney)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	–
Asia Pacifico (Tokyo)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	–
Canada (Centrale)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	–
Canada occidentale (Calgary)	–	–	–	–
Cina (Pechino)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	–
Cina (Ningxia)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	–
Europa (Francoforte)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	–
Europa (Irlanda)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	–
Europa (Londra)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	–
Europa (Milano)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	–

Regione	RDS per SQL Server 2019	RDS per SQL Server 2017	RDS per SQL Server 2016	RDS per SQL Server 2014
Europa (Parigi)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	–
Europa (Spagna)	–	–	–	–
Europa (Stoccolma)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	–
Europa (Zurigo)	–	–	–	–
Israele (Tel Aviv)	–	–	–	–
Medio Oriente (Bahrein)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	–
Medio Oriente (Emirati Arabi Uniti)	–	–	–	–
Sud America (San Paolo)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	–
AWS GovCloud (Stati Uniti orientali)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	–
AWS GovCloud (Stati Uniti occidentali)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	–

Regioni e motori DB supportati per l'esportazione di snapshot in S3 in Amazon RDS

È possibile esportare i dati dello snapshot DB RDS in un bucket Amazon S3. È possibile esportare tutti i tipi di snapshot DB, inclusi snapshot manuali, snapshot automatici di sistema e snapshot

creati dal servizio AWS Backup. Dopo l'esportazione dei dati, è possibile analizzare i dati esportati direttamente mediante strumenti quali Amazon Athena o Amazon Redshift Spectrum. Per ulteriori informazioni, consulta Esportazione dei dati dello snapshot DB in Simple Storage Service (Amazon S3).

L'esportazione di istantanee in S3 non è disponibile per i seguenti motori:

- RDS per Db2
- RDS per Oracle
- RDS per SQL Server

Argomenti

- Esportazione di snapshot in S3 con RDS per MariaDB
- Esportazione di snapshot in S3 con RDS per MySQL
- Esportazione di snapshot in S3 con RDS per PostgreSQL

Esportazione di snapshot in S3 con RDS per MariaDB

Di seguito sono riportate le regioni e le versioni di motore disponibili per l'esportazione di snapshot in S3 con RDS per MariaDB.

Regione	RDS per MariaDB 10.11	RDS per MariaDB 10.6	RDS per MariaDB 10.5	RDS per MariaDB 10.4	RDS per MariaDB 10.3
Stati Uniti orientali (Ohio)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Stati Uniti orientali (Virginia settentrionale)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili

Regione	RDS per MariaDB 10.11	RDS per MariaDB 10.6	RDS per MariaDB 10.5	RDS per MariaDB 10.4	RDS per MariaDB 10.3
Stati Uniti occidentali (California settentrionale)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
US West (Oregon)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Africa (Città del Capo)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Asia Pacifico (Hong Kong)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Asia Pacific (Hyderabad)	–	–	–	–	–
Asia Pacifico (Giacarta)	–	–	–	–	–
Asia Pacifico (Melbourne)	–	–	–	–	–
Asia Pacifico (Mumbai)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Asia Pacifico (Osaka-Locale)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili

Regione	RDS per MariaDB 10.11	RDS per MariaDB 10.6	RDS per MariaDB 10.5	RDS per MariaDB 10.4	RDS per MariaDB 10.3
Asia Pacifico (Seoul)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Asia Pacifico (Singapore)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Asia Pacifico (Sydney)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Asia Pacifico (Tokyo)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Canada (Centrale)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Canada occidentale (Calgary)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Cina (Pechino)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Cina (Ningxia)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Europa (Francoforte)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili

Regione	RDS per MariaDB 10.11	RDS per MariaDB 10.6	RDS per MariaDB 10.5	RDS per MariaDB 10.4	RDS per MariaDB 10.3
Europa (Irlanda)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Europa (Londra)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Europa (Milano)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Europa (Parigi)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Europa (Spagna)	–	–	–	–	–
Europa (Stoccolma)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Europa (Zurigo)	–	–	–	–	–
Israele (Tel Aviv)	–	–	–	–	–
Medio Oriente (Bahrein)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili

Regione	RDS per MariaDB 10.11	RDS per MariaDB 10.6	RDS per MariaDB 10.5	RDS per MariaDB 10.4	RDS per MariaDB 10.3
Medio Oriente (Emirati Arabi Uniti)	–	–	–	–	–
Sud America (San Paolo)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
AWS GovCloud (Stati Uniti orientali)	–	–	–	–	–
AWS GovCloud (Stati Uniti occidentali)	–	–	–	–	–

Esportazione di snapshot in S3 con RDS per MySQL

Di seguito sono riportate le versioni di motore e le regioni disponibili per l'esportazione di snapshot in S3 con RDS per MySQL.

Regione	RDS per MySQL 8.0	RDS per MySQL 5.7
Stati Uniti orientali (Ohio)	Tutte le versioni disponibili	Tutte le versioni disponibili
Stati Uniti orientali (Virginia settentrionale)	Tutte le versioni disponibili	Tutte le versioni disponibili
Stati Uniti occidentali (California settentrionale)	Tutte le versioni disponibili	Tutte le versioni disponibili

Regione	RDS per MySQL 8.0	RDS per MySQL 5.7
US West (Oregon)	Tutte le versioni disponibili	Tutte le versioni disponibili
Africa (Città del Capo)	Tutte le versioni disponibili	Tutte le versioni disponibili
Asia Pacifico (Hong Kong)	Tutte le versioni disponibili	Tutte le versioni disponibili
Asia Pacific (Hyderabad)	–	–
Asia Pacifico (Giacarta)	–	–
Asia Pacifico (Melbourne)	–	–
Asia Pacifico (Mumbai)	Tutte le versioni disponibili	Tutte le versioni disponibili
Asia Pacifico (Osaka-Locale)	Tutte le versioni disponibili	Tutte le versioni disponibili
Asia Pacifico (Seoul)	Tutte le versioni disponibili	Tutte le versioni disponibili
Asia Pacifico (Singapore)	Tutte le versioni disponibili	Tutte le versioni disponibili
Asia Pacifico (Sydney)	Tutte le versioni disponibili	Tutte le versioni disponibili
Asia Pacifico (Tokyo)	Tutte le versioni disponibili	Tutte le versioni disponibili
Canada (Centrale)	Tutte le versioni disponibili	Tutte le versioni disponibili
Canada occidentale (Calgary)	–	–
Cina (Pechino)	Tutte le versioni disponibili	Tutte le versioni disponibili
Cina (Ningxia)	Tutte le versioni disponibili	Tutte le versioni disponibili
Europa (Francoforte)	Tutte le versioni disponibili	Tutte le versioni disponibili
Europa (Irlanda)	Tutte le versioni disponibili	Tutte le versioni disponibili
Europa (Londra)	Tutte le versioni disponibili	Tutte le versioni disponibili
Europa (Milano)	Tutte le versioni disponibili	Tutte le versioni disponibili

Regione	RDS per MySQL 8.0	RDS per MySQL 5.7
Europa (Parigi)	Tutte le versioni disponibili	Tutte le versioni disponibili
Europa (Spagna)	–	–
Europa (Stoccolma)	Tutte le versioni disponibili	Tutte le versioni disponibili
Europa (Zurigo)	–	–
Israele (Tel Aviv)	–	–
Medio Oriente (Bahrein)	Tutte le versioni disponibili	Tutte le versioni disponibili
Medio Oriente (Emirati Arabi Uniti)	–	–
Sud America (San Paolo)	Tutte le versioni disponibili	Tutte le versioni disponibili
AWS GovCloud (Stati Uniti orientali)	–	–
AWS GovCloud (Stati Uniti occidentali)	–	–

Esportazione di snapshot in S3 con RDS per PostgreSQL

Di seguito sono riportate le versioni di motore e le regioni disponibili per l'esportazione di snapshot in S3 con RDS per PostgreSQL.

Regione	RDS per PostgreSQL 16	RDS per PostgreSQL 15	RDS per PostgreSQL 14	RDS per PostgreSQL 13	RDS per PostgreSQL 12	RDS per PostgreSQL 11	RDS per PostgreSQL 10
Stati Uniti orientali (Ohio)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili

Regione	RDS per PostgreSQL 16	RDS per PostgreSQL 15	RDS per PostgreSQL 14	RDS per PostgreSQL 13	RDS per PostgreSQL 12	RDS per PostgreSQL 11	RDS per PostgreSQL 10
Stati Uniti orientali (Virginia settentrionale)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Stati Uniti occidentali (California settentrionale)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
US West (Oregon)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Africa (Città del Capo)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Asia Pacifico (Hong Kong)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Asia Pacific (Hyderabad)	–	–	–	–	–	–	–

Regione	RDS per PostgreSQL 16	RDS per PostgreSQL 15	RDS per PostgreSQL 14	RDS per PostgreSQL 13	RDS per PostgreSQL 12	RDS per PostgreSQL 11	RDS per PostgreSQL 10
Asia Pacifico (Giacarta)	–	–	–	–	–	–	–
Asia Pacifico (Melbourne)	–	–	–	–	–	–	–
Asia Pacifico (Mumbai)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Asia Pacifico (Osaka-Locale)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Asia Pacifico (Seoul)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Asia Pacifico (Singapore)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Asia Pacifico (Sydney)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili

Regione	RDS per PostgreSQL 16	RDS per PostgreSQL 15	RDS per PostgreSQL 14	RDS per PostgreSQL 13	RDS per PostgreSQL 12	RDS per PostgreSQL 11	RDS per PostgreSQL 10
Asia Pacifico (Tokyo)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Canada (Centrale)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Canada occidentale (Calgary)	–	–	–	–	–	–	–
Cina (Pechino)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Cina (Ningxia)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Europa (Francoforte)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Europa (Irlanda)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili

Regione	RDS per PostgreSQL 16	RDS per PostgreSQL 15	RDS per PostgreSQL 14	RDS per PostgreSQL 13	RDS per PostgreSQL 12	RDS per PostgreSQL 11	RDS per PostgreSQL 10
Europa (Londra)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Europa (Milano)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Europa (Parigi)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Europa (Spagna)	–	–	–	–	–	–	–
Europa (Stoccolma)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Europa (Zurigo)	–	–	–	–	–	–	–
Israele (Tel Aviv)	–	–	–	–	–	–	–
Medio Oriente (Bahrein)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili

Regione	RDS per PostgreSQL 16	RDS per PostgreSQL 15	RDS per PostgreSQL 14	RDS per PostgreSQL 13	RDS per PostgreSQL 12	RDS per PostgreSQL 11	RDS per PostgreSQL 10
Medio Oriente (Emirati Arabi Uniti)	–	–	–	–	–	–	–
Sud America (San Paolo)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
AWS GovCloud (Stati Uniti orientali)	–	–	–	–	–	–	–
AWS GovCloud (Stati Uniti occidentali)	–	–	–	–	–	–	–

Regioni e motori DB supportati per l'autenticazione del database IAM in Amazon RDS

Utilizzando l'autenticazione database IAM in Amazon RDS, puoi autenticarti senza password quando ti connetti a un'istanza database. Utilizzi invece un token di autenticazione. Per ulteriori informazioni, consulta Autenticazione del database IAM per MariaDB, MySQL e PostgreSQL.

L'autenticazione database IAM è disponibile per i motori di database seguenti:

- RDS per Db2
- RDS per Oracle
- RDS per SQL Server

Argomenti

Autenticazione database IAM con RDS per MariaDB

Di seguito sono riportate le versioni di motore e le regioni disponibili per l'autenticazione del database IAM con RDS per MariaDB.

Regione	RDS per MariaDB 10.11	RDS per MariaDB 10.6	RDS per MariaDB 10.5	RDS per MariaDB 10.4	RDS per MariaDB 10.3
Stati Uniti orientali (Ohio)	Tutte le versioni disponibili	Tutte le versioni disponibili	–	–	–
Stati Uniti orientali (Virginia settentrionale)	Tutte le versioni disponibili	Tutte le versioni disponibili	–	–	–
Stati Uniti occidentali (California settentrionale)	Tutte le versioni disponibili	Tutte le versioni disponibili	–	–	–

Regione	RDS per MariaDB 10.11	RDS per MariaDB 10.6	RDS per MariaDB 10.5	RDS per MariaDB 10.4	RDS per MariaDB 10.3
US West (Oregon)	Tutte le versioni disponibili	Tutte le versioni disponibili	–	–	–
Africa (Città del Capo)	Tutte le versioni disponibili	Tutte le versioni disponibili	–	–	–
Asia Pacifico (Hong Kong)	Tutte le versioni disponibili	Tutte le versioni disponibili	–	–	–
Asia Pacific (Hyderabad)	–	–	–	–	–
Asia Pacifico (Giacarta)	Tutte le versioni disponibili	Tutte le versioni disponibili	–	–	–
Asia Pacifico (Melbourne)	–	–	–	–	–
Asia Pacifico (Mumbai)	Tutte le versioni disponibili	Tutte le versioni disponibili	–	–	–
Asia Pacifico (Osaka-Lo cale)	Tutte le versioni disponibili	Tutte le versioni disponibili	–	–	–
Asia Pacifico (Seoul)	Tutte le versioni disponibili	Tutte le versioni disponibili	–	–	–

Regione	RDS per MariaDB 10.11	RDS per MariaDB 10.6	RDS per MariaDB 10.5	RDS per MariaDB 10.4	RDS per MariaDB 10.3
Asia Pacifico (Singapore)	Tutte le versioni disponibili	Tutte le versioni disponibili	–	–	–
Asia Pacifico (Sydney)	Tutte le versioni disponibili	Tutte le versioni disponibili	–	–	–
Asia Pacifico (Tokyo)	Tutte le versioni disponibili	Tutte le versioni disponibili	–	–	–
Canada (Centrale)	Tutte le versioni disponibili	Tutte le versioni disponibili	–	–	–
Canada occidentale (Calgary)	Tutte le versioni disponibili	Tutte le versioni disponibili	–	–	–
Cina (Pechino)	Tutte le versioni disponibili	Tutte le versioni disponibili	–	–	–
Cina (Ningxia)	Tutte le versioni disponibili	Tutte le versioni disponibili	–	–	–
Europa (Francoforte)	Tutte le versioni disponibili	Tutte le versioni disponibili	–	–	–
Europa (Irlanda)	Tutte le versioni disponibili	Tutte le versioni disponibili	–	–	–

Regione	RDS per MariaDB 10.11	RDS per MariaDB 10.6	RDS per MariaDB 10.5	RDS per MariaDB 10.4	RDS per MariaDB 10.3
Europa (Londra)	Tutte le versioni disponibili	Tutte le versioni disponibili	–	–	–
Europa (Milano)	Tutte le versioni disponibili	Tutte le versioni disponibili	–	–	–
Europa (Parigi)	Tutte le versioni disponibili	Tutte le versioni disponibili	–	–	–
Europa (Spagna)	–	–	–	–	–
Europa (Stoccolma)	Tutte le versioni disponibili	Tutte le versioni disponibili	–	–	–
Europa (Zurigo)	–	–	–	–	–
Israele (Tel Aviv)	–	–	–	–	–
Medio Oriente (Bahrein)	Tutte le versioni disponibili	Tutte le versioni disponibili	–	–	–
Medio Oriente (Emirati Arabi Uniti)	–	–	–	–	–

Regione	RDS per MariaDB 10.11	RDS per MariaDB 10.6	RDS per MariaDB 10.5	RDS per MariaDB 10.4	RDS per MariaDB 10.3
Sud America (San Paolo)	Tutte le versioni disponibili	Tutte le versioni disponibili	–	–	–
AWS GovCloud (Stati Uniti orientali)	Tutte le versioni disponibili	Tutte le versioni disponibili	–	–	–
AWS GovCloud (Stati Uniti occidentali)	Tutte le versioni disponibili	Tutte le versioni disponibili	–	–	–

Autenticazione database IAM con RDS per MySQL

L'autenticazione del database IAM con RDS per MySQL è disponibile in tutte le Regioni per le seguenti versioni:

- RDS per MySQL 8.0: tutte le versioni disponibili
- RDS per MySQL 5.7: tutte le versioni disponibili

Autenticazione database IAM per RDS per PostgreSQL

L'autenticazione del database IAM con RDS per PostgreSQL è disponibile in tutte le Regioni per le seguenti versioni:

- RDS per PostgreSQL 16 — Tutte le versioni disponibili
- RDS per PostgreSQL 15: tutte le versioni disponibili
- RDS per PostgreSQL 14: tutte le versioni disponibili
- RDS per PostgreSQL 13: tutte le versioni disponibili
- RDS per PostgreSQL 12: tutte le versioni disponibili

- RDS per PostgreSQL 11: tutte le versioni disponibili
- RDS per PostgreSQL 10: tutte le versioni disponibili

Regioni e motori DB supportati per l'autenticazione Kerberos in Amazon RDS

L'utilizzo dell'autenticazione Kerberos in Amazon RDS abilita il supporto dell'autenticazione esterna degli utenti del database mediante Kerberos e Microsoft Active Directory. L'utilizzo di Kerberos e Active Directory offre i vantaggi dell'autenticazione Single Sign-On centralizzata degli utenti dei database.

L'autenticazione Kerberos è disponibile per i motori di database seguenti:

- RDS per MariaDB

Sebbene la maggior parte AWS delle regioni sia attiva per impostazione predefinita per il tuo AWS account, alcune regioni vengono attivate solo quando le selezioni manualmente. Queste regioni sono denominate regioni opt-in. Al contrario, le regioni che sono attive per impostazione predefinita, non appena viene creato l' AWS account, vengono chiamate regioni commerciali o semplicemente regioni. Per le regioni opzionali, è necessario utilizzare un servizio principale regionalizzato del modulo. `directoryservice.rds.`*`region_name`*`.amazonaws.com` Ad esempio, per l'Africa (Città del Capo), è necessario aggiungere il service principal `directoryservice.rds.region-af-south-1.amazonaws.com` alla propria politica di fiducia. Per ulteriori informazioni, consulta Autenticazione Kerberos.

Argomenti

- Autenticazione Kerberos con RDS per Db2
- Autenticazione Kerberos con Amazon RDS per MySQL
- Utilizzo dell'autenticazione Kerberos con RDS per Oracle
- Autenticazione Kerberos con RDS per PostgreSQL
- Autenticazione Kerberos con Amazon RDS per SQL Server

Autenticazione Kerberos con RDS per Db2

Le seguenti regioni e versioni del motore sono disponibili per l'autenticazione Kerberos con RDS for Db2.

Regione	RDS per Db2 11.5
Stati Uniti orientali (Ohio)	Tutte le versioni
Stati Uniti orientali (Virginia settentrionale)	Tutte le versioni
Stati Uniti occidentali (California settentrionale)	Tutte le versioni
US West (Oregon)	Tutte le versioni
Africa (Città del Capo)	–
Asia Pacifico (Hong Kong)	–
Asia Pacific (Hyderabad)	–
Asia Pacifico (Giacarta)	–
Asia Pacifico (Melbourne)	–
Asia Pacifico (Mumbai)	Tutte le versioni
Asia Pacifico (Osaka-Locale)	–
Asia Pacific (Seul)	Tutte le versioni
Asia Pacifico (Singapore)	Tutte le versioni
Asia Pacifico (Sydney)	Tutte le versioni
Asia Pacifico (Tokyo)	Tutte le versioni
Canada (Centrale)	Tutte le versioni
Canada occidentale (Calgary)	–
Cina (Pechino)	Tutte le versioni

Regione	RDS per Db2 11.5
Cina (Ningxia)	Tutte le versioni
Europa (Francoforte)	Tutte le versioni
Europa (Irlanda)	Tutte le versioni
Europa (Londra)	Tutte le versioni
Europa (Milano)	–
Europa (Parigi)	–
Europa (Spagna)	–
Europa (Stoccolma)	Tutte le versioni
Europa (Zurigo)	–
Israele (Tel Aviv)	–
Medio Oriente (Bahrein)	–
Medio Oriente (Emirati Arabi Uniti)	–
Sud America (San Paolo)	Tutte le versioni
AWS GovCloud (Stati Uniti orientali)	–
AWS GovCloud (Stati Uniti occidentali)	–

Autenticazione Kerberos con Amazon RDS per MySQL

Di seguito sono riportate le versioni di motore e le regioni disponibili per l'autenticazione Kerberos con RDS per MySQL.

Regione	RDS per MySQL 8.0	RDS per MySQL 5.7	RDS per MySQL 5.6
Stati Uniti orientali (Ohio)	Tutte le versioni	Tutte le versioni	Tutte le versioni
Stati Uniti orientali (Virginia settentri onale)	Tutte le versioni	Tutte le versioni	Tutte le versioni
Stati Uniti occidenta li (California settentri onale)	Tutte le versioni	Tutte le versioni	Tutte le versioni
US West (Oregon)	Tutte le versioni	Tutte le versioni	Tutte le versioni
Africa (Città del Capo)	–	–	–
Asia Pacifico (Hong Kong)	–	–	–
Asia Pacific (Hyderabad)	–	–	–
Asia Pacifico (Giacarta)	–	–	–
Asia Pacifico (Melbourne)	–	–	–
Asia Pacifico (Mumbai)	Tutte le versioni	Tutte le versioni	Tutte le versioni
Asia Pacifico (Osaka-Locale)	–	–	–
Asia Pacific (Seul)	Tutte le versioni	Tutte le versioni	Tutte le versioni
Asia Pacifico (Singapore)	Tutte le versioni	Tutte le versioni	Tutte le versioni

Regione	RDS per MySQL 8.0	RDS per MySQL 5.7	RDS per MySQL 5.6
Asia Pacifico (Sydney)	Tutte le versioni	Tutte le versioni	Tutte le versioni
Asia Pacifico (Tokyo)	Tutte le versioni	Tutte le versioni	Tutte le versioni
Canada (Centrale)	Tutte le versioni	Tutte le versioni	Tutte le versioni
Canada occidentale (Calgary)	–	–	–
Cina (Pechino)	Tutte le versioni	Tutte le versioni	Tutte le versioni
Cina (Ningxia)	Tutte le versioni	Tutte le versioni	Tutte le versioni
Europa (Francoforte)	Tutte le versioni	Tutte le versioni	Tutte le versioni
Europa (Irlanda)	Tutte le versioni	Tutte le versioni	Tutte le versioni
Europa (Londra)	Tutte le versioni	Tutte le versioni	Tutte le versioni
Europa (Milano)	–	–	–
Europa (Parigi)	–	–	–
Europa (Spagna)	–	–	–
Europa (Stoccolma)	Tutte le versioni	Tutte le versioni	Tutte le versioni
Europa (Zurigo)	–	–	–
Israele (Tel Aviv)	–	–	–
Medio Oriente (Bahrein)	–	–	–
Medio Oriente (Emirati Arabi Uniti)	–	–	–

Regione	RDS per MySQL 8.0	RDS per MySQL 5.7	RDS per MySQL 5.6
Sud America (San Paolo)	Tutte le versioni	Tutte le versioni	Tutte le versioni
AWS GovCloud (Stati Uniti orientali)	–	–	–
AWS GovCloud (Stati Uniti occidentali)	–	–	–

Utilizzo dell'autenticazione Kerberos con RDS per Oracle

Di seguito sono riportate le versioni di motore e le regioni disponibili per l'autenticazione Kerberos con RDS per Oracle.

Regione	RDS per Oracle 21c	RDS per Oracle 19c
Stati Uniti orientali (Ohio)	Tutte le versioni	Tutte le versioni
Stati Uniti orientali (Virginia settentrionale)	Tutte le versioni	Tutte le versioni
Stati Uniti occidentali (California settentrionale)	Tutte le versioni	Tutte le versioni
US West (Oregon)	Tutte le versioni	Tutte le versioni
Africa (Città del Capo) (regione opzionale)	Tutte le versioni	Tutte le versioni
Asia Pacifico (Hong Kong) (regione opzionale)	Tutte le versioni	Tutte le versioni
Asia Pacifico (Hyderabad) (regione opzionale)	Tutte le versioni	Tutte le versioni
Asia Pacifico (Giacarta) (regione opzionale)	Tutte le versioni	Tutte le versioni

Regione	RDS per Oracle 21c	RDS per Oracle 19c
Asia Pacifico (Melbourne) (regione opzionale)	Tutte le versioni	Tutte le versioni
Asia Pacifico (Mumbai)	Tutte le versioni	Tutte le versioni
Asia Pacifico (Osaka-Locale)	–	–
Asia Pacific (Seul)	Tutte le versioni	Tutte le versioni
Asia Pacifico (Singapore)	Tutte le versioni	Tutte le versioni
Asia Pacifico (Sydney)	Tutte le versioni	Tutte le versioni
Asia Pacifico (Tokyo)	Tutte le versioni	Tutte le versioni
Canada (Centrale)	Tutte le versioni	Tutte le versioni
Canada occidentale (Calgary)	–	–
Cina (Pechino)	–	–
China (Ningxia)	–	–
Europa (Francoforte)	Tutte le versioni	Tutte le versioni
Europa (Irlanda)	Tutte le versioni	Tutte le versioni
Europa (Londra)	Tutte le versioni	Tutte le versioni
Europa (Milano) (regione opzionale)	Tutte le versioni	Tutte le versioni
Europa (Parigi)	–	–
Europa (Spagna) (regione opzionale)	Tutte le versioni	Tutte le versioni
Europa (Stoccolma)	Tutte le versioni	Tutte le versioni

Regione	RDS per Oracle 21c	RDS per Oracle 19c
Europa (Zurigo) (regione opzionale)	Tutte le versioni	Tutte le versioni
Israele (Tel Aviv) (regione opzionale)	Tutte le versioni	Tutte le versioni
Medio Oriente (Bahrain) (regione opzionale)	Tutte le versioni	Tutte le versioni
Medio Oriente (Emirati Arabi Uniti) (regione opzionale)	Tutte le versioni	Tutte le versioni
Sud America (San Paolo)	Tutte le versioni	Tutte le versioni
AWS GovCloud (Stati Uniti orientali)	Tutte le versioni	Tutte le versioni
AWS GovCloud (Stati Uniti occidentali)	Tutte le versioni	Tutte le versioni

Autenticazione Kerberos con RDS per PostgreSQL

Di seguito sono riportate le versioni di motore e le regioni disponibili per l'autenticazione Kerberos con RDS per PostgreSQL.

Regione	RDS per PostgreSQL 16	RDS per PostgreSQL 15	RDS per PostgreSQL 14	RDS per PostgreSQL 13	RDS per PostgreSQL 12	RDS per PostgreSQL 11	RDS per PostgreSQL 10
Stati Uniti orientali (Ohio)	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni
Stati Uniti orientali (Virginia	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni

Regione	RDS per PostgreSQL 16	RDS per PostgreSQL 15	RDS per PostgreSQL 14	RDS per PostgreSQL 13	RDS per PostgreSQL 12	RDS per PostgreSQL 11	RDS per PostgreSQL 10
settentrionale)							
Stati Uniti occidentali (California settentrionale)	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni
US West (Oregon)	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni
Africa (Città del Capo)	–	–	–	–	–	–	–
Asia Pacifico (Hong Kong)	–	–	–	–	–	–	–
Asia Pacific (Hyderabad)	–	–	–	–	–	–	–
Asia Pacifico (Giacarta)	–	–	–	–	–	–	–

Regione	RDS per PostgreSQL 16	RDS per PostgreSQL 15	RDS per PostgreSQL 14	RDS per PostgreSQL 13	RDS per PostgreSQL 12	RDS per PostgreSQL 11	RDS per PostgreSQL 10
Asia Pacifico (Melbourne)	–	–	–	–	–	–	–
Asia Pacifico (Mumbai)	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni
Asia Pacifico (Osaka-Locale)	–	–	–	–	–	–	–
Asia Pacific (Seul)	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni
Asia Pacifico (Singapore)	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni
Asia Pacifico (Sydney)	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni
Asia Pacifico (Tokyo)	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni
Canada (Centrale)	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni

Regione	RDS per PostgreSQL 16	RDS per PostgreSQL 15	RDS per PostgreSQL 14	RDS per PostgreSQL 13	RDS per PostgreSQL 12	RDS per PostgreSQL 11	RDS per PostgreSQL 10
Canada occidentale (Calgary)	–	–	–	–	–	–	–
Cina (Pechino)	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni
Cina (Ningxia)	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni
Europa (Francoforte)	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni
Europa (Irlanda)	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni
Europa (Londra)	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni
Europa (Milano)	–	–	–	–	–	–	–
Europa (Parigi)	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni
Europa (Spagna)	–	–	–	–	–	–	–
Europa (Stoccolma)	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni

Regione	RDS per PostgreSQL 16	RDS per PostgreSQL 15	RDS per PostgreSQL 14	RDS per PostgreSQL 13	RDS per PostgreSQL 12	RDS per PostgreSQL 11	RDS per PostgreSQL 10
Europa (Zurigo)	–	–	–	–	–	–	–
Israele (Tel Aviv)	–	–	–	–	–	–	–
Medio Oriente (Bahrein)	–	–	–	–	–	–	–
Medio Oriente (Emirati Arabi Uniti)	–	–	–	–	–	–	–
Sud America (San Paolo)	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni
AWS GovCloud (Stati Uniti orientali)	–	–	–	–	–	–	–
AWS GovCloud (Stati Uniti occidentali)	–	–	–	–	–	–	–

Autenticazione Kerberos con Amazon RDS per SQL Server

Di seguito sono riportate le versioni di motore e le regioni disponibili per l'autenticazione Kerberos con RDS per SQL Server.

Regione	RDS per SQL Server 2022	RDS per SQL Server 2019	RDS per SQL Server 2017	RDS per SQL Server 2016	RDS per SQL Server 2014
Stati Uniti orientali (Ohio)	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni
Stati Uniti orientali (Virginia settentrionale)	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni
Stati Uniti occidentali (California settentrionale)	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni
US West (Oregon)	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni
Africa (Città del Capo)	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni
Asia Pacifico (Hong Kong)	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni
Asia Pacific (Hyderabad)	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni
Asia Pacifico (Giacarta)	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni

Regione	RDS per SQL Server 2022	RDS per SQL Server 2019	RDS per SQL Server 2017	RDS per SQL Server 2016	RDS per SQL Server 2014
Asia Pacifico (Melbourne)	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni
Asia Pacifico (Mumbai)	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni
Asia Pacifico (Osaka-Locale)	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni
Asia Pacifico (Seoul)	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni
Asia Pacifico (Singapore)	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni
Asia Pacifico (Sydney)	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni
Asia Pacifico (Tokyo)	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni
Canada (Centrale)	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni
Canada occidentale (Calgary)	–	–	–	–	–
Cina (Pechino)	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni
Cina (Ningxia)	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni

Regione	RDS per SQL Server 2022	RDS per SQL Server 2019	RDS per SQL Server 2017	RDS per SQL Server 2016	RDS per SQL Server 2014
Europa (Francoforte)	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni
Europa (Irlanda)	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni
Europa (Londra)	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni
Europa (Milano)	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni
Europa (Parigi)	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni
Europa (Spagna)	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni
Europa (Stoccolma)	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni
Europa (Zurigo)	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni
Israele (Tel Aviv)	–	–	–	–	–
Medio Oriente (Bahrein)	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni
Medio Oriente (Emirati Arabi Uniti)	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni

Regione	RDS per SQL Server 2022	RDS per SQL Server 2019	RDS per SQL Server 2017	RDS per SQL Server 2016	RDS per SQL Server 2014
Sud America (San Paolo)	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni
AWS GovCloud (Stati Uniti orientali)	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni
AWS GovCloud (Stati Uniti occidentali)	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni	Tutte le versioni

Regioni e motori DB supportati per cluster DB Multi-AZ in Amazon RDS

Un'implementazione di cluster di database Multi-AZ è una modalità di implementazione ad alta disponibilità di Amazon RDS con due istanze database in standby leggibili. Un cluster di database Multi-AZ ha un'istanza database di scrittore e due istanze database di lettore in tre zone di disponibilità separate nella stessa Regione. I cluster di database multi-AZ offrono elevata disponibilità, maggiore capacità per i carichi di lavoro in lettura e minore latenza di scrittura rispetto alle implementazioni di istanze database Multi-AZ. Per ulteriori informazioni, consulta Implementazioni cluster di database multi-AZ.

I cluster di database Multi-AZ non sono disponibili per i seguenti motori:

- RDS per Db2
- RDS per MariaDB
- RDS per Oracle
- RDS per SQL Server

Argomenti

- Cluster di database Multi-AZ con RDS per MySQL
- Cluster di database Multi-AZ con RDS per PostgreSQL

Cluster di database Multi-AZ con RDS per MySQL

Di seguito sono riportate le versioni di motore e le regioni disponibili per i cluster di database Multi-AZ con RDS per MySQL.

Regione	RDS per MySQL 8.0
Stati Uniti orientali (Ohio)	Versione 8.0.28 e successive
Stati Uniti orientali (Virginia settentrionale)	Versione 8.0.28 e successive
Stati Uniti occidentali (California settentrionale)	–
US West (Oregon)	Versione 8.0.28 e successive
Africa (Città del Capo)	Versione 8.0.28 e successive
Asia Pacifico (Hong Kong)	Versione 8.0.28 e successive
Asia Pacific (Hyderabad)	–
Asia Pacifico (Giacarta)	Versione 8.0.28 e successive
Asia Pacifico (Melbourne)	–
Asia Pacifico (Mumbai)	Versione 8.0.28 e successive
Asia Pacifico (Osaka-Locale)	Versione 8.0.28 e successive
Asia Pacifico (Seoul)	Versione 8.0.28 e successive
Asia Pacifico (Singapore)	Versione 8.0.28 e successive
Asia Pacifico (Sydney)	Versione 8.0.28 e successive
Asia Pacifico (Tokyo)	Versione 8.0.28 e successive
Canada (Centrale)	Versione 8.0.28 e successive

Regione	RDS per MySQL 8.0
Canada (Centrale)	Versione 8.0.28 e successive
Canada occidentale (Calgary)	Versione 8.0.28 e successive
Cina (Pechino)	Versione 8.0.28 e successive
Cina (Ningxia)	Versione 8.0.28 e successive
Europa (Francoforte)	Versione 8.0.28 e successive
Europa (Irlanda)	Versione 8.0.28 e successive
Europa (Londra)	Versione 8.0.28 e successive
Europa (Milano)	Versione 8.0.28 e successive
Europa (Parigi)	Versione 8.0.28 e successive
Europa (Spagna)	–
Europa (Stoccolma)	Versione 8.0.28 e successive
Europa (Zurigo)	–
Israele (Tel Aviv)	–
Medio Oriente (Bahrein)	Versione 8.0.28 e successive
Medio Oriente (Emirati Arabi Uniti)	–
Sud America (San Paolo)	Versione 8.0.28 e successive
AWS GovCloud (Stati Uniti orientali)	–
AWS GovCloud (Stati Uniti occidentali)	–

È possibile elencare le versioni disponibili in una regione per una determinata classe di istanze DB utilizzando. AWS CLI Modificate la classe dell'istanza DB per mostrare le relative versioni del motore disponibili.

Per LinuxmacOS, oUnix:

```
aws rds describe-orderable-db-instance-options \
--engine mysql \
--db-instance-class db.r5d.large \
--query '*[]|[?SupportsClusters == `true`].[EngineVersion]'  \
--output text
```

Per Windows:

```
aws rds describe-orderable-db-instance-options ^
--engine mysql ^
--db-instance-class db.r5d.large ^
--query "*[]|[?SupportsClusters == `true`].[EngineVersion]"  ^
--output text
```

Cluster di database Multi-AZ con RDS per PostgreSQL

Di seguito sono riportate le versioni di motore e le regioni disponibili per i cluster di database Multi-AZ con RDS per PostgreSQL.

Regione	RDS per PostgreSQL 16	RDS per PostgreSQL 15	RDS per PostgreSQL 14	RDS per PostgreSQL 13
Stati Uniti orientali (Ohio)	Tutte le versioni di PostgreSQL 16	Tutte le versioni di PostgreSQL 15	Versione 14.5 e versioni successive	Versione 13.4 e versione 13.7 e successive
Stati Uniti orientali (Virginia settentrionale)	Tutte le versioni di PostgreSQL 16	Tutte le versioni di PostgreSQL 15	Versione 14.5 e versioni successive	Versione 13.4 e versione 13.7 e successive
Stati Uniti occidentali (California settentrionale)	–	–	–	–

Regione	RDS per PostgreSQL 16	RDS per PostgreSQL 15	RDS per PostgreSQL 14	RDS per PostgreSQL 13
US West (Oregon)	Tutte le versioni di PostgreSQL 16	Tutte le versioni di PostgreSQL 15	Versione 14.5 e versioni successive	Versione 13.4 e versione 13.7 e successive
Africa (Città del Capo)	Tutte le versioni di PostgreSQL 16	Tutte le versioni di PostgreSQL 15	Versione 14.5 e versioni successive	Versione 13.4 e versione 13.7 e successive
Asia Pacifico (Hong Kong)	Tutte le versioni di PostgreSQL 16	Tutte le versioni di PostgreSQL 15	Versione 14.5 e versioni successive	Versione 13.4 e versione 13.7 e successive
Asia Pacific (Hyderabad)	–	–	–	–
Asia Pacifico (Giacarta)	Tutte le versioni di PostgreSQL 16	Tutte le versioni di PostgreSQL 15	Versione 14.5 e versioni successive	Versione 13.4 e versione 13.7 e successive
Asia Pacifico (Melbourne)	–	–	–	–
Asia Pacifico (Mumbai)	Tutte le versioni di PostgreSQL 16	Tutte le versioni di PostgreSQL 15	Versione 14.5 e versioni successive	Versione 13.4 e versione 13.7 e successive
Asia Pacifico (Osaka-Locale)	Tutte le versioni di PostgreSQL 16	Tutte le versioni di PostgreSQL 15	Versione 14.5 e versioni successive	Versione 13.4 e versione 13.7 e successive
Asia Pacifico (Seoul)	Tutte le versioni di PostgreSQL 16	Tutte le versioni di PostgreSQL 15	Versione 14.5 e versioni successive	Versione 13.4 e versione 13.7 e successive

Regione	RDS per PostgreSQL 16	RDS per PostgreSQL 15	RDS per PostgreSQL 14	RDS per PostgreSQL 13
Asia Pacifico (Singapore)	Tutte le versioni di PostgreSQL 16	Tutte le versioni di PostgreSQL 15	Versione 14.5 e versioni successive	Versione 13.4 e versione 13.7 e successive
Asia Pacifico (Sydney)	Tutte le versioni di PostgreSQL 16	Tutte le versioni di PostgreSQL 15	Versione 14.5 e versioni successive	Versione 13.4 e versione 13.7 e successive
Asia Pacifico (Tokyo)	Tutte le versioni di PostgreSQL 16	Tutte le versioni di PostgreSQL 15	Versione 14.5 e versioni successive	Versione 13.4 e versione 13.7 e successive
Canada (Centrale)	Tutte le versioni di PostgreSQL 16	Tutte le versioni di PostgreSQL 15	Versione 14.5 e versioni successive	Versione 13.4 e versione 13.7 e successive
Canada occidentale (Calgary)	Tutte le versioni di PostgreSQL 16	Tutte le versioni di PostgreSQL 15	Versione 14.5 e versioni successive	Versione 13.4 e versione 13.7 e successive
Cina (Pechino)	Tutte le versioni di PostgreSQL 16	Tutte le versioni di PostgreSQL 15	Versione 14.5 e versioni successive	Versione 13.4 e versione 13.7 e successive
Cina (Ningxia)	Tutte le versioni di PostgreSQL 16	Tutte le versioni di PostgreSQL 15	Versione 14.5 e versioni successive	Versione 13.4 e versione 13.7 e successive
Europa (Francoforte)	Tutte le versioni di PostgreSQL 16	Tutte le versioni di PostgreSQL 15	Versione 14.5 e versioni successive	Versione 13.4 e versione 13.7 e successive
Europa (Irlanda)	Tutte le versioni di PostgreSQL 16	Tutte le versioni di PostgreSQL 15	Versione 14.5 e versioni successive	Versione 13.4 e versione 13.7 e successive

Regione	RDS per PostgreSQL 16	RDS per PostgreSQL 15	RDS per PostgreSQL 14	RDS per PostgreSQL 13
Europa (Londra)	Tutte le versioni di PostgreSQL 16	Tutte le versioni di PostgreSQL 15	Versione 14.5 e versioni successive	Versione 13.4 e versione 13.7 e successive
Europa (Milano)	Tutte le versioni di PostgreSQL 16	Tutte le versioni di PostgreSQL 15	Versione 14.5 e versioni successive	Versione 13.4 e versione 13.7 e successive
Europa (Parigi)	Tutte le versioni di PostgreSQL 16	Tutte le versioni di PostgreSQL 15	Versione 14.5 e versioni successive	Versione 13.4 e versione 13.7 e successive
Europa (Spagna)	–	–	–	–
Europa (Stoccolma)	Tutte le versioni di PostgreSQL 16	Tutte le versioni di PostgreSQL 15	Versione 14.5 e versioni successive	Versione 13.4 e versione 13.7 e successive
Europa (Zurigo)	–	–	–	–
Israele (Tel Aviv)	–	–	–	–
Medio Oriente (Bahrein)	Tutte le versioni di PostgreSQL 16	Tutte le versioni di PostgreSQL 15	Versione 14.5 e versioni successive	Versione 13.4 e versione 13.7 e successive
Medio Oriente (Emirati Arabi Uniti)	–	–	–	–
Sud America (San Paolo)	Tutte le versioni di PostgreSQL 16	Tutte le versioni di PostgreSQL 15	Versione 14.5 e versioni successive	Versione 13.4 e versione 13.7 e successive

Regione	RDS per PostgreSQL 16	RDS per PostgreSQL 15	RDS per PostgreSQL 14	RDS per PostgreSQL 13
AWS GovCloud (Stati Uniti orientali)	–	–	–	
AWS GovCloud (Stati Uniti occidentali)	–	–	–	–

È possibile elencare le versioni disponibili in una regione per una determinata classe di istanze DB utilizzando. AWS CLI Modificate la classe dell'istanza DB per mostrare le relative versioni del motore disponibili.

Per LinuxmacOS, oUnix:

```
aws rds describe-orderable-db-instance-options \
--engine postgres \
--db-instance-class db.r5d.large \
--query '*[]|[?SupportsClusters == `true`].[EngineVersion]'  \
--output text
```

Per Windows:

```
aws rds describe-orderable-db-instance-options ^
--engine postgres ^
--db-instance-class db.r5d.large ^
--query "*[]|[?SupportsClusters == `true`].[EngineVersion]"  ^
--output text
```

Regioni e motori DB supportati per Performance Insights in Amazon RDS

Approfondimenti sulle prestazioni in Amazon RDS espande le sue potenzialità grazie alle funzionalità di monitoraggio esistenti di Amazon RDS a supporto delle operazioni di analisi delle prestazioni del database. Il pannello di controllo di Approfondimenti sulle prestazioni consente di visualizzare il carico del database sull'istanza database Amazon RDS. Puoi anche filtrare il carico in base alle attese, alle istruzioni SQL, agli host o agli utenti. Per ulteriori informazioni, consulta Monitoraggio del carico DB con Performance Insights su Amazon RDS.

Performance Insights è disponibile per tutti i motori RDS DB, ad eccezione di RDS for Db2.

Per i motori DB disponibili, Performance Insights è disponibile con tutte le versioni del motore disponibili e in tutte Regioni AWS.

Per informazioni sul supporto della regione, del motore DB e della classe di istanze per le funzionalità di Performance Insights, vedereSupporto di classe di istanza, regione e motore di database Amazon RDS per funzionalità Performance Insights.

Regioni e motori DB supportati per RDS Custom

Amazon RDS Custom automatizza le attività e le operazioni di amministrazione del database. RDS Custom consente all'amministratore del database di accedere e personalizzare l'ambiente di database e il sistema operativo. Con RDS Custom, è possibile personalizzare per soddisfare i requisiti delle applicazioni legacy, personalizzate e in pacchetti. Per ulteriori informazioni, consulta Utilizzo di Amazon RDS Custom.

RDS Custom è supportato solo per i seguenti motori di database:

Argomenti

- Regioni e motori DB supportati per RDS Custom for Oracle
- Regioni e motori DB supportati per RDS Custom per SQL Server

Regioni e motori DB supportati per RDS Custom for Oracle

Di seguito sono riportate le versioni di motore e le regioni disponibili per RDS Custom per Oracle.

Regione	Oracle Database 19c	Oracle Database 18c	Database Oracle 12c
Stati Uniti orientali (Ohio)	19c con RU/RUR di gennaio 2021 o versioni successive	18c con RU/RUR di gennaio 2021 o versioni successive	12.1 e 12.2 con RU/RUR di gennaio 2021 o versioni successive
Stati Uniti orientali (Virginia settentrionale)	19c con RU/RUR di gennaio 2021 o versioni successive	18c con RU/RUR di gennaio 2021 o versioni successive	12.1 e 12.2 con RU/RUR di gennaio 2021 o versioni successive
Stati Uniti occidentali	–	–	–

Regione	Oracle Database 19c	Oracle Database 18c	Database Oracle 12c
(California settentrionale)			
US West (Oregon)	19c con RU/RUR di gennaio 2021 o versioni successive	18c con RU/RUR di gennaio 2021 o versioni successive	12.1 e 12.2 con RU/RUR di gennaio 2021 o versioni successive
Africa (Città del Capo)	–	–	–
Asia Pacifico (Hong Kong)	–	–	–
Asia Pacifico (Giacarta)	19c con RU/RUR di gennaio 2021 o versioni successive	18c con RU/RUR di gennaio 2021 o versioni successive	12.1 e 12.2 con RU/RUR di gennaio 2021 o versioni successive
Asia Pacifico (Melbourne)	–	–	–
Asia Pacifico (Mumbai)	19c con RU/RUR di gennaio 2021 o versioni successive	18c con RU/RUR di gennaio 2021 o versioni successive	12.1 e 12.2 con RU/RUR di gennaio 2021 o versioni successive
Asia Pacifico (Osaka-Locale)	19c con RU/RUR di gennaio 2021 o versioni successive	18c con RU/RUR di gennaio 2021 o versioni successive	12.1 e 12.2 con RU/RUR di gennaio 2021 o versioni successive
Asia Pacifico (Seoul)	19c con RU/RUR di gennaio 2021 o versioni successive	18c con RU/RUR di gennaio 2021 o versioni successive	12.1 e 12.2 con RU/RUR di gennaio 2021 o versioni successive
Asia Pacifico (Singapore)	19c con RU/RUR di gennaio 2021 o versioni successive	18c con RU/RUR di gennaio 2021 o versioni successive	12.1 e 12.2 con RU/RUR di gennaio 2021 o versioni successive

Regione	Oracle Database 19c	Oracle Database 18c	Database Oracle 12c
Asia Pacifico (Sydney)	19c con RU/RUR di gennaio 2021 o versioni successive	18c con RU/RUR di gennaio 2021 o versioni successive	12.1 e 12.2 con RU/RUR di gennaio 2021 o versioni successive
Asia Pacifico (Tokyo)	19c con RU/RUR di gennaio 2021 o versioni successive	18c con RU/RUR di gennaio 2021 o versioni successive	12.1 e 12.2 con RU/RUR di gennaio 2021 o versioni successive
Canada (Centrale)	19c con RU/RUR di gennaio 2021 o versioni successive	18c con RU/RUR di gennaio 2021 o versioni successive	12.1 e 12.2 con RU/RUR di gennaio 2021 o versioni successive
Canada occidentale (Calgary)	–	–	–
Cina (Pechino)	–	–	–
China (Ningxia)	–	–	–
Europa (Francoforte)	19c con RU/RUR di gennaio 2021 o versioni successive	18c con RU/RUR di gennaio 2021 o versioni successive	12.1 e 12.2 con RU/RUR di gennaio 2021 o versioni successive
Europa (Irlanda)	19c con RU/RUR di gennaio 2021 o versioni successive	18c con RU/RUR di gennaio 2021 o versioni successive	12.1 e 12.2 con RU/RUR di gennaio 2021 o versioni successive
Europa (Londra)	19c con RU/RUR di gennaio 2021 o versioni successive	18c con RU/RUR di gennaio 2021 o versioni successive	12.1 e 12.2 con RU/RUR di gennaio 2021 o versioni successive
Europa (Milano)	19c con RU/RUR di gennaio 2021 o versioni successive	18c con RU/RUR di gennaio 2021 o versioni successive	12.1 e 12.2 con RU/RUR di gennaio 2021 o versioni successive

Regione	Oracle Database 19c	Oracle Database 18c	Database Oracle 12c
Europa (Parigi)	19c con RU/RUR di gennaio 2021 o versioni successive	18c con RU/RUR di gennaio 2021 o versioni successive	12.1 e 12.2 con RU/RUR di gennaio 2021 o versioni successive
Europa (Stoccolma)	19c con RU/RUR di gennaio 2021 o versioni successive	18c con RU/RUR di gennaio 2021 o versioni successive	12.1 e 12.2 con RU/RUR di gennaio 2021 o versioni successive
Israele (Tel Aviv)	–	–	–
Medio Oriente (Bahrein)	–	–	–
Medio Oriente (Emirati Arabi Uniti)	19c con RU/RUR di gennaio 2021 o versioni successive	18c con RU/RUR di gennaio 2021 o versioni successive	12.1 e 12.2 con RU/RUR di gennaio 2021 o versioni successive
Sud America (San Paolo)	19c con RU/RUR di gennaio 2021 o versioni successive	18c con RU/RUR di gennaio 2021 o versioni successive	12.1 e 12.2 con RU/RUR di gennaio 2021 o versioni successive
AWS GovCloud (Stati Uniti orientali)	19c con RU/RUR di gennaio 2021 o versioni successive	18c con RU/RUR di gennaio 2021 o versioni successive	12.1 e 12.2 con RU/RUR di gennaio 2021 o versioni successive
AWS GovCloud (Stati Uniti occidentali)	19c con RU/RUR di gennaio 2021 o versioni successive	18c con RU/RUR di gennaio 2021 o versioni successive	12.1 e 12.2 con RU/RUR di gennaio 2021 o versioni successive

Regioni e motori DB supportati per RDS Custom per SQL Server

È possibile implementare RDS Custom per SQL Server utilizzando una versione del motore fornita da RDS (RPEV) o una versione del motore personalizzato (CEV):

- Se si utilizzai una versione RPEV, includere l'installazione predefinita di Amazon Machine Image (AMI) e SQL Server. Se si personalizza o modifica il sistema operativo, le modifiche potrebbero

non essere persistenti durante l'applicazione di patch, il ripristino di snapshot o il ripristino automatico.

- Se utilizzi una versione CEV, scegliere l'AMI con Microsoft SQL Server o SQL Server preinstallato che viene installata mediante il proprio supporto. Quando utilizzi un CEV AWS fornito, scegli l'immagine Amazon EC2 (AMI) più recente disponibile AWS da, che ha l'aggiornamento cumulativo (CU) supportato da RDS Custom for SQL Server. Con una versione CEV, è possibile personalizzare la configurazione del sistema operativo e di SQL Server in base a specifiche esigenze aziendali.

Le seguenti versioni Regioni AWS e del motore DB sono disponibili per RDS Custom for SQL Server. Il supporto della versione del motore dipende dal fatto che si stia utilizzando RDS Custom per SQL Server con una versione RPEV, una versione CEV fornita da AWS o una versione CEV fornita dal cliente.

Regione	RPEV	AWS ha fornito CEV	CEV fornita dal cliente
Stati Uniti orientali (Ohio)	SQL Server 2022 Enterprise, Standard o Web, con CU9. SQL Server 2019 Enterprise, Standard o Web, con CU8, CU17, CU18, CU20, CU24	SQL Server 2022 Enterprise, Standard o Web, con CU9. SQL Server 2019 Enterprise, Standard o Web, con CU17, CU18, CU20, CU24	SQL Server 2022 Enterprise, Standard o Developer, con CU9. SQL Server 2019 Enterprise, Standard o Developer, con CU17, CU18, CU20, CU24
Stati Uniti orientali (Virginia settentrionale)	SQL Server 2022 Enterprise, Standard o Web, con CU9. SQL Server 2019 Enterprise, Standard o Web, con CU8, CU17, CU18, CU20, CU24	SQL Server 2022 Enterprise, Standard o Web, con CU9. SQL Server 2019 Enterprise, Standard o Web, con CU17, CU18, CU20, CU24	SQL Server 2022 Enterprise, Standard o Developer, con CU9. SQL Server 2019 Enterprise, Standard o Developer, con CU17, CU18, CU20, CU24
Stati Uniti occidentali (California settentrionale)	–	–	–

Regione	RPEV	AWS ha fornito CEV	CEV fornita dal cliente
US West (Oregon)	SQL Server 2022 Enterprise, Standard o Web, con CU9. SQL Server 2019 Enterprise, Standard o Web, con CU8, CU17, CU18, CU20, CU24	SQL Server 2022 Enterprise, Standard o Web, con CU9. SQL Server 2019 Enterprise, Standard o Web, con CU17, CU18, CU20, CU24	SQL Server 2022 Enterprise, Standard o Developer, con CU9. SQL Server 2019 Enterprise, Standard o Developer, con CU17, CU18, CU20, CU24
Africa (Città del Capo)	–	–	–
Asia Pacifico (Hong Kong)	–	–	–
Asia Pacific (Hyderabad)	–	–	–
Asia Pacifico (Giacarta)	–	–	–
Asia Pacifico (Melbourne)	–	–	–
Asia Pacifico (Mumbai)	SQL Server 2022 Enterprise, Standard o Web, con CU9. SQL Server 2019 Enterprise, Standard o Web, con CU8, CU17, CU18, CU20, CU24	SQL Server 2022 Enterprise, Standard o Web, con CU9. SQL Server 2019 Enterprise, Standard o Web, con CU17, CU18, CU20, CU24	SQL Server 2022 Enterprise, Standard o Developer, con CU9. SQL Server 2019 Enterprise, Standard o Developer, con CU17, CU18, CU20, CU24
Asia Pacifico (Osaka-Locale)	–	–	–

Regione	RPEV	AWS ha fornito CEV	CEV fornita dal cliente
Asia Pacific (Seul)	SQL Server 2022 Enterprise, Standard o Web, con CU9. SQL Server 2019 Enterprise, Standard o Web, con CU8, CU17, CU18, CU20, CU24	SQL Server 2022 Enterprise, Standard o Web, con CU9. SQL Server 2019 Enterprise, Standard o Web, con CU17, CU18, CU20, CU24	SQL Server 2022 Enterprise, Standard o Developer, con CU9. SQL Server 2019 Enterprise, Standard o Developer, con CU17, CU18, CU20, CU24
Asia Pacifico (Singapore)	SQL Server 2022 Enterprise, Standard o Web, con CU9. SQL Server 2019 Enterprise, Standard o Web, con CU8, CU17, CU18, CU20, CU24	SQL Server 2022 Enterprise, Standard o Web, con CU9. SQL Server 2019 Enterprise, Standard o Web, con CU17, CU18, CU20, CU24	SQL Server 2022 Enterprise, Standard o Developer, con CU9. SQL Server 2019 Enterprise, Standard o Developer, con CU17, CU18, CU20, CU24
Asia Pacifico (Sydney)	SQL Server 2022 Enterprise, Standard o Web, con CU9. SQL Server 2019 Enterprise, Standard o Web, con CU8, CU17, CU18, CU20, CU24	SQL Server 2022 Enterprise, Standard o Web, con CU9. SQL Server 2019 Enterprise, Standard o Web, con CU17, CU18, CU20, CU24	SQL Server 2022 Enterprise, Standard o Developer, con CU9. SQL Server 2019 Enterprise, Standard o Developer, con CU17, CU18, CU20, CU24
Asia Pacifico (Tokyo)	SQL Server 2022 Enterprise, Standard o Web, con CU9. SQL Server 2019 Enterprise, Standard o Web, con CU8, CU17, CU18, CU20, CU24	SQL Server 2022 Enterprise, Standard o Web, con CU9. SQL Server 2019 Enterprise, Standard o Web, con CU17, CU18, CU20, CU24	SQL Server 2022 Enterprise, Standard o Developer, con CU9. SQL Server 2019 Enterprise, Standard o Developer, con CU17, CU18, CU20, CU24

Regione	RPEV	AWS ha fornito CEV	CEV fornita dal cliente
Canada (Centrale)	SQL Server 2022 Enterprise, Standard o Web, con CU9. SQL Server 2019 Enterprise, Standard o Web, con CU8, CU17, CU18, CU20, CU24	SQL Server 2022 Enterprise, Standard o Web, con CU9. SQL Server 2019 Enterprise, Standard o Web, con CU17, CU18, CU20, CU24	SQL Server 2022 Enterprise, Standard o Developer, con CU9. SQL Server 2019 Enterprise, Standard o Developer, con CU17, CU18, CU20, CU24
Canada occidentale (Calgary)	–	–	–
Cina (Pechino)	–	–	–
China (Ningxia)	–	–	–
Europa (Francoforte)	SQL Server 2022 Enterprise, Standard o Web, con CU9. SQL Server 2019 Enterprise, Standard o Web, con CU8, CU17, CU18, CU20, CU24	SQL Server 2022 Enterprise, Standard o Web, con CU9. SQL Server 2019 Enterprise, Standard o Web, con CU17, CU18, CU20, CU24	SQL Server 2022 Enterprise, Standard o Developer, con CU9. SQL Server 2019 Enterprise, Standard o Developer, con CU17, CU18, CU20, CU24
Europa (Irlanda)	SQL Server 2022 Enterprise, Standard o Web, con CU9. SQL Server 2019 Enterprise, Standard o Web, con CU8, CU17, CU18, CU20, CU24	SQL Server 2022 Enterprise, Standard o Web, con CU9. SQL Server 2019 Enterprise, Standard o Web, con CU17, CU18, CU20, CU24	SQL Server 2022 Enterprise, Standard o Developer, con CU9. SQL Server 2019 Enterprise, Standard o Developer, con CU17, CU18, CU20, CU24

Regione	RPEV	AWS ha fornito CEV	CEV fornita dal cliente
Europa (Londra)	SQL Server 2022 Enterprise, Standard o Web, con CU9. SQL Server 2019 Enterprise, Standard o Web, con CU8, CU17, CU18, CU20, CU24	SQL Server 2022 Enterprise, Standard o Web, con CU9. SQL Server 2019 Enterprise, Standard o Web, con CU17, CU18, CU20, CU24	SQL Server 2022 Enterprise, Standard o Developer, con CU9. SQL Server 2019 Enterprise, Standard o Developer, con CU17, CU18, CU20, CU24
Europa (Milano)	–	–	–
Europa (Parigi)	–	–	–
Europa (Spagna)	–	–	–
Europa (Stoccolma)	SQL Server 2022 Enterprise, Standard o Web, con CU9. SQL Server 2019 Enterprise, Standard o Web, con CU8, CU17, CU18, CU20, CU24	SQL Server 2022 Enterprise, Standard o Web, con CU9. SQL Server 2019 Enterprise, Standard o Web, con CU17, CU18, CU20, CU24	SQL Server 2022 Enterprise, Standard o Developer, con CU9. SQL Server 2019 Enterprise, Standard o Developer, con CU17, CU18, CU20, CU24
Europa (Zurigo)	–	–	–
Israele (Tel Aviv)	–	–	–
Medio Oriente (Bahrein)	–	–	–
Medio Oriente (Emirati Arabi Uniti)	–	–	–

Regione	RPEV	AWS ha fornito CEV	CEV fornita dal cliente
Sud America (San Paolo)	SQL Server 2022 Enterprise, Standard o Web, con CU9. SQL Server 2019 Enterprise, Standard o Web, con CU8, CU17, CU18, CU20, CU24	SQL Server 2022 Enterprise, Standard o Web, con CU9. SQL Server 2019 Enterprise, Standard o Web, con CU17, CU18, CU20, CU24	SQL Server 2022 Enterprise, Standard o Developer, con CU9. SQL Server 2019 Enterprise, Standard o Developer, con CU17, CU18, CU20, CU24
AWS GovCloud (Stati Uniti orientali)	–	–	–
AWS GovCloud (Stati Uniti occidentali)	–	–	–

Regioni e motori DB supportati per Amazon RDS Proxy

Amazon RDS Proxy è un proxy di database completamente gestito e ad alta disponibilità che rende le applicazioni più scalabili mediante il pooling e la condivisione di connessioni di database consolidate. Per ulteriori informazioni, consulta Utilizzo di Server proxy per Amazon RDS.

Il proxy RDS non è disponibile per i seguenti motori:

- RDS per Db2
- RDS per Oracle

Argomenti

- Proxy RDS con RDS per MariaDB
- Proxy RDS con RDS per MySQL
- Proxy RDS con RDS per PostgreSQL
- Server proxy per RDS con RDS per SQL Server

Proxy RDS con RDS per MariaDB

Di seguito sono riportate le versioni di motore e le regioni disponibili per RDS Proxy con RDS per MariaDB.

Regione	RDS per MariaDB 10.11	RDS per MariaDB 10.6	RDS per MariaDB 10.5	RDS per MariaDB 10.4	RDS per MariaDB 10.3
Stati Uniti orientali (Ohio)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Stati Uniti orientali (Virginia settentrionale)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Stati Uniti occidentali (California settentrionale)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
US West (Oregon)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Africa (Città del Capo)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Asia Pacifico (Hong Kong)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili

Regione	RDS per MariaDB 10.11	RDS per MariaDB 10.6	RDS per MariaDB 10.5	RDS per MariaDB 10.4	RDS per MariaDB 10.3
Asia Pacific (Hyderabad)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Asia Pacifico (Giacarta)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Asia Pacifico (Melbourne)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Asia Pacifico (Mumbai)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Asia Pacifico (Osaka-Locale)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Asia Pacifico (Seoul)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Asia Pacifico (Singapore)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Asia Pacifico (Sydney)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Asia Pacifico (Tokyo)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili

Regione	RDS per MariaDB 10.11	RDS per MariaDB 10.6	RDS per MariaDB 10.5	RDS per MariaDB 10.4	RDS per MariaDB 10.3
Canada (Centrale)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Canada occidentale (Calgary)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Cina (Pechino)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Cina (Ningxia)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Europa (Francoforte)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Europa (Irlanda)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Europa (Londra)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Europa (Milano)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Europa (Parigi)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili

Regione	RDS per MariaDB 10.11	RDS per MariaDB 10.6	RDS per MariaDB 10.5	RDS per MariaDB 10.4	RDS per MariaDB 10.3
Europa (Spagna)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Europa (Stoccolma)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Europa (Zurigo)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Israele (Tel Aviv)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Medio Oriente (Bahrein)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Medio Oriente (Emirati Arabi Uniti)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Sud America (San Paolo)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
AWS GovCloud (Stati Uniti orientali)	–	–	–	–	–

Regione	RDS per MariaDB 10.11	RDS per MariaDB 10.6	RDS per MariaDB 10.5	RDS per MariaDB 10.4	RDS per MariaDB 10.3
AWS GovCloud (Stati Uniti occidentali)	–	–	–	–	–

Proxy RDS con RDS per MySQL

Di seguito sono riportate le versioni di motore e le regioni disponibili per RDS Proxy con RDS per MySQL.

Regione	RDS per MySQL 8.0	RDS per MySQL 5.7
Stati Uniti orientali (Ohio)	Tutte le versioni disponibili	Tutte le versioni disponibili
Stati Uniti orientali (Virginia settentrionale)	Tutte le versioni disponibili	Tutte le versioni disponibili
Stati Uniti occidentali (California settentrionale)	Tutte le versioni disponibili	Tutte le versioni disponibili
US West (Oregon)	Tutte le versioni disponibili	Tutte le versioni disponibili
Africa (Città del Capo)	Tutte le versioni disponibili	Tutte le versioni disponibili
Asia Pacifico (Hong Kong)	Tutte le versioni disponibili	Tutte le versioni disponibili
Asia Pacific (Hyderabad)	Tutte le versioni disponibili	Tutte le versioni disponibili
Asia Pacifico (Giacarta)	Tutte le versioni disponibili	Tutte le versioni disponibili
Asia Pacifico (Melbourne)	Tutte le versioni disponibili	Tutte le versioni disponibili
Asia Pacifico (Mumbai)	Tutte le versioni disponibili	Tutte le versioni disponibili
Asia Pacifico (Osaka-Locale)	Tutte le versioni disponibili	Tutte le versioni disponibili

Regione	RDS per MySQL 8.0	RDS per MySQL 5.7
Asia Pacifico (Seoul)	Tutte le versioni disponibili	Tutte le versioni disponibili
Asia Pacifico (Singapore)	Tutte le versioni disponibili	Tutte le versioni disponibili
Asia Pacifico (Sydney)	Tutte le versioni disponibili	Tutte le versioni disponibili
Asia Pacifico (Tokyo)	Tutte le versioni disponibili	Tutte le versioni disponibili
Canada (Centrale)	Tutte le versioni disponibili	Tutte le versioni disponibili
Canada occidentale (Calgary)	Tutte le versioni disponibili	Tutte le versioni disponibili
Cina (Pechino)	Tutte le versioni disponibili	Tutte le versioni disponibili
Cina (Ningxia)	Tutte le versioni disponibili	Tutte le versioni disponibili
Europa (Francoforte)	Tutte le versioni disponibili	Tutte le versioni disponibili
Europa (Irlanda)	Tutte le versioni disponibili	Tutte le versioni disponibili
Europa (Londra)	Tutte le versioni disponibili	Tutte le versioni disponibili
Europa (Milano)	Tutte le versioni disponibili	Tutte le versioni disponibili
Europa (Parigi)	Tutte le versioni disponibili	Tutte le versioni disponibili
Europa (Spagna)	Tutte le versioni disponibili	Tutte le versioni disponibili
Europa (Stoccolma)	Tutte le versioni disponibili	Tutte le versioni disponibili
Europa (Zurigo)	Tutte le versioni disponibili	Tutte le versioni disponibili
Israele (Tel Aviv)	Tutte le versioni disponibili	Tutte le versioni disponibili
Medio Oriente (Bahrein)	Tutte le versioni disponibili	Tutte le versioni disponibili
Medio Oriente (Emirati Arabi Uniti)	Tutte le versioni disponibili	Tutte le versioni disponibili
Sud America (San Paolo)	Tutte le versioni disponibili	Tutte le versioni disponibili

Regione	RDS per MySQL 8.0	RDS per MySQL 5.7
AWS GovCloud (Stati Uniti orientali)	–	–
AWS GovCloud (Stati Uniti occidentali)	–	–

Proxy RDS con RDS per PostgreSQL

Di seguito sono riportate le versioni di motore e le regioni disponibili per RDS Proxy con RDS per PostgreSQL.

Regione	RDS per PostgreSQL 16	RDS per PostgreSQL 15	RDS per PostgreSQL 14	RDS per PostgreSQL 13	RDS per PostgreSQL 12	RDS per PostgreSQL 11	RDS per PostgreSQL 10
Stati Uniti orientali (Ohio)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Stati Uniti orientali (Virginia settentrionale)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Stati Uniti occidentali (California settentrionale)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili

Regione	RDS per PostgreSQL 16	RDS per PostgreSQL 15	RDS per PostgreSQL 14	RDS per PostgreSQL 13	RDS per PostgreSQL 12	RDS per PostgreSQL 11	RDS per PostgreSQL 10
US West (Oregon)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Africa (Città del Capo)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Asia Pacifico (Hong Kong)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Asia Pacific (Hyderabad)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Asia Pacifico (Giacarta)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Asia Pacifico (Melbourne)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Asia Pacifico (Mumbai)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili

Regione	RDS per PostgreSQL 16	RDS per PostgreSQL 15	RDS per PostgreSQL 14	RDS per PostgreSQL 13	RDS per PostgreSQL 12	RDS per PostgreSQL 11	RDS per PostgreSQL 10
Asia Pacifico (Osaka-Locale)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Asia Pacifico (Seoul)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Asia Pacifico (Singapore)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Asia Pacifico (Sydney)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Asia Pacifico (Tokyo)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Canada (Centrale)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Canada occidentale (Calgary)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili

Regione	RDS per PostgreSQL 16	RDS per PostgreSQL 15	RDS per PostgreSQL 14	RDS per PostgreSQL 13	RDS per PostgreSQL 12	RDS per PostgreSQL 11	RDS per PostgreSQL 10
Cina (Pechino)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Cina (Ningxia)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Europa (Francoforte)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Europa (Irlanda)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Europa (Londra)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Europa (Milano)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Europa (Parigi)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili

Regione	RDS per PostgreSQL 16	RDS per PostgreSQL 15	RDS per PostgreSQL 14	RDS per PostgreSQL 13	RDS per PostgreSQL 12	RDS per PostgreSQL 11	RDS per PostgreSQL 10
Europa (Spagna)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Europa (Stoccolma)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Europa (Zurigo)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Israele (Tel Aviv)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Medio Oriente (Bahrein)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Medio Oriente (Emirati Arabi Uniti)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili

Regione	RDS per PostgreSQL 16	RDS per PostgreSQL 15	RDS per PostgreSQL 14	RDS per PostgreSQL 13	RDS per PostgreSQL 12	RDS per PostgreSQL 11	RDS per PostgreSQL 10
Sud America (San Paolo)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
AWS GovCloud (Stati Uniti orientali)	–	–	–	–	–	–	–
AWS GovCloud (Stati Uniti occidentali)	–	–	–	–	–	–	–

Server proxy per RDS con RDS per SQL Server

Di seguito sono riportate le versioni di motore e le regioni disponibili per RDS Proxy con RDS per SQL Server.

Regione	RDS per SQL Server 2019	RDS per SQL Server 2017	RDS per SQL Server 2016	RDS per SQL Server 2014
Stati Uniti orientali (Ohio)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Stati Uniti orientali (Virginia settentrionale)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili

Regione	RDS per SQL Server 2019	RDS per SQL Server 2017	RDS per SQL Server 2016	RDS per SQL Server 2014
Stati Uniti occidentali (California settentrionale)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
US West (Oregon)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Africa (Città del Capo)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Asia Pacifico (Hong Kong)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Asia Pacific (Hyderabad)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Asia Pacifico (Giacarta)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Asia Pacifico (Melbourne)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Asia Pacifico (Mumbai)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Asia Pacifico (Osaka-Locale)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Asia Pacifico (Seoul)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Asia Pacifico (Singapore)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili

Regione	RDS per SQL Server 2019	RDS per SQL Server 2017	RDS per SQL Server 2016	RDS per SQL Server 2014
Asia Pacifico (Sydney)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Asia Pacifico (Tokyo)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Canada (Centrale)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Canada occidentale (Calgary)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Cina (Pechino)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Cina (Ningxia)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Europa (Francoforte)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Europa (Irlanda)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Europa (Londra)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Europa (Milano)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Europa (Parigi)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Europa (Spagna)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili

Regione	RDS per SQL Server 2019	RDS per SQL Server 2017	RDS per SQL Server 2016	RDS per SQL Server 2014
Europa (Stoccolma)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Europa (Zurigo)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Israele (Tel Aviv)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Medio Oriente (Bahrein)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Medio Oriente (Emirati Arabi Uniti)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
Sud America (San Paolo)	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili	Tutte le versioni disponibili
AWS GovCloud (Stati Uniti orientali)	–	–	–	–
AWS GovCloud (Stati Uniti occidentali)	–	–	–	–

Regioni e motori DB supportati per l'integrazione di Secrets Manager con Amazon RDS

Con AWS Secrets Manager, puoi sostituire le credenziali codificate nel codice, incluse le password del database, con una chiamata API a Secrets Manager per recuperare il segreto a livello di codice. Per ulteriori informazioni su Secrets Manager, consulta la Guida per l'utente di AWS Secrets Manager.

Puoi specificare che Amazon RDS gestisca la password dell'utente master in Secrets Manager per un'istanza database Amazon RDS o un cluster di database multi-AZ. RDS genera la password, la memorizza in Secrets Manager e la ruota regolarmente. Per ulteriori informazioni, consulta Gestione delle password con Amazon RDS e AWS Secrets Manager.

L'integrazione di Secrets Manager è supportata per tutti i motori di database RDS e tutte le versioni.

L'integrazione con Secrets Manager è supportata in tutti i Regioni AWS casi tranne i seguenti:

- Canada occidentale (Calgary)
- AWS GovCloud (Stati Uniti orientali)
- AWS GovCloud (Stati Uniti occidentali)

Regioni e motori DB supportati per le integrazioni Zero-ETL di Amazon RDS con Amazon Redshift

Le integrazioni RDS Zero-ETL con Amazon Redshift sono una soluzione completamente gestita per rendere disponibili i dati transazionali in Amazon Redshift dopo averli scritti su un'istanza database Amazon RDS. Per ulteriori informazioni, consulta Utilizzo di integrazioni zero-ETL (anteprima).

Le seguenti regioni e versioni del motore sono disponibili per le integrazioni Zero-ETL con Amazon Redshift.

Regione	RDS per MySQL 8.0
Stati Uniti orientali (Virginia settentrionale)	Versione 8.0.28 e successive
Stati Uniti orientali (Ohio)	Versione 8.0.28 e successive
US West (Oregon)	Versione 8.0.28 e successive
Asia Pacifico (Tokyo)	Versione 8.0.28 e successive
Europa (Irlanda)	Versione 8.0.28 e successive

Funzionalità native del motore in Amazon RDS

I motori di database Amazon RDS supportano anche molte delle caratteristiche e funzionalità native dei motori più comuni. Queste caratteristiche sono diverse dalle caratteristiche native di Amazon RDS elencate in questa pagina. Alcune funzionalità native del motore potrebbero avere supporto o privilegi limitati.

Per ulteriori informazioni sulle funzionalità native del motore, consulta:

- Funzionalità RDS per Db2
- Supporto funzionalità MariaDB su Amazon RDS
- Supporto delle funzionalità MySQL su Amazon RDS
- Funzionalità di RDS for Oracle
- Utilizzo delle caratteristiche di PostgreSQL supportate da Amazon RDS for PostgreSQL
- Funzionalità di Microsoft SQL Server su Amazon RDS

Fatturazione delle istanze database per Amazon RDS

Le istanze Amazon RDS vengono fatturate in base ai componenti seguenti:

- Ore dell'istanza database (all'ora) – In base alla classe di istanza database (ad esempio, db.t2.small or db.m4.large). I prezzi sono calcolati in base a una tariffa oraria, mentre le fatture sono calcolate al secondo e mostrano i valori in formato decimale. L'utilizzo di RDS viene fatturato in incrementi di 1 secondo, con un minimo di 10 minuti. Per ulteriori informazioni, consulta Classi di istanze database.
- Storage (per GiB al mese) – Capacità di storage assegnata all'istanza database. Se la capacità di storage assegnata viene dimensionata nel corso del mese, l'addebito sarà ripartito proporzionalmente. Per ulteriori informazioni, consulta Storage delle istanze di database Amazon RDS.
- Richieste di input/output (I/O) (per 1 milione di richieste): numero totale di richieste di I/O di archiviazione effettuate in un ciclo di fatturazione, solo per l'archiviazione magnetica Amazon RDS.
- Capacità di IOPS allocata (per IOPS al mese) – Quantità di capacità di IOPS allocata, indipendentemente dal consumo di IOPS, solo per archiviazione gp3 di capacità di IOPS allocata (SSD) Amazon RDS e uso generico (SSD). L'archiviazione allocata per i volumi EBS viene fatturata in incrementi di 1 secondo, con un minimo di 10 minuti.
- Storage di backup (per GiB al mese) – Lo storage di backup è lo storage associato ai backup di database automatici e a qualsiasi snapshot DB attivo acquisito. Estendendo il periodo di retention dei backup o creando ulteriori snapshot del database, si aumenta lo storage di backup consumato dal database. La fatturazione per secondo non si applica allo storage di backup (misurato in GB/mese).

  Per ulteriori informazioni, consulta Backup, ripristino ed esportazione dei dati.
- Trasferimento dati (per GB): il trasferimento dati da e verso l'istanza database da o verso Internet e altre regioni AWS.

Amazon RDS offre le opzioni di acquisto seguenti per permetterti di ottimizzare i costi in base alle esigenze:

- Istanze on demand – Paghi all'ora per le ore dell'istanza database usate. I prezzi sono calcolati in base a una tariffa oraria, mentre le fatture sono calcolate al secondo e mostrano i valori in formato decimale. L'utilizzo di RDS ora viene fatturato in incrementi di 1 secondo, con un minimo di 10 minuti.

- Istanze riservate – Prenotando un'istanza database con durata di un anno o tre anni, puoi ricevere uno sconto significativo rispetto ai prezzi delle istanze database on demand. Grazie all'uso delle istanze riservate, è possibile avviare, eliminare e arrestare più istanze entro un'ora e ottenere il vantaggio delle istanze riservate per tutte le istanze.

Per informazioni sui prezzi di Amazon RDS, consulta la pagina Prezzi di Amazon RDS.

Argomenti

- Istanze database on demand per Amazon RDS
- Istanze database riservate per Amazon RDS

Istanze database on demand per Amazon RDS

Le istanze database on demand Amazon RDS vengono fatturate in base alla classe di istanza database, ad esempio db.t3.small o db.m5.large. Per informazioni sui prezzi di Amazon RDS, consulta la pagina del prodotto Amazon RDS.

La fatturazione per un'istanza database ha inizio non appena l'istanza database è disponibile. I prezzi sono calcolati in base a una tariffa oraria, mentre le fatture sono calcolate al secondo e mostrano i valori in formato decimale. L'utilizzo di Amazon RDS viene fatturato in incrementi di un secondo, con un minimo di 10 minuti. In caso di modifica della configurazione fatturabile, come l'elaborazione della scalabilità o la capacità di storage, viene effettuato l'addebito di un minimo di 10 minuti. La fatturazione continua finché l'istanza database non viene terminata, il che avviene quando elimini l'istanza database o in caso di errore dell'istanza database.

Se non vuoi più pagare per l'istanza database, devi arrestare o eliminarla in modo da evitare la fatturazione di altre ore dell'istanza database. Per ulteriori informazioni sugli stati dell'istanza database fatturata, consulta Visualizzazione dello stato dell'istanza database di Amazon RDS .

Istanze database arrestate

Mentre l'istanza database è arrestata, ti viene addebitato lo storage assegnato, incluso lo storage Provisioned IOPS. Ti viene addebitato anche lo storage dei backup, incluso quello per gli snapshot manuali e i backup automatici all'interno della finestra di retention specificata. Non è previsto alcun addebito per le ore dell'istanza database.

Istanze database Multi-AZ

Se specifichi che l'istanza database deve essere un'implementazione Multi-AZ, l'istanza ti verrà addebitata in base ai prezzi delle implementazioni Multi-AZ, pubblicati nella pagina dei prezzi di Amazon RDS.

Istanze database riservate per Amazon RDS

Con le istanze database riservate puoi prenotare un'istanza database per un periodo di uno o tre anni. Le istanze database riservate offrono una notevole riduzione di prezzo rispetto alle istanze database on demand. Le istanze database riservate non sono istanze fisiche, ma piuttosto si tratta di uno sconto sulla fattura applicato all'uso di determinate istanze database nell'account. Gli sconti per le istanze database riservate sono legati al tipo di istanza e alla Regione AWS.

Il processo generale per l'uso delle istanze database riservate è il seguente: prima di tutto ottieni informazioni sulle offerte disponibili per le istanze database riservate, quindi acquisti un'offerta di istanza database riservata e infine ottieni informazioni sulle tue istanze database riservate esistenti.

Panoramica delle istanze database riservate

Quando acquisti un'istanza database riservata in Amazon RDS, acquisti un impegno che ti permette di ottenere una tariffa scontata per un tipo di istanza database specifico, per la durata dell'istanza database riservata. Per usare un'istanza database Amazon RDS riservata, devi creare un'istanza database con una procedura analoga a quella per la creazione di un'istanza on demand.

La nuova istanza database creata deve avere le stesse specifiche dell'istanza database riservata relativamente a quanto segue:

- Regione AWS
- Motore database
- Tipo di istanza database
- Dimensione dell'istanza DB (licenza RDS per Microsoft SQL Server e Amazon RDS for Oracle inclusa)
- Edizione (RDS per SQL Server e RDS per Oracle)
- Tipo di licenza (licenza inclusa o) bring-your-own-license

Se le specifiche della nuova istanza database corrispondono a un'istanza database riservata esistente per il tuo account, viene fatturata la tariffa scontata per l'istanza database riservata. In caso contrario, l'istanza database viene fatturata in base a una tariffa on demand.

Puoi modificare un'istanza database che usi come istanza database riservata. Se la modifica rientra nelle specifiche dell'istanza database riservata, la parte o la totalità dello sconto viene comunque applicata all'istanza database modificata. Se la modifica è al di fuori alle specifiche, ad esempio la

modifica della classe di istanza, lo sconto non viene più applicato. Per ulteriori informazioni, consulta Istanze database riservate con dimensioni flessibili.

Argomenti

Per ulteriori informazioni sulle istanze riservate e sui relativi prezzi, consulta Istanze riservate di Amazon RDS.

Tipi offerta

Le istanze database riservate disponibili sono di tre tipi, ovvero —nessun pagamento anticipato, pagamento anticipato parziale e pagamento anticipato dell'intero costo—per permetterti di ottimizzare i costi di Amazon RDS in base all'utilizzo previsto.

Nessun pagamento anticipato

Questa opzione permette di accedere a un'istanza database riservata senza un pagamento anticipato. L'istanza riservata senza pagamento anticipato viene fatturata applicando una tariffa oraria scontata per ogni ora durante il periodo della prenotazione, indipendentemente dall'utilizzo, e non è richiesto alcun pagamento anticipato. Questa opzione è disponibile solo per le prenotazioni della durata di un anno.

Pagamento anticipato parziale

Questa opzione richiede il pagamento anticipato di una parte dell'istanza database riservata. Le ore rimanenti del periodo di prenotazione vengono fatturate a una tariffa oraria scontata, indipendentemente dall'utilizzo. Questa opzione sostituisce l'opzione precedente per utilizzo pesante.

Pagamento anticipato intero costo

Il pagamento viene effettuato per intero all'inizio del periodo della prenotazione e non vengono addebitati altri costi per il resto del periodo, indipendentemente dal numero di ore di utilizzo.

Se usi la fatturazione consolidata, tutti gli account all'interno dell'organizzazione vengono trattati come se fossero un account unico. Questo significa che tutti gli account di un'organizzazione possono usufruire del vantaggio in termini di costi orari delle istanze database riservate acquistate da un altro account. Per ulteriori informazioni sulla fatturazione consolidata, consulta Istanze database riservate di Amazon RDS nella AWS Guida per l'utente di Billing and Cost Management.

Istanze database riservate con dimensioni flessibili

Quando acquisti un'istanza database riservata, devi specificare la classe di istanza, ad esempio db.r5.large. Per altre informazioni sulle classi di istanza database, consulta Classi di istanze database.

Se hai un'istanza database e devi dimensionarla per aumentarne la capacità, l'istanza database riservata viene applicata automaticamente all'istanza database ridimensionata. Ciò significa che le istanze database riservate vengono applicate automaticamente a tutte le dimensioni di classi di istanze database. Le istanze DB riservate con dimensioni flessibili sono disponibili per le istanze DB con lo stesso motore di database. Regione AWS Le istanze database riservate con dimensioni flessibili sono scalabili solo nel loro tipo di classe istanza. Ad esempio, un'istanza database riservata per db.r5.large è applicabile a db.r5.xlarge, ma non a db.r6g.large, perché db.r5 e db.r6g sono tipi di classe istanza diversi.

I vantaggi delle istanze database riservate si applicano sia alle configurazioni Multi-AZ che a quelle Single-AZ. Flessibilità significa che è possibile spostarsi liberamente tra le configurazioni all'interno dello stesso tipo di classe di istanza database. Ad esempio, è possibile passare da una distribuzione Single-AZ in esecuzione su un'istanza DB di grandi dimensioni (quattro unità normalizzate all'ora) a una distribuzione Multi-AZ in esecuzione su due istanze DB medie (2+2 = 4 unità normalizzate all'ora).

Le istanze database riservate con dimensioni flessibili sono disponibili per i motori di database Amazon RDS seguenti:

- RDS per MariaDB
- RDS for MySQL
- RDS per Oracle, Bring Your Own License
- RDS per PostgreSQL.

La flessibilità delle dimensioni non si applica a RDS per SQL Server e RDS per Oracle License Included.

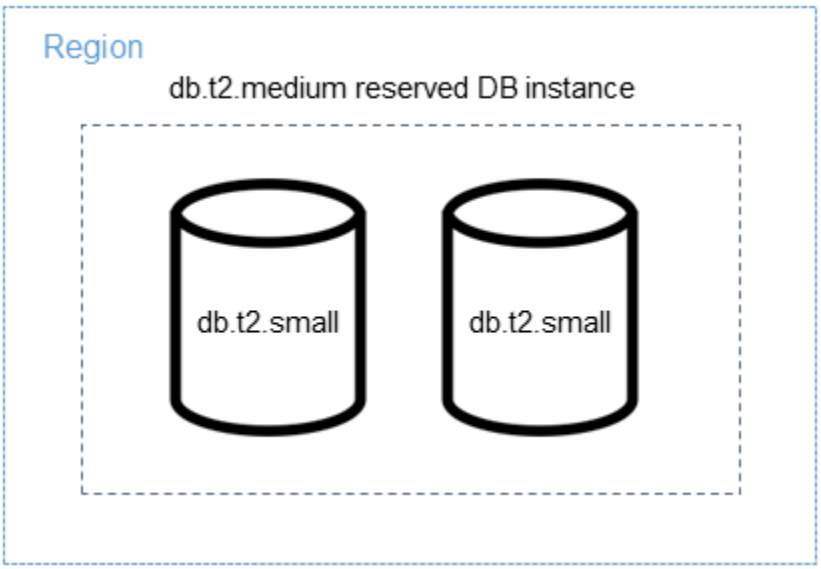
Per i dettagli sull'utilizzo di istanze riservate flessibili in base alle dimensioni con Aurora, consulta Istanze database riservate per Aurora.

È possibile confrontare l'utilizzo per le diverse dimensioni di istanze database riservate usando unità normalizzate all'ora. Ad esempio, un'unità di utilizzo in due istanze database db.r3.large equivale a 8 unità normalizzate di utilizzo all'ora in un'istanza db.r3.small. La tabella seguente mostra il numero di unità normalizzate all'ora per ogni dimensione di istanza database.

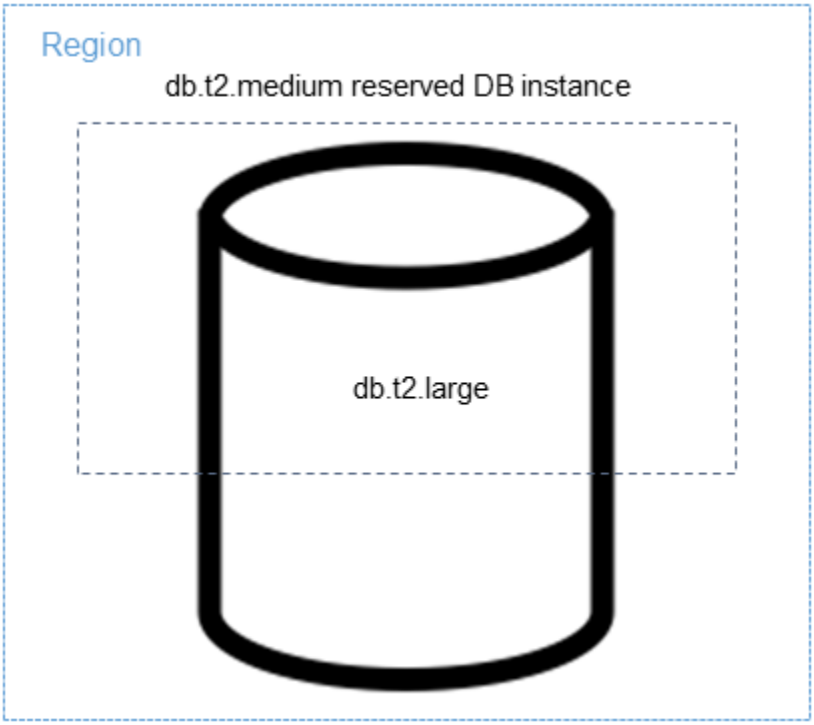
Dimensioni istanza	Unità normalizzate Single-AZ all'ora (implementazione con un'istanza database)	Unità normalizzate dell'istanza database Multi-AZ all'ora (implementazione con un'istanza database e un'istanza in standby)	Unità normalizzate del cluster di database Multi-AZ all'ora (implementazione con un'istanza database e due istanze in standby)
micro	0,5	1	1.5
small	1	2	3
medium	2	4	6
large	4	8	12
xlarge	8	16	24
2xlarge	16	32	48
4xlarge	32	64	96
6xlarge	48	96	144
8xlarge	64	128	192
10xlarge	80	160	240
12xlarge	96	192	288
16xlarge	128	256	384

Dimensioni istanza	Unità normalizzate Single-AZ all'ora (implementazione con un'istanza database)	Unità normalizzate dell'istanza database Multi-AZ all'ora (implementazione con un'istanza database e un'istanza in standby)	Unità normalizzate del cluster di database Multi-AZ all'ora (implementazione con un'istanza database e due istanze in standby)
24xlarge	192	384	576
32xlarge	256	512	768

Supponiamo ad esempio che acquisti un'istanza database riservata `db.t2.medium` e che siano presenti due istanze database `db.t2.small` in esecuzione nel tuo account nella stessa Regione AWS. In questo caso, il vantaggio di fatturazione viene applicato per intero a entrambe le istanze.



In alternativa, se nel tuo account è in esecuzione un'`db.t2.large`istanza nello stesso account Regione AWS, il vantaggio di fatturazione viene applicato al 50 percento dell'utilizzo dell'istanza DB.



Esempio di fatturazione di istanze database riservate

Il prezzo di un'istanza database riservata non prevede uno sconto sui costi associati all'archiviazione, ai backup e all'I/O. È disponibile uno sconto solo sull'utilizzo orario di istanze on demand. L'esempio seguente mostra il costo mensile totale per un'istanza database riservata.

- Una classe di istanza database single-AZ db.r5.large riservata di RDS for MySQL nella regione Stati Uniti orientali (Virginia settentrionale) con l'opzione Nessun anticipo al costo di 0,12 USD per l'istanza o di 90 USD al mese
- 400 GiB di storage General Purpose SSD (gp2) al costo di 0,115 USD per GiB al mese oppure di 45,60 USD al mese
- 600 GiB di storage di backup a 0,095 USD oppure 19 USD al mese (400 GiB gratis)

Sommando tutti questi costi (90 USD + 45,60 USD + 19 USD) all'istanza database riservata, il costo mensile totale è di 154,60 USD.

Se hai scelto di usare un'istanza database on demand anziché un'istanza database riservata, una classe di istanza database Single-AZ db.r5.large di RDS for MySQL nella regione Stati Uniti orientali (Virginia settentrionale) costa 0,1386 USD all'ora o 101,18 USD al mese. Quindi, per un'istanza database on demand, sommando tutte queste opzioni (101,18 USD + 45,60 USD + 19 USD), il costo

mensile totale è di 165,78 USD. Risparmi poco più di $11 al mese utilizzando l'istanza database riservata.

> **Note**
>
> I prezzi citati in questo esempio sono prezzi di esempio e potrebbero non corrispondere ai prezzi effettivi. Per informazioni sui prezzi di Amazon RDS, consulta Prezzi di Amazon RDS.

Istanze database riservate per un cluster di database Multi-AZ

Per acquistare le istanze database riservate equivalenti per un cluster di database Multi-AZ, è possibile eseguire una delle seguenti operazioni:

- Riservare tre istanze database Single-AZ aventi le stesse dimensioni delle istanze nel cluster.
- Riservare un'istanza database Multi-AZ e un'istanza database Single-AZ aventi le stesse dimensioni delle istanze database nel cluster.

Ad esempio, supponi di avere un cluster composto da tre istanze database db.m6gd.large. In questo caso, è possibile acquistare tre istanze database riservate Single-AZ db.m6gd.large oppure un'istanza database riservata Multi-AZ db.m6gd.large e un'istanza database riservata Single-AZ db.m6gd.large. Entrambe queste opzioni sono caratterizzate dallo sconto massimo sulle istanze riservate per il cluster di database Multi-AZ.

In alternativa, puoi utilizzare istanze database con dimensioni flessibili e acquistare un'istanza database più grande per coprire istanze database più piccole in uno o più cluster. Ad esempio, se disponi di due cluster con sei istanze database db.m6gd.large in totale, puoi acquistare tre istanze database riservate Single-AZ db.m6gd.xl. In questo modo vengono riservate tutte e sei le istanze database nei due cluster. Per ulteriori informazioni, consulta Istanze database riservate con dimensioni flessibili.

È possibile riservare istanze database della stessa dimensione delle istanze database nel cluster, ma riservare un numero inferiore di istanze database rispetto al numero totale di istanze database nel cluster. Tuttavia, se si esegue questa operazione, il cluster viene riservato solo parzialmente. Ad esempio, supponi di avere un cluster con tre istanze database db.m6gd.large e di acquistare un'istanza database riservata Multi-AZ db.m6gd.large. In questo caso, il cluster è riservato solo parzialmente, poiché solo due delle tre istanze nel cluster sono coperte da istanze database riservate. L'istanza database rimanente viene addebitata alla tariffa oraria db.m6gd.large on demand.

Per ulteriori informazioni sui cluster di database Multi-AZ, consulta Implementazioni cluster di database multi-AZ.

Eliminazione di un'istanza database riservata

Un'istanza database riservata può essere prenotata con un impegno per un periodo di un anno o di tre anni. Non è possibile annullare un'istanza database riservata. È comunque possibile eliminare un'istanza database coperta da uno sconto per istanza database riservata. Il processo di eliminazione di un'istanza database coperta da uno sconto per istanza database riservata è uguale a quello per l'eliminazione di qualsiasi altra istanza database.

I costi iniziali vengono fatturati indipendentemente dal fatto che si utilizzi le risorse.

Se elimini un'istanza database coperta da uno sconto per istanza database riservata, puoi avviare un'altra istanza database con specifiche compatibili. In questo caso, continuare a usufruire della tariffa scontata durante il periodo della prenotazione (un anno o tre anni).

Utilizzo delle istanze database riservate

Puoi utilizzare l'API AWS Management Console AWS CLI, the e RDS per lavorare con istanze DB riservate.

Console

È possibile utilizzarla AWS Management Console per lavorare con istanze DB riservate, come illustrato nelle seguenti procedure.

Per ottenere informazioni sui prezzi e sulle offerte disponibili per le istanze database riservate

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione scegliere Reserved instances (Istanze riservate).
3. Scegliere Purchase Reserved DB Instance (Acquista istanza database riservata).
4. Per Product description (Descrizione prodotto) scegliere il motore di database e il tipo di licenza.
5. Per DB instance class (Classe istanza database), scegliere la classe di istanza database.
6. In Opzione di implementazione, specifica se desideri un'implementazione Single-AZ o Multi-AZ per le istanze database.

> **Note**
>
> Per acquistare le istanze database riservate equivalenti per un'implementazione Multi-AZ di cluster di database, acquista tre istanze database riservate Single-AZ o un'istanza database riservata Multi-AZ e un'istanza database riservata Single-AZ. Per ulteriori informazioni, consulta Istanze database riservate per un cluster di database Multi-AZ.

7. In Periodo, scegli per quanto tempo riservare l'istanza database.
8. Per Offering type (Tipo di offerta), scegliere il tipo di offerta.

   Dopo aver selezionato il tipo di offerta, vengono visualizzate le informazioni sui prezzi.

> **Important**
>
> Scegliere Cancel (Annulla) per evitare di acquistare un'istanza database riservata con il conseguente addebito dei relativi costi.

Dopo avere ottenuto le informazioni sulle offerte disponibili per le istanze database riservate, puoi basarti su tali informazioni per acquistare un'offerta, come illustrato nella procedura seguente.

Per acquistare un'istanza database riservata

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione scegliere Reserved instances (Istanze riservate).
3. Scegli Purchase reserved DB instance (Acquista istanza database riservata).
4. Per Product description (Descrizione prodotto) scegliere il motore di database e il tipo di licenza.
5. Per DB instance class (Classe istanza database), scegliere la classe di istanza database.
6. In Implementazione Multi-AZ, specifica se desideri un'implementazione Single-AZ o Multi-AZ per le istanze database.

> **Note**
>
> Per acquistare le istanze database riservate equivalenti per un'implementazione Multi-AZ di cluster di database, acquista tre istanze database riservate Single-AZ o un'istanza

database riservata Multi-AZ e un'istanza database riservata Single-AZ. Per ulteriori informazioni, consulta Istanze database riservate per un cluster di database Multi-AZ.

7. Per Term (Periodo), scegliere per quanto tempo prenotare l'istanza database.
8. Per Offering type (Tipo di offerta), scegliere il tipo di offerta.

   Dopo aver selezionato il tipo di offerta, vengono visualizzate le informazioni sui prezzi.
9. (Facoltativo) È possibile assegnare un identificatore alle istanze database riservate acquistate, per tenerne traccia. Per Reserved Id (ID istanza riservata), digitare un identificatore per l'istanza database riservata.
10. Seleziona Invia.

    L'istanza database riservata viene acquistata, quindi visualizzata nell'elenco Reserved instances (Istanze riservate).

Dopo avere acquistato le istanze database riservate, puoi ottenere informazioni sulle tue istanze database riservate come illustrato nella procedura seguente.

Per ottenere informazioni sulle istanze DB riservate per il tuo account AWS

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione scegliere Reserved Instances (Istanze riservate).

   Verranno visualizzate le istanze database riservate per l'account. Per visualizzare informazioni dettagliate su una particolare istanza database riservata, scegli l'istanza nell'elenco. Le informazioni dettagliate su quell'istanza verranno visualizzate nel riquadro dei dettagli nella parte inferiore della console.

AWS CLI

Puoi usarla AWS CLI per lavorare con istanze DB riservate, come mostrato negli esempi seguenti.

Example di recuperare le offerte disponibili per le istanze database riservate

Per ottenere informazioni sulle offerte disponibili di istanze DB riservate, chiamate il AWS CLI comando. `describe-reserved-db-instances-offerings`

```
aws rds describe-reserved-db-instances-offerings
```

Questa chiamata restituisce un output simile al seguente:

```
OFFERING  OfferingId                            Class         Multi-AZ  Duration  Fixed
 Price  Usage Price  Description  Offering Type
OFFERING  438012d3-4052-4cc7-b2e3-8d3372e0e706  db.r3.large   y         1y
 1820.00 USD  0.368 USD    mysql        Partial  Upfront
OFFERING  649fd0c8-cf6d-47a0-bfa6-060f8e75e95f  db.r3.small   n         1y
 227.50 USD  0.046 USD    mysql        Partial  Upfront
OFFERING  123456cd-ab1c-47a0-bfa6-12345667232f  db.r3.small   n         1y
 162.00 USD   0.00 USD    mysql        All      Upfront
    Recurring Charges:   Amount  Currency  Frequency
    Recurring Charges:   0.123   USD       Hourly
OFFERING  123456cd-ab1c-37a0-bfa6-12345667232d  db.r3.large   y         1y
 700.00 USD   0.00 USD    mysql        All      Upfront
    Recurring Charges:   Amount  Currency  Frequency
    Recurring Charges:   1.25    USD       Hourly
OFFERING  123456cd-ab1c-17d0-bfa6-12345667234e  db.r3.xlarge  n         1y
 4242.00 USD   2.42 USD    mysql        No       Upfront
```

Dopo avere ottenuto le informazioni sulle offerte disponibili per le istanze database riservate, puoi basarti su tali informazioni per acquistare un'offerta.

Per acquistare un'istanza DB riservata, utilizza il AWS CLI comando `purchase-reserved-db-instances-offering`con i seguenti parametri:

- `--reserved-db-instances-offering-id` – L'ID dell'offerta da acquistare. Per ottenere l'ID dell'offerta, consulta l'esempio precedente.
- `--reserved-db-instance-id` – Puoi assegnare un identificatore alle istanze database riservate acquistate, per tenerne traccia.

Example di acquistare un'istanza database riservata

L'esempio seguente acquista l'offerta di istanze DB riservate con ID *649fd0c8-cf6d-47a0-bfa6-060f8e75e95f* e assegna l'identificatore di. *MyReservation*

Per, oLinux: macOS Unix

```
aws rds purchase-reserved-db-instances-offering \
```

```
    --reserved-db-instances-offering-id 649fd0c8-cf6d-47a0-bfa6-060f8e75e95f \
    --reserved-db-instance-id MyReservation
```

Per Windows:

```
aws rds purchase-reserved-db-instances-offering ^
    --reserved-db-instances-offering-id 649fd0c8-cf6d-47a0-bfa6-060f8e75e95f ^
    --reserved-db-instance-id MyReservation
```

Il comando restituisce un output simile al seguente:

```
RESERVATION  ReservationId      Class        Multi-AZ  Start Time
 Duration  Fixed Price  Usage Price  Count  State           Description  Offering Type
RESERVATION  MyReservation      db.r3.small  y         2011-12-19T00:30:23.247Z  1y
    455.00 USD   0.092 USD    1      payment-pending  mysql        Partial  Upfront
```

Dopo avere acquistato le istanze database riservate, puoi ottenere informazioni sulle tue istanze database riservate.

Per ottenere informazioni sulle istanze DB riservate per il tuo AWS account, chiama il AWS CLI comando `describe-reserved-db-instances`, come mostrato nell'esempio seguente.

Example di ottenere le istanze DB riservate

```
aws rds describe-reserved-db-instances
```

Il comando restituisce un output simile al seguente:

```
RESERVATION  ReservationId     Class        Multi-AZ  Start Time
 Duration  Fixed Price  Usage Price  Count  State    Description  Offering Type
RESERVATION  MyReservation     db.r3.small  y         2011-12-09T23:37:44.720Z  1y
   455.00 USD   0.092 USD    1      retired  mysql        Partial  Upfront
```

API RDS

Puoi utilizzare l'API RDS per lavorare con istanze database riservate:

- Per ottenere informazioni sulle offerte disponibili per le istanze database riservate, chiamare l'operazione API Amazon RDS `DescribeReservedDBInstancesOfferings`.

- Dopo avere ottenuto le informazioni sulle offerte disponibili per le istanze database riservate, puoi basarti su tali informazioni per acquistare un'offerta. Richiama l'operazione API RDS `PurchaseReservedDBInstancesOffering` con i seguenti parametri:
  - `--reserved-db-instances-offering-id` – L'ID dell'offerta da acquistare.
  - `--reserved-db-instance-id` – Puoi assegnare un identificatore alle istanze database riservate acquistate, per tenerne traccia.
- Dopo avere acquistato le istanze database riservate, puoi ottenere informazioni sulle tue istanze database riservate. Richiama l'operazione API RDS `DescribeReservedDBInstances`.

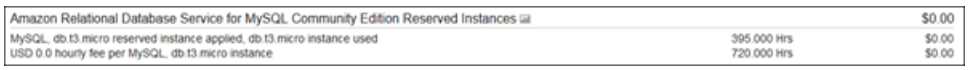
Visualizzazione della fatturazione per le istanze database riservate

Puoi visualizzare la fatturazione per le istanze database riservate nel pannello di controllo di fatturazione nella AWS Management Console.

Per visualizzare la fatturazione di istanze database riservate

1. Accedi alla AWS Management Console.
2. Dal menu account in alto a destra, scegliere Billing Dashboard (Pannello di controllo di fatturazione).
3. Scegliere Dettagli di fatturazione nell'angolo in alto a destra del pannello di controllo.
4. In Costi di servizio AWS , espandere Servizio di database relazionale.
5. Espandi la Regione AWS posizione delle tue istanze DB riservate, ad esempio US West (Oregon).

   Le istanze database riservate e i relativi addebiti orari per il mese corrente sono mostrati sotto Amazon Relational Database Service per ***Motore del database*** Istanze riservate.

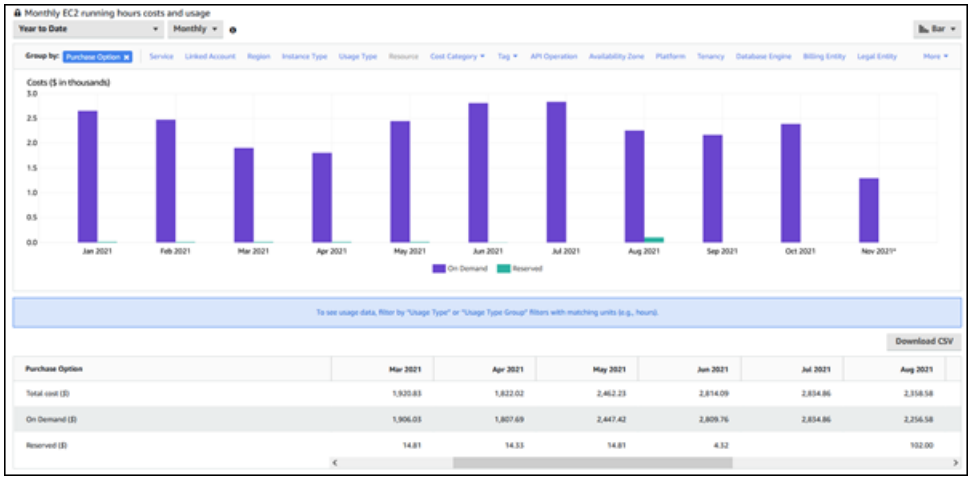
   

   L'istanza database riservata in questo esempio è stata acquistata con pagamento anticipato, quindi non ci sono costi orari.
6. Scegliere l'icona Cost Explorer (grafico a barre) accanto alla voce Istanze riservate.

   Il Cost Explorer visualizza il grafico Costi e utilizzo delle ore di esecuzione mensili di EC2.
7. Cancellare il filtro Usage Type Group (Gruppo tipi di utilizzo) a destra del grafico.

8. Scegliere il periodo di tempo e l'unità di tempo per la quale si desidera esaminare i costi di utilizzo.

   L'esempio seguente mostra i costi di utilizzo per istanze database on-demand e riservate per l'anno in corso per mese.



   I costi delle istanza database riservate da gennaio a giugno 2021 sono oneri mensili per un'istanza con parziale pagamento anticipato, mentre il costo nell'agosto 2021 è un addebito una tantum per un'istanza con pagamento anticipato completo.

   Lo sconto dell'istanza riservata per l'istanza con parziale pagamento anticipato è scaduto nel giugno 2021, ma l'istanza database non è stata eliminata. Dopo la data di scadenza, è stata semplicemente addebitata la tariffa on-demand.

Configurazione di Amazon RDS

Prima di utilizzare Amazon Relational Database Service per la prima volta, completa le seguenti attività.

Argomenti

- Registrati per un Account AWS
- Crea un utente con accesso amministrativo
- Concessione dell'accesso programmatico
- Determinazione dei requisiti
- Fornisci accesso alla istanza database nel VPC creando un gruppo di sicurezza

Se ne possiedi già uno Account AWS, conosci i requisiti di Amazon RDS e preferisci utilizzare le impostazioni predefinite per i gruppi di sicurezza IAM e VPC, passa subito a. Nozioni di base su Amazon RDS

Registrati per un Account AWS

Se non ne hai uno Account AWS, completa i seguenti passaggi per crearne uno.

Per iscriverti a un Account AWS

1. Apri la pagina all'indirizzo https://portal.aws.amazon.com/billing/signup.
2. Segui le istruzioni online.

   Nel corso della procedura di registrazione riceverai una telefonata, durante la quale sarà necessario inserire un codice di verifica attraverso la tastiera del telefono.

   Quando ti iscrivi a un Account AWS, Utente root dell'account AWSviene creato un. L'utente root dispone dell'accesso a tutte le risorse e tutti i Servizi AWS nell'account. Come procedura consigliata in materia di sicurezza, assegnate l'accesso amministrativo a un utente e utilizzate solo l'utente root per eseguire attività che richiedono l'accesso da parte dell'utente root.

AWS ti invia un'e-mail di conferma dopo il completamento della procedura di registrazione. È possibile visualizzare l'attività corrente dell'account e gestire l'account in qualsiasi momento accedendo all'indirizzo https://aws.amazon.com/ e selezionando Il mio account.

Crea un utente con accesso amministrativo

Dopo esserti registrato Account AWS, proteggi Utente root dell'account AWS AWS IAM Identity Center, abilita e crea un utente amministrativo in modo da non utilizzare l'utente root per le attività quotidiane.

Proteggi i tuoi Utente root dell'account AWS

1. Accedi AWS Management Consolecome proprietario dell'account scegliendo Utente root e inserendo il tuo indirizzo Account AWS email. Nella pagina successiva, inserisci la password.

   Per informazioni sull'accesso utilizzando un utente root, consulta la pagina Signing in as the root user della Guida per l'utente di Accedi ad AWS .
2. Abilita l'autenticazione a più fattori (MFA) per l'utente root.

   Per istruzioni, consulta Abilitare un dispositivo MFA virtuale per l'utente Account AWS root (console) nella Guida per l'utente IAM.

Crea un utente con accesso amministrativo

1. Abilita Centro identità IAM.

   Per istruzioni, consulta Abilitazione di AWS IAM Identity Center nella Guida per l'utente di AWS IAM Identity Center .
2. In IAM Identity Center, concedi l'accesso amministrativo a un utente.

   Per un tutorial sull'utilizzo di IAM Identity Center directory come fonte di identità, consulta Configurare l'accesso utente con le impostazioni predefinite IAM Identity Center directory nella Guida per l'AWS IAM Identity Center utente.

Accedi come utente con accesso amministrativo

- Per accedere con l'utente IAM Identity Center, utilizza l'URL di accesso che è stato inviato al tuo indirizzo e-mail quando hai creato l'utente IAM Identity Center.

  Per informazioni sull'accesso utilizzando un utente IAM Identity Center, consulta AWS Accedere al portale di accesso nella Guida per l'Accedi ad AWS utente.

Assegna l'accesso ad altri utenti

1. In IAM Identity Center, crea un set di autorizzazioni che segua la migliore pratica di applicazione delle autorizzazioni con privilegi minimi.

   Per istruzioni, consulta Creare un set di autorizzazioni nella Guida per l'utente.AWS IAM Identity Center
2. Assegna gli utenti a un gruppo, quindi assegna l'accesso Single Sign-On al gruppo.

   Per istruzioni, consulta Aggiungere gruppi nella Guida per l'utente.AWS IAM Identity Center

Concessione dell'accesso programmatico

Gli utenti hanno bisogno di un accesso programmatico se vogliono interagire con l' AWS AWS Management Console esterno di. Il modo per concedere l'accesso programmatico dipende dal tipo di utente che accede. AWS

Per fornire agli utenti l'accesso programmatico, scegli una delle seguenti opzioni.

Quale utente necessita dell'accesso programmatico?	Per	Come
Identità della forza lavoro (Utenti gestiti nel centro identità IAM)	Utilizza credenziali temporane e per firmare le richieste programmatiche agli AWS CLI AWS SDK o alle API. AWS	Segui le istruzioni per l'interfa ccia che desideri utilizzare. • Per la AWS CLI, consulta Configurazione dell'uso AWS IAM Identity Center nella Guida AWS CLI per l'utente.AWS Command Line Interface • Per AWS SDK, strumenti e AWS API, consulta l'autenticazione IAM Identity Center nella Guida di riferimento agli AWS SDK e agli strumenti.

Quale utente necessita dell'accesso programmatico?	Per	Come
IAM	Utilizza credenziali temporane e per firmare le richieste programmatiche agli SDK o alle API AWS CLI. AWS AWS	Segui le istruzioni in Uso delle credenziali temporanee con AWS risorse nella Guida per l'utente IAM.
IAM	(Non consigliato) Utilizza credenziali a lungo termine per firmare le richieste programmatiche agli AWS CLI AWS SDK o alle API. AWS	Segui le istruzioni per l'interfa ccia che desideri utilizzare. • Per la AWS CLI, consulta Autenticazione tramite credenziali utente IAM nella Guida per l'utente.AWS Command Line Interface • Per gli AWS SDK e gli strumenti, consulta Autenticazione tramite credenziali a lungo termine nella Guida di riferimen to agli SDK e agli AWS strumenti. • Per le AWS API, consulta Gestione delle chiavi di accesso per gli utenti IAM nella Guida per l'utente IAM.

Determinazione dei requisiti

L'istanza database rappresenta l'elemento di base di Amazon RDS. In un'istanza database, puoi creare i database. Un'istanza database fornisce un indirizzo di rete che si chiama un endpoint. Le applicazioni utilizzano questo endpoint per connettersi all'istanza database. Quando crei un'istanza database, specifichi dettagli come storage, memoria, motore di database e versione, configurazione di rete, sicurezza e periodi di manutenzione. Controlli l'accesso alla rete a un'istanza database tramite un gruppo di sicurezza.

Prima di creare un'istanza database e un gruppo di sicurezza, devi conoscere le necessità dell'istanza database e della rete. Ecco alcune cose importanti da considerare:

- Requisiti delle risorse – Quali sono i requisiti di memoria e del processore per l'applicazione o il servizio? Utilizzi queste impostazioni per aiutare a determinare quale classe di istanza database da utilizzare. Per specifiche sulle classi di istanza database, consulta Classi di istanze database.
- VPC, sottorete e gruppo di sicurezza– L'istanza database si trova molto probabilmente in un Virtual Private Cloud (VPC). Per connetterti all'istanza database, devi configurare le regole del gruppo di sicurezza. Queste regole sono configurate in maniera diversa in base a quale tipo di VPC utilizzi e come lo utilizzi. Ad esempio, puoi usare un VPC predefinito o un VPC definito dall'utente.

  L'elenco seguente descrive le regole per ogni opzione VPC:

  - VPC predefinito: se il tuo AWS account ha un VPC predefinito nella regione corrente AWS , quel VPC è configurato per supportare le istanze DB. Se specifichi il VPC predefinito quando crei l'istanza database, esegui quanto segue:
    - Assicurati di creare un gruppo di sicurezza VPC che autorizzi le connessioni dall'applicazione o dal servizio all'istanza database Amazon RDS. Utilizza l'opzione Security Group sulla console VPC o AWS CLI per creare gruppi di sicurezza VPC. Per informazioni, consulta Fase 3: creazione di un gruppo di sicurezza VPC.
    - Specifica il gruppo di sottoreti del database predefinito. Se questa è la prima istanza DB creata in questa AWS regione, Amazon RDS crea il gruppo di sottoreti DB predefinito quando crea l'istanza DB.
  - VPC definito dall'utente – Se desideri specificare un VPC definito dall'utente quando crei un'istanza database, tieni presente quanto segue:
    - Assicurati di creare un gruppo di sicurezza VPC che autorizzi le connessioni dall'applicazione o dal servizio all'istanza database Amazon RDS. Utilizza l'opzione Security Group sulla console VPC o AWS CLI per creare gruppi di sicurezza VPC. Per informazioni, consulta Fase 3: creazione di un gruppo di sicurezza VPC.
    - Il VPC deve soddisfare certi requisiti per ospitare istanze database, come avere almeno due sottoreti, ognuna in una zona di disponibilità separata. Per informazioni, consulta VPC di Amazon VPC e Amazon RDS.
    - Assicurati di specificare un gruppo di sottoreti del database che definisce quali sottoreti in quel VPC possono essere utilizzate dall'istanza database. Per informazioni, consulta la sezione del gruppo di sottoreti del database in Uso di un'istanza database in un VPC.

- Elevata disponibilità: hai bisogno di supporto per il failover? Su Amazon RDS, un'implementazione Multi-AZ crea un'istanza database primaria e un'istanza database secondaria in standby in un'altra zona di disponibilità per il supporto per il failover. Consigliamo implementazioni Multi-AZ per carichi di lavoro di produzione per mantenere alta disponibilità. Per scopi di sviluppo e di test, puoi utilizzare un'implementazione che non è Multi-AZ. Per ulteriori informazioni, consulta Configurazione e gestione di un'implementazione multi-AZ.
- Policy IAM: il tuo AWS account dispone di politiche che concedono le autorizzazioni necessarie per eseguire le operazioni di Amazon RDS? Se ti connetti AWS tramite credenziali IAM, il tuo account IAM deve disporre di policy IAM che concedano le autorizzazioni necessarie per eseguire le operazioni di Amazon RDS. Per ulteriori informazioni, consulta Gestione accessi e identità per Amazon RDS.
- Porte aperte: tramite quale porta TCP/IP ascolta il database? I firewall in alcune aziende possono bloccare le connessioni alla porta predefinita del motore di database. Se il firewall dell'azienda blocca la porta predefinita, seleziona un'altra porta per la nuova istanza database. Quando si crea un'istanza database che ascolta su una porta che specifichi, puoi cambiare la porta modificando l'istanza database.
- AWS Regione: in quale AWS regione vuoi inserire il tuo database? Avere il database in prossimità ravvicinata all'applicazione o servizio web può ridurre la latenza di rete. Per ulteriori informazioni, consulta Regioni, zone di disponibilità e Local Zones.
- Sottosistema di dischi dei database: quali sono i requisiti di archiviazione? Amazon RDS fornisce tre tipi di archiviazione:
  - General Purpose (SSD)
  - Provisioned IOPS (PIOPS)
  - Magnetico (noto anche come memoria standard)

  Per ulteriori informazioni sullo storage Amazon RDS, consulta Storage delle istanze di database Amazon RDS.

Quando disponi delle informazioni, devi creare un gruppo di sicurezza e istanza database, continua alla fase successiva.

Fornisci accesso alla istanza database nel VPC creando un gruppo di sicurezza

I gruppi di sicurezza VPC forniscono l'accesso alle istanze database in un VPC. Fungono da firewall per l'istanza database associata, controllando sia il traffico in entrata che in uscita a livello di istanza database. Le istanze database vengono create come impostazione predefinita con un firewall e un gruppo di sicurezza predefinito che protegge l'istanza database.

Prima di connetterti a un'istanza database, devi aggiungere regole al gruppo di sicurezza che consentono di connettersi. Utilizza le informazione di rete e di configurazione per creare regole per permettere l'accesso all'istanza database.

Supponiamo, ad esempio, di avere un'applicazione che accede a un database nell'istanza database in un VPC. In questo caso, devi aggiungere una regola TCP personalizzata che specifichi l'intervallo di porte e gli indirizzi IP che l'applicazione utilizza per accedere al database. Se hai un'applicazione in un'istanza Amazon EC2, puoi utilizzare il gruppo di sicurezza configurato per l'istanza Amazon EC2.

Puoi configurare la connettività tra un'istanza Amazon EC2 e un'istanza database quando crei l'istanza database. Per ulteriori informazioni, consulta Configurazione della connettività di rete automatica con un'istanza EC2.

> **Tip**
>
> Quando crei un'istanza database, puoi configurare automaticamente la connettività di rete tra un'istanza Amazon EC2 e un'istanza database. Per ulteriori informazioni, consulta Configurazione della connettività di rete automatica con un'istanza EC2.

Per informazioni sugli scenari comuni per l'accesso a un'istanza database, consult Scenari per accedere a un'istanza database in un VPC.

Per creare un gruppo di sicurezza VPC

1. Accedi AWS Management Console e apri la console Amazon VPC all'indirizzo https://console.aws.amazon.com/vpc.

> **Note**
>
> Assicurati di essere nella console VPC, non nella console RDS.

2. Nell'angolo in alto a destra di AWS Management Console, scegli la AWS regione in cui desideri creare il gruppo di sicurezza VPC e l'istanza DB. Nell'elenco delle risorse Amazon VPC per quella regione AWS , dovresti vedere almeno un VPC e diverse sottoreti. In caso contrario, non disponi di un VPC predefinito in quella AWS regione.
3. Fare clic su Security Groups (Gruppi di sicurezza) nel pannello di navigazione.
4. Scegliere Create Security Group (Crea gruppo di sicurezza).

   Viene visualizzata la pagina Create security group (Crea gruppo di sicurezza).
5. In Basic details (Dettagli di base), immettere il Security group name (Nome del gruppo di sicurezza) e la Description (Descrizione). Per VPC, seleziona il VPC nel quale desideri creare l'istanza database.
6. Per Inbound rules (Regole in entrata), scegli Add rule (Aggiungi regola).
   a. Per Type (Tipo), scegliere Custom TCP (TCP personalizzato).
   b. Per Port Range (Intervallo porte), digita il valore della porta da utilizzare per l’istanza database.
   c. Per Source (Origine), seleziona il nome del gruppo di sicurezza o digita l'intervallo dell'indirizzo IP (valore CIDR) da dove accedi all'istanza database. Se scegli My IP (Il mio IP), questo consente l'accesso all'istanza database dall'indirizzo IP rilevato nel browser.
7. Se occorre aggiungere altri indirizzi IP o intervalli di porta diversi, scegliere Add rule (Aggiungi regola) e immettere le informazioni relative alla regola.
8. (Facoltativo) In Outbound Rules (Regole in uscita), aggiungi regole per il traffico in uscita. Come impostazione predefinita, tutto il traffico in uscita è permesso.
9. Scegliere Create Security Group (Crea gruppo di sicurezza).

Puoi utilizzare il gruppo di sicurezza VPC appena creato come gruppo di sicurezza per l'istanza database al momento della creazione.

> **Note**
>
> Se utilizzi un VPC predefinito, viene creato un gruppo di sottoreti predefinito che include tutte le sottoreti VPC. Quando si crea un'istanza database, è possibile selezionare il VPC predefinito e utilizzare default (predefinito) per DB Subnet Group (Gruppo di sottoreti del database).

Quando hai completato i requisiti di configurazione, puoi creare un'istanza database utilizzando i requisiti e il gruppo di sicurezza. Per farlo, segui le istruzioni in Creazione di un'istanza database Amazon RDS. Per informazioni su come iniziare con la creazione di un'istanza database che utilizzi un motore DB specifico, consulta la documentazione pertinente nella tabella seguente.

Motore di database	Documentazione
MariaDB	Creazione e connessione di un'istanza database MariaDB
Microsoft SQL Server	Creazione e connessione a un'istanza database Microsoft SQL Server
MySQL	Creazione e connessione di un'istanza database MySQL
Oracle	Creazione e connessione a un'istanza database Oracle
PostgreSQL	Creazione e connessione di un'istanza database PostgreSQL

> **Note**
>
> Se non è possibile connettersi a un'istanza database dopo averla creata, consulta le informazioni sulla risoluzione dei problemi in Impossibile connettersi all'istanza database di Amazon RDS.

Nozioni di base su Amazon RDS

Negli esempi seguenti, sono disponibili informazioni su come creare ed eseguire la connessione a un'istanza database utilizzando Amazon Relational Database Service (Amazon RDS). È possibile creare un'istanza DB che utilizza Db2, MariaDB, MySQL, Microsoft SQL Server, Oracle o PostgreSQL.

> **Important**
>
> Devi completare le attività indicate su Configurazione di Amazon RDS prima di poter creare o connetterti a un'istanza database.

Creare un'istanza database e connettersi a un database su un'istanza database sono operazioni che variano leggermente in base al motore di database utilizzato. Scegli uno dei motori di database seguenti che desideri utilizzare per informazioni dettagliate sulla creazione e connessione all'istanza database. Dopo aver creato ed esserti connesso alla tua istanza database, visualizzi delle istruzioni che ti aiutano a eliminare l'istanza database.

Argomenti

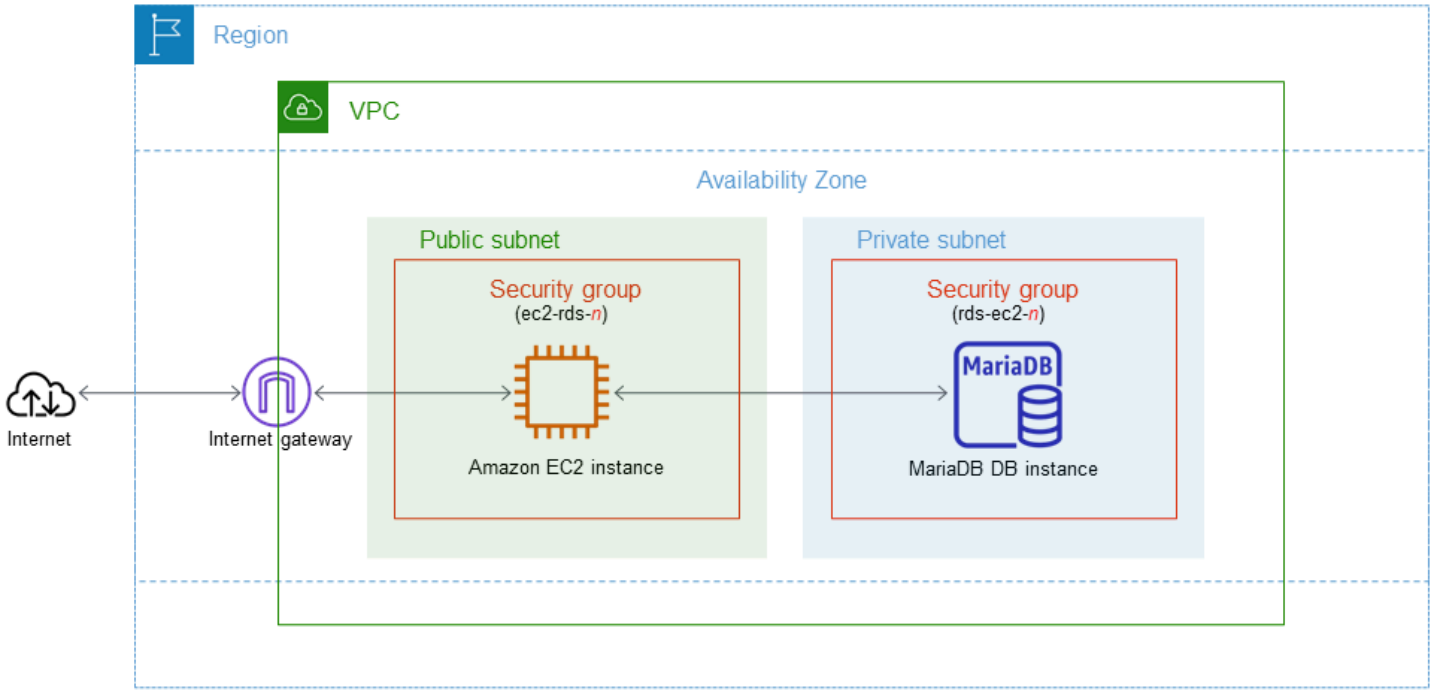
Creazione e connessione di un'istanza database MariaDB

Questo tutorial illustra come creare un'istanza EC2 e un'istanza database RDS per MariaDB. Il tutorial mostra come accedere all'istanza database dall'istanza EC2 utilizzando il client MySQL standard. Come best practice, questo tutorial spiega come creare un'istanza database privata in un cloud privato virtuale (VPC). Nella maggior parte dei casi, le risorse presenti nello stesso VPC, come le istanze EC2, possono accedere all'istanza database, mentre le risorse esterne al VPC non possono accedervi.

Dopo aver completato il tutorial, è presente una sottorete pubblica e una privata in ogni zona di disponibilità del VPC. In una zona di disponibilità, l'istanza EC2 si trova nella sottorete pubblica mentre l'istanza database si trova nella sottorete privata.

> **Important**
>
> Non ci sono costi per la creazione di un Account AWS. Tuttavia, completando l'esercitazione, potresti incorrere in costi per le risorse utilizzate. È possibile eliminare queste risorse dopo aver completato l'esercitazione se non sono più necessarie.

Il seguente diagramma illustra la configurazione al completamento del tutorial.



Questo tutorial ti consente di creare le tue risorse utilizzando uno dei seguenti metodi:

1. Usa AWS Management Console - Fase 1: creazione di un'istanza EC2 e Fase 2: creazione di un'istanza database MariaDB
2. Utilizzare AWS CloudFormation per creare l'istanza del database e l'istanza EC2 - (Facoltativo) Crea VPC, istanza EC2 e istanza MariaDB utilizzando AWS CloudFormation

Il primo metodo utilizza Easy create per creare un'istanza privata di MariadB DB con. AWS Management Console Qui, si specificano solo il tipo di motore DB, la dimensione dell'istanza DB e l'identificatore dell'istanza DB. Easy create (Creazione rapida) utilizza l'impostazione predefinita per altre opzioni di configurazione.

Se invece utilizzi Standard create, puoi specificare più opzioni di configurazione quando crei un'istanza DB. Queste opzioni includono impostazioni per la disponibilità, la sicurezza, i backup e la manutenzione. Per creare un'istanza database pubblica, è necessario utilizzare la Creazione standard. Per informazioni, consulta Creazione di un'istanza database Amazon RDS.

Argomenti

Prerequisiti

Prima di iniziare, completa le fasi descritte in questa sezione:

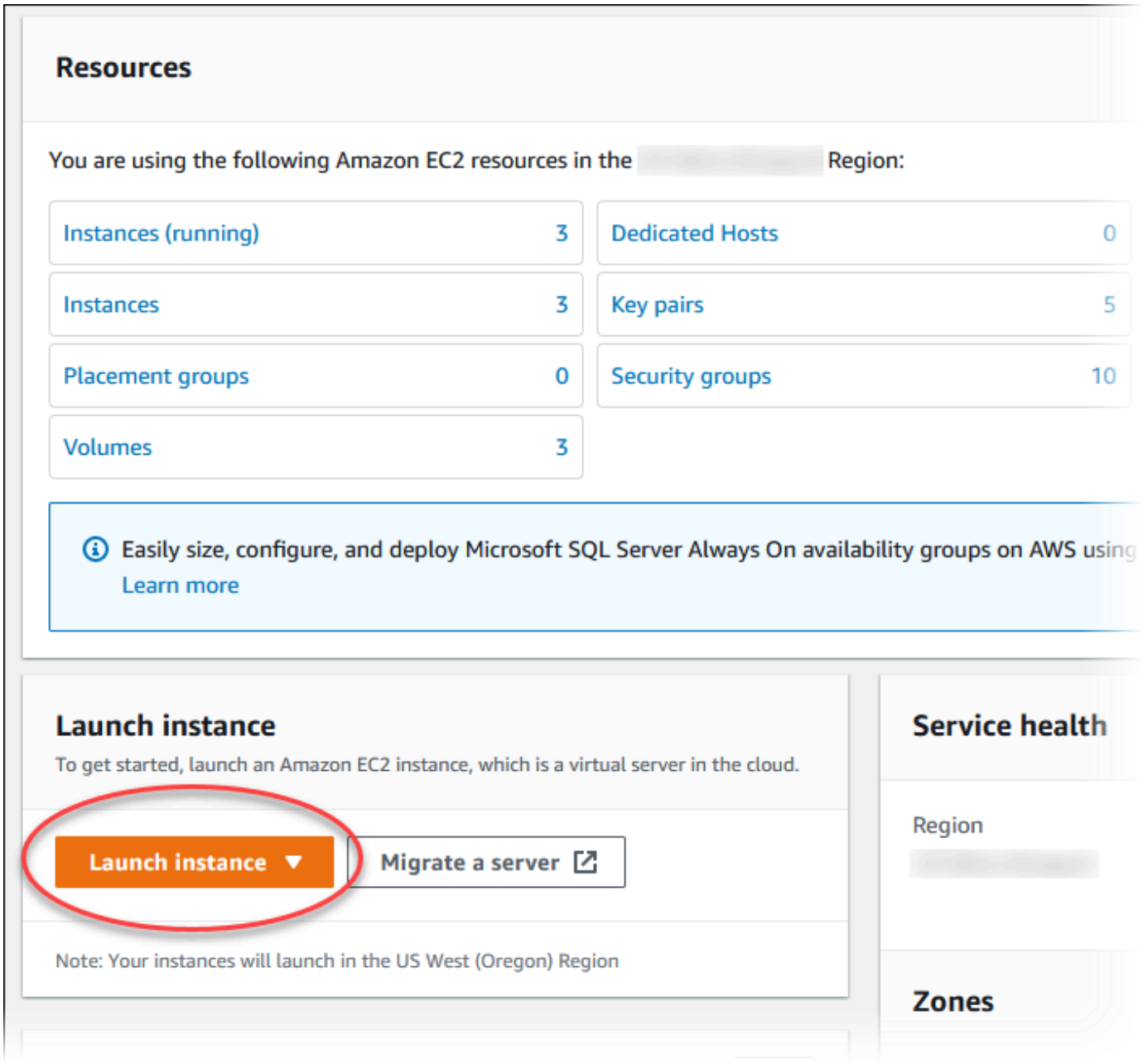
- Registrati per un Account AWS
- Crea un utente con accesso amministrativo

Fase 1: creazione di un'istanza EC2

Crea un'istanza Amazon EC2 da utilizzare per connetterti al database.

Per creare un'istanza EC2

1. Accedi AWS Management Console e apri la console Amazon EC2 all'indirizzo https://console.aws.amazon.com/ec2/.
2. Nell'angolo in alto a destra di AWS Management Console, scegli l'istanza EC2 Regione AWS in cui desideri creare l'istanza EC2.
3. Seleziona Pannello di controllo EC2, quindi Avvia istanza, come visualizzato di seguito.

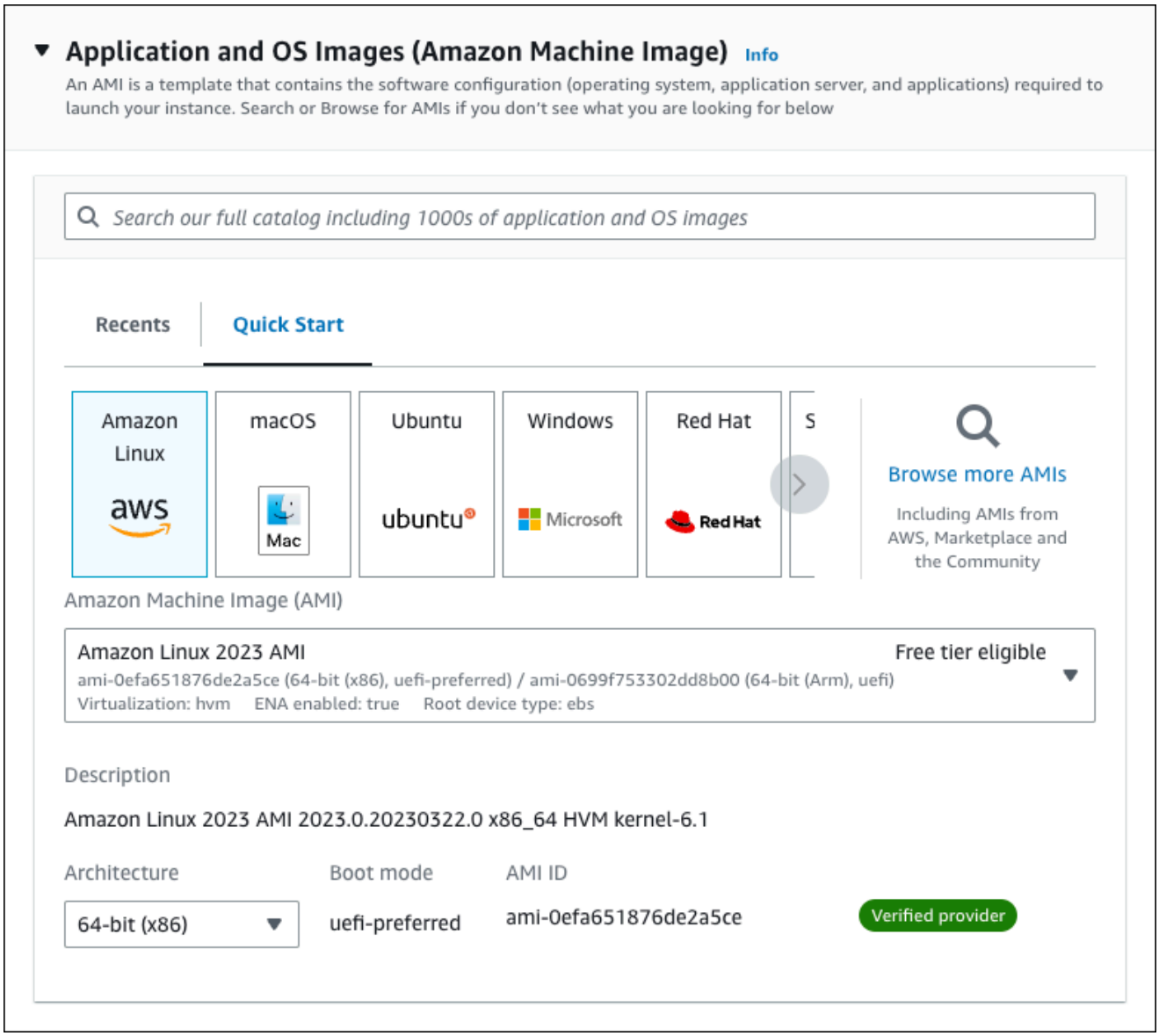


Viene visualizzata la pagina Avvia un'istanza.

4. Scegli le seguenti impostazioni nella pagina Avvia un'istanza.

   a. Nell'area Name and tags (Nome e tag), in Name (Nome) inserisci **`ec2-database-connect`**.

   b. In Immagini applicazione e sistema operativo (Amazon Machine Image), scegli Amazon Linux, quindi AMI Amazon Linux 2023. Mantieni le selezioni predefinite per le altre opzioni.



   c. In Instance type (Tipo di istanza), scegli t2.micro.

   d. In Key pair (login) (Coppia di chiavi (login), per Key pair name (Nome della coppia di chiavi), scegli una coppia di chiavi esistente. Per creare una nuova coppia di chiavi per l'istanza Amazon EC2, scegli Create new key pair (Crea nuova coppia di chiavi) e quindi utilizza la finestra Create key pair (Crea coppia di chiavi) per crearla.

Per ulteriori informazioni sulla creazione di una nuova coppia di chiavi, consulta Creazione di una coppia di chiavi nella Guida per l'utente di Amazon EC2 per istanze Linux.

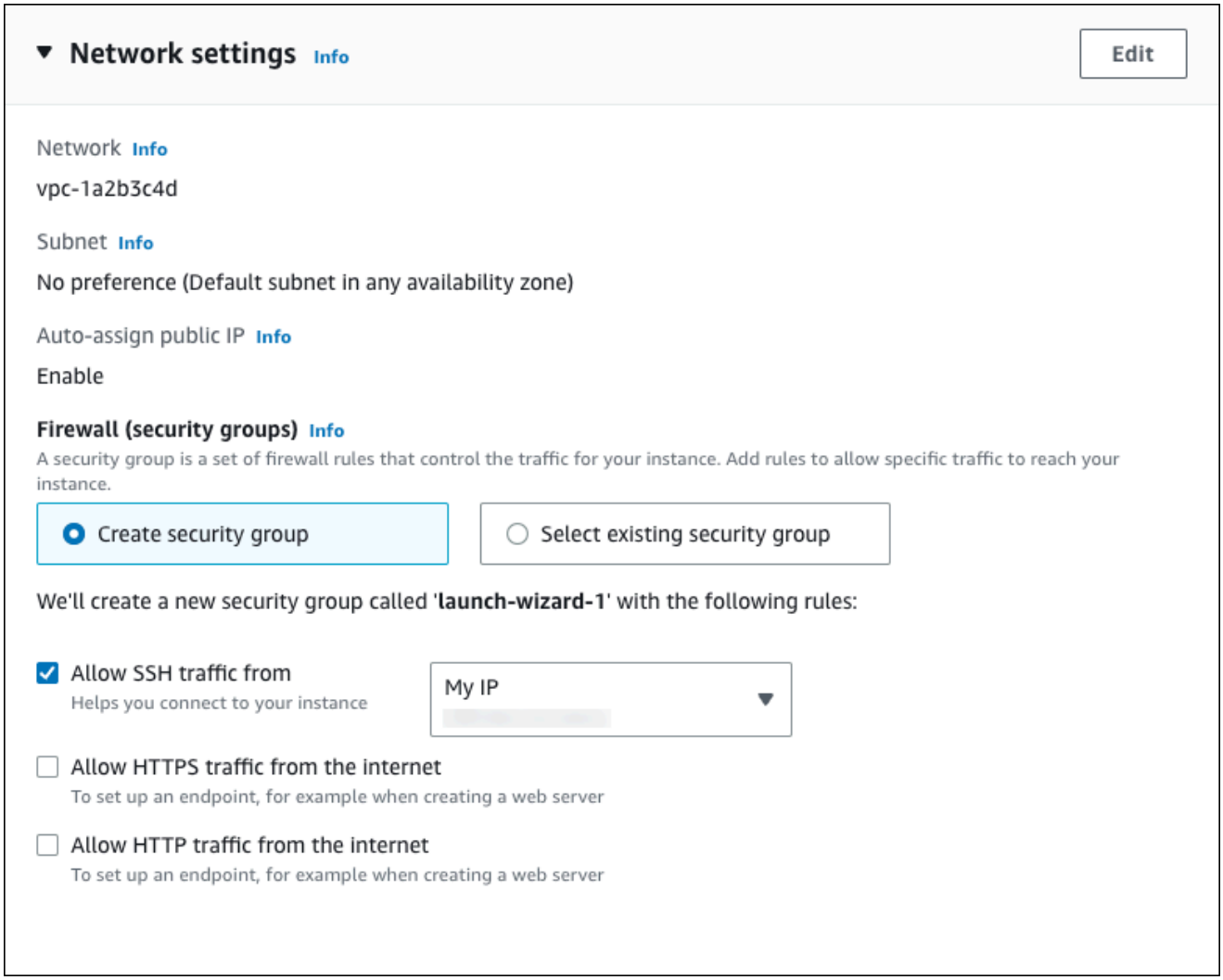
e. In Consenti traffico SSH, nell'area Impostazioni di rete scegliere l'origine delle connessioni SSH all'istanza EC2.

   È possibile scegliere My IP (Il mio IP) se l'indirizzo IP visualizzato è corretto per le connessioni SSH. In caso contrario, è possibile determinare l'indirizzo IP da utilizzare per connettersi alle istanze EC2 nel VPC utilizzando Secure Shell (SSH). Per determinare l'indirizzo IP pubblico, in una finestra o una scheda del browser diversa, è possibile utilizzare il servizio all'indirizzo https://checkip.amazonaws.com. Un esempio di indirizzo IP è 192.0.2.1/32.

   In molti casi, è possibile eseguire la connessione tramite un fornitore di servizi Internet (ISP) o con la protezione di un firewall senza un indirizzo IP statico. In tal caso, accertati di determinare l'intervallo di indirizzi IP utilizzati dai computer client.

   > **Warning**
   >
   > Se utilizzi `0.0.0.0/0` per l'accesso SSH, consenti a tutti gli indirizzi IP di accedere alle istanze EC2 pubbliche utilizzando SSH. Questo approccio è accettabile per un breve periodo di tempo in un ambiente di test, ma non è sicuro per gli ambienti di produzione. In produzione, autorizza solo un determinato indirizzo IP o un intervallo di indirizzi per accedere alle istanze EC2 utilizzando SSH.

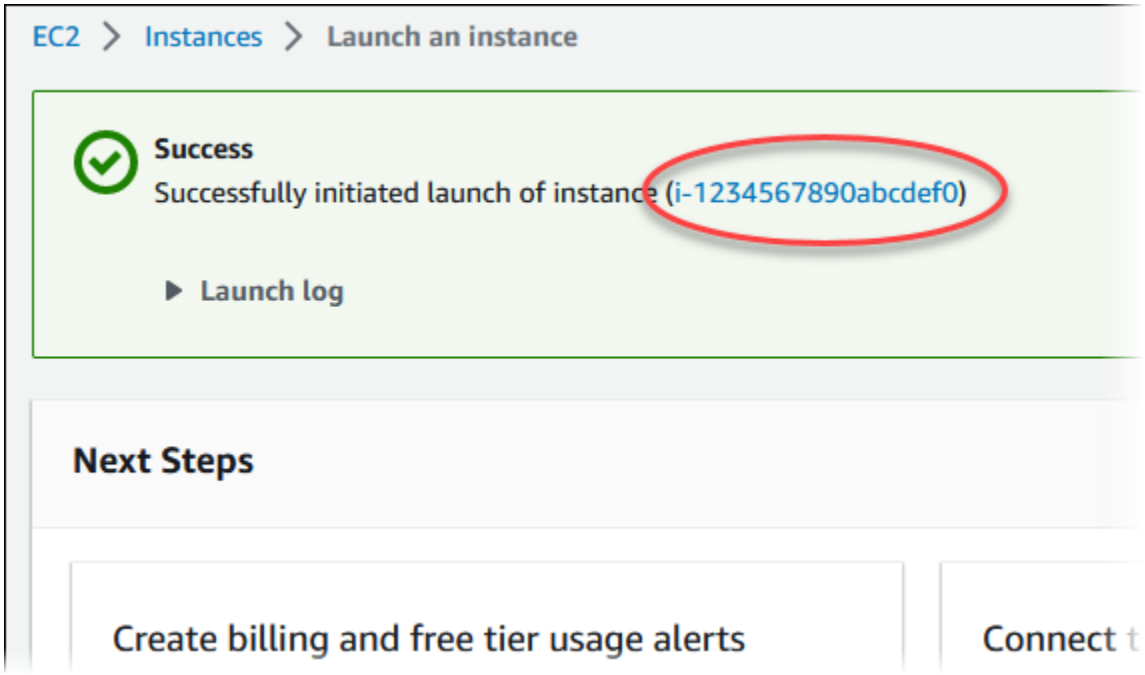
   L'immagine seguente mostra un esempio della sezione Impostazioni di rete.



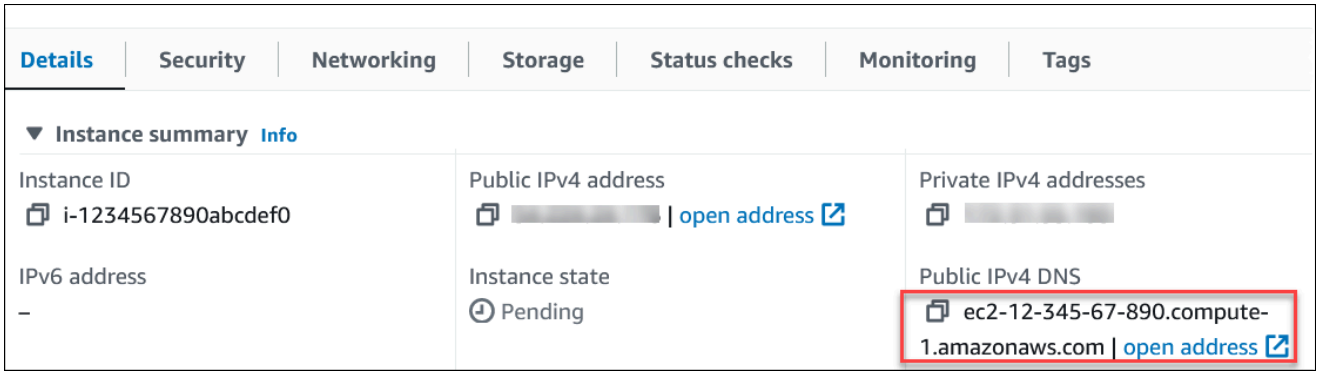
f. Lascia i valori predefiniti per le sezioni rimanenti.

g. Analizza un riepilogo della configurazione dell'istanza EC2 nel pannello Riepilogo e, quando è tutto pronto, scegli Avvia istanza.

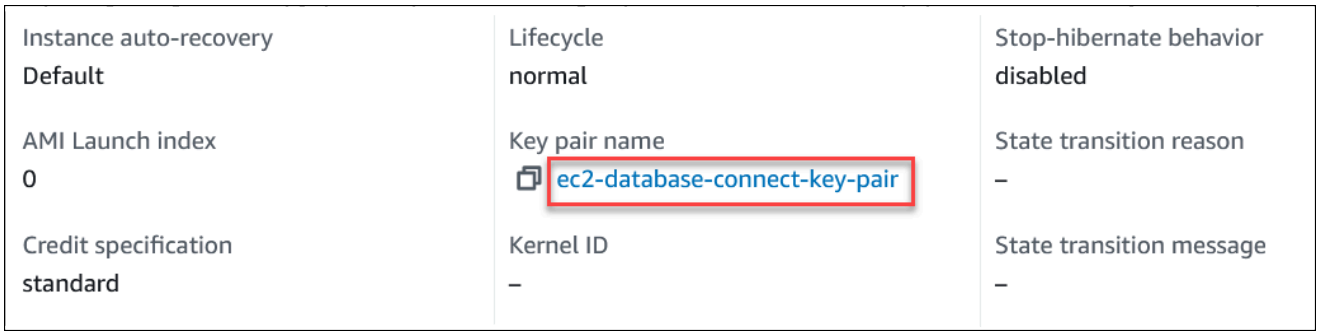
5. Nella pagina Stato avvio prendi nota dell'identificatore per la nuova istanza EC2, ad esempio: `i-1234567890abcdef0`.



6. Scegli l'identificatore dell'istanza EC2 per aprire l'elenco delle istanze EC2, quindi seleziona l'istanza EC2.
7. Nella scheda Dettagli, annota i seguenti valori, necessari quando ti connetti tramite SSH:
   a. In Riepilogo istanza, annota il valore visualizzato in DNS IPv4 pubblico.



   b. In Dettagli istanza, annota il valore visualizzato in Nome coppia di chiavi.



8. Attendi che Stato dell'istanza diventi In esecuzione per l'istanza EC2 prima di continuare.

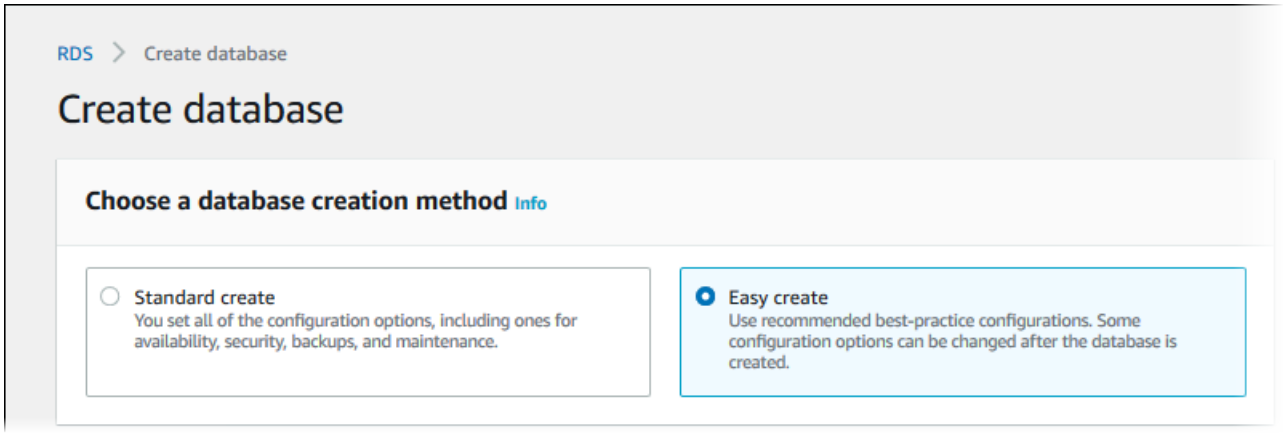
Fase 2: creazione di un'istanza database MariaDB

L'istanza database rappresenta l'elemento di base di Amazon RDS. Questo è l'ambiente dove esegui i tuoi database MariaDB.

Per questo esempio, utilizzi la Creazione semplice per creare un'istanza database che esegue un motore di database MariaDB con una classe di istanza database db.t3.micro.

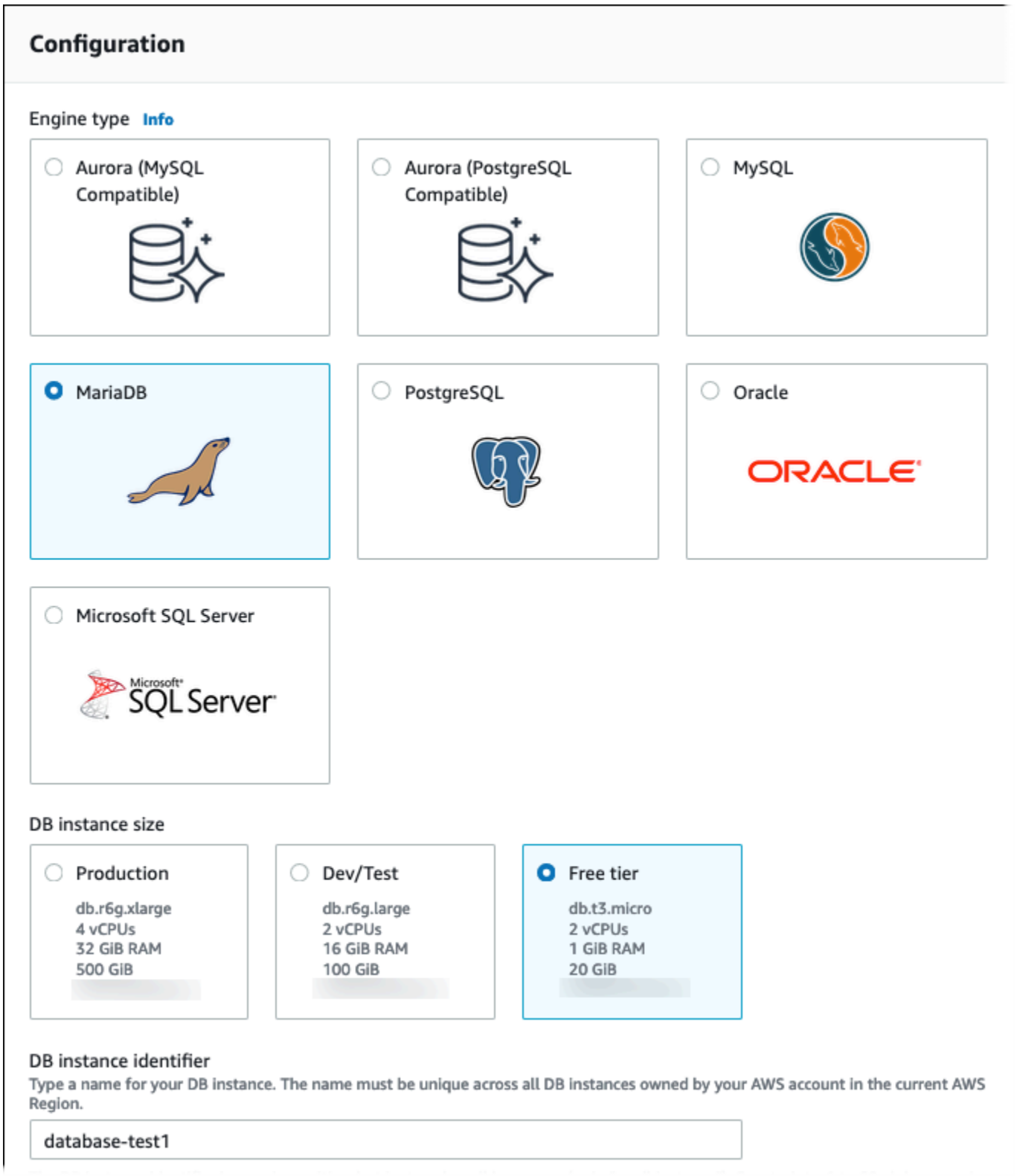
Per creare un'istanza database MariaDB con Easy create (Creazione rapida)

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nell'angolo in alto a destra della console Amazon RDS, scegli l'istanza database Regione AWS in cui desideri creare l'istanza DB.
3. Nel riquadro di navigazione, scegliere Databases (Database).
4. Scegliere Create database (Crea database) e verificare che l'opzione Easy Create (Creazione rapida) sia selezionata.



5. In Configuration (Configurazione), seleziona MariaDB.
6. Per DB instance size (Dimensione istanza database), seleziona Free tier (Piano gratuito).
7. Per l'identificatore dell'istanza DB, inserisci **`database-test1`**.
8. Per Nome utente master, inserisci un nome per l'utente master o lascia il nome predefinito.

   La pagina Create database (Crea database) la pagina dovrebbe apparire simile alla seguente immagine.

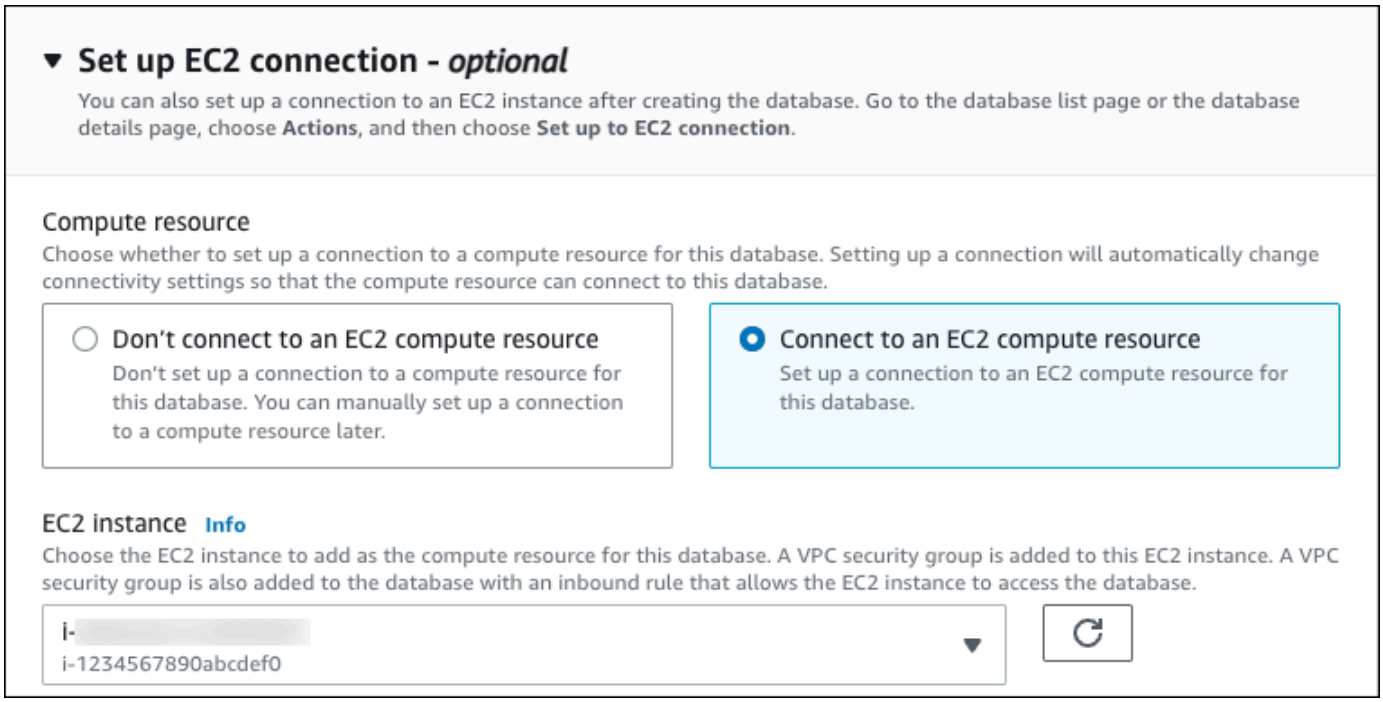


9. Per utilizzare una password master generata automaticamente per l'istanza database, seleziona Genera automaticamente una password.

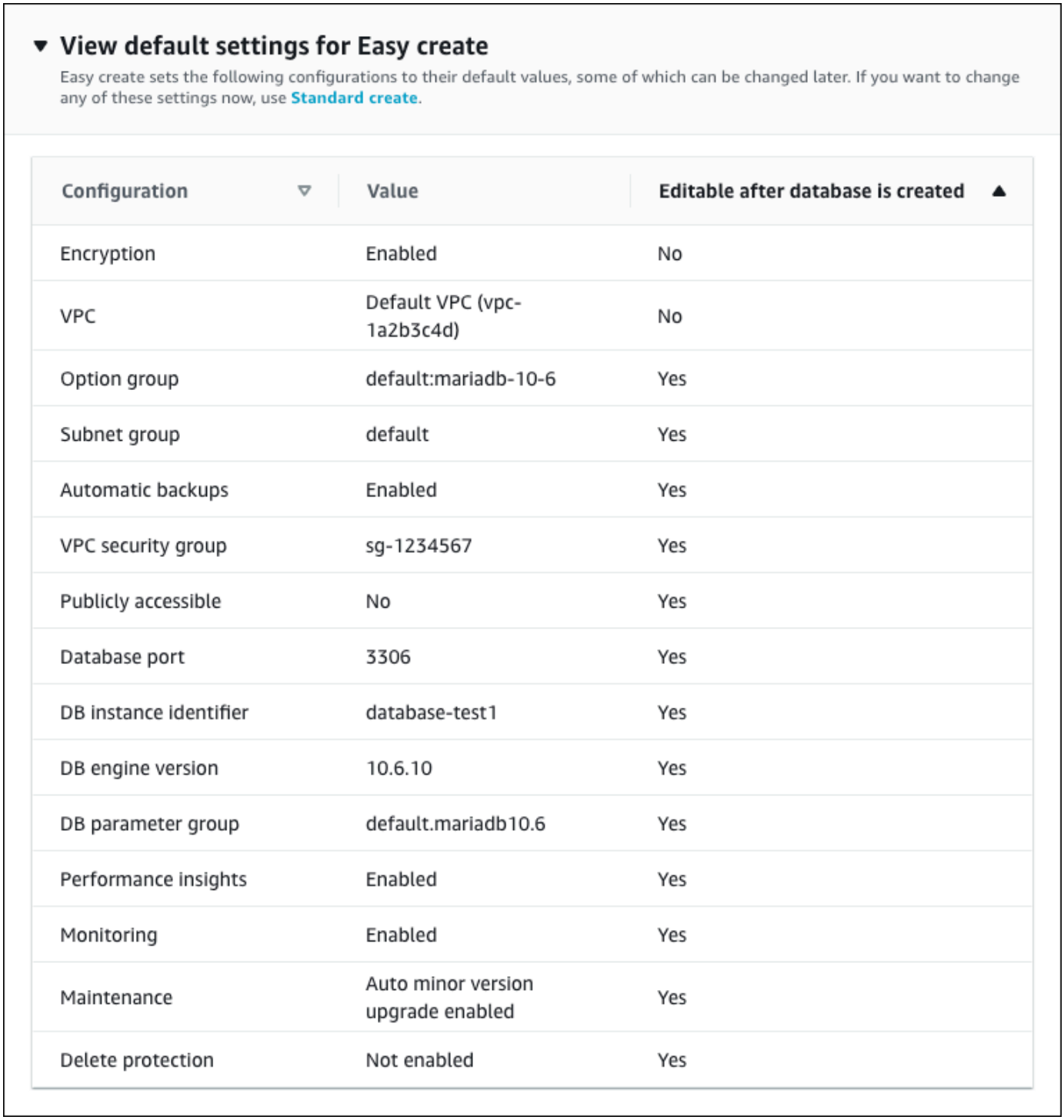
Per inserire la password master, deseleziona la casella Genera automaticamente una password e inserisci la stessa password in Password master e Conferma password.

10. Per configurare una connessione con l'istanza EC2 creata in precedenza, apri Configura connessione EC2 - opzionale.

    Seleziona Connetti a una risorsa di calcolo EC2. Scegli l'istanza EC2 creata in precedenza.



11. Apri Visualizza le impostazioni predefinite per la creazione Semplice.



View default settings for Easy create

Easy create sets the following configurations to their default values, some of which can be changed later. If you want to change any of these settings now, use **Standard create**.

Configuration	Value	Editable after database is created
Encryption	Enabled	No
VPC	Default VPC (vpc-1a2b3c4d)	No
Option group	default:mariadb-10-6	Yes
Subnet group	default	Yes
Automatic backups	Enabled	Yes
VPC security group	sg-1234567	Yes
Publicly accessible	No	Yes
Database port	3306	Yes
DB instance identifier	database-test1	Yes
DB engine version	10.6.10	Yes
DB parameter group	default.mariadb10.6	Yes
Performance insights	Enabled	Yes
Monitoring	Enabled	Yes
Maintenance	Auto minor version upgrade enabled	Yes
Delete protection	Not enabled	Yes

Puoi esaminare le impostazioni predefinite utilizzate con Easy create (Creazione rapida). La colonna Modificabile dopo la creazione del database mostra le opzioni che puoi modificare dopo aver creato il database.

- Se un'impostazione contiene No in quella colonna e desideri cambiarla, puoi utilizzare la Creazione standard per creare l'istanza database.
- Se un'impostazione contiene Sì in quella colonna e desideri cambiarla, puoi utilizzare la Creazione standard per creare l'istanza database o modificare l'istanza database dopo averla creata per cambiare l'impostazione.

12. Scegliere Crea database.

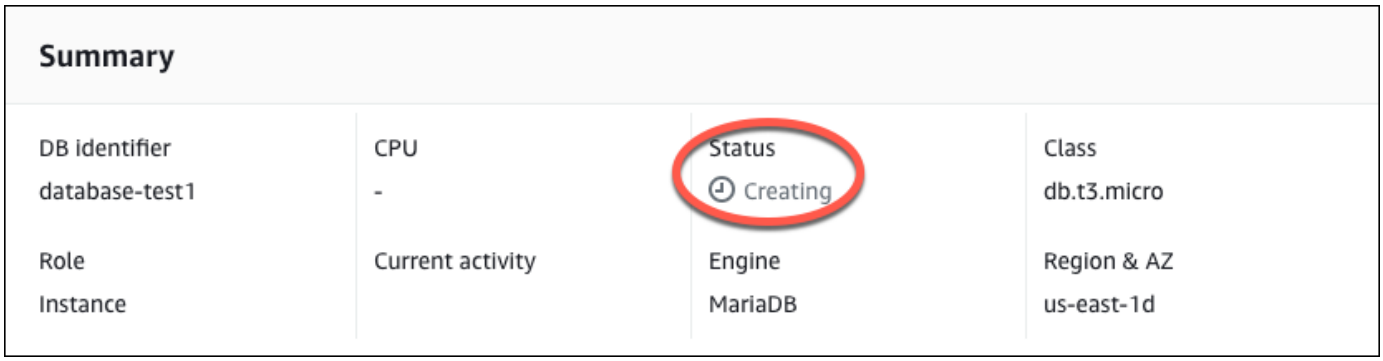
    Per vedere nome utente e password per l'istanza database, seleziona View credential details (Vedi dettagli delle credenziali).

    Per connetterti all'istanza database come utente principale, utilizza il nome utente e la password visualizzati.

    > **Important**
    >
    > Non potrai visualizzare di nuovo la password dell'utente principale. Se non la registri, potresti doverla modificare.
    > Se devi modificare la password dell'utente principale dopo che l'istanza database è disponibile, puoi modificare l'istanza database per eseguire tale operazione. Per ulteriori informazioni sulla modifica di un'istanza database , consulta Modifica di un'istanza database Amazon RDS.

13. Nell'elenco Database seleziona il nome della nuova istanza database MariaDB per visualizzarne i dettagli.

    L'istanza database ha lo stato Creazione in corso fino a quando non è pronta per essere utilizzata.



Quando lo stato cambia in Available (Disponibile), puoi connettersi all'istanza database. A seconda della classe di istanza database e della quantità di storage, prima che la nuova istanza sia disponibile possono trascorrere fino a 20 minuti.

(Facoltativo) Crea VPC, istanza EC2 e istanza MariaDB utilizzando AWS CloudFormation

Invece di usare la console per creare il tuo VPC, l'istanza EC2 e l'istanza MariaDB, puoi usarla AWS CloudFormation per fornire AWS risorse trattando l'infrastruttura come codice. Per aiutarti a organizzare AWS le tue risorse in unità più piccole e più gestibili, puoi utilizzare la funzionalità nested stack. AWS CloudFormation Per ulteriori informazioni, consulta Creare uno stack sulla AWS CloudFormation console e Lavorare con gli stack annidati.

> **Important**
>
> AWS CloudFormation è gratuito, ma le risorse che CloudFormation crea sono attive. Ti verranno addebitati i costi di utilizzo standard per queste risorse fino alla loro cessazione. L'addebito totale sarà minimo. Per informazioni su come ridurre al minimo gli addebiti, vai al AWS piano gratuito.

Per creare le tue risorse utilizzando la AWS CloudFormation console, completa i seguenti passaggi:

- Passaggio 1: scarica il CloudFormation modello
- Passaggio 2: configura le tue risorse utilizzando CloudFormation

Scarica il CloudFormation modello

Un CloudFormation modello è un file di testo JSON o YAML che contiene le informazioni di configurazione sulle risorse che desideri creare nello stack. Questo modello crea anche un VPC e un bastion host per te insieme all'istanza RDS.

Per scaricare il file modello, apri il seguente link, MariaDB CloudFormation template.

Nella pagina Github, fai clic sul pulsante Scarica file raw per salvare il file YAML del modello.

Configura le tue risorse usando CloudFormation

> **Note**
>
> Prima di iniziare questo processo, assicurati di avere una coppia di chiavi per un'istanza EC2 nel tuo Account AWS. Per ulteriori informazioni, consulta Coppie di chiavi Amazon EC2 e istanze Linux.

Quando utilizzi il AWS CloudFormation modello, devi selezionare i parametri corretti per assicurarti che le risorse vengano create correttamente. Segui la procedura riportata di seguito:

1. Accedi AWS Management Console e apri la AWS CloudFormation console all'indirizzo https://console.aws.amazon.com/cloudformation.
2. Scegli Crea stack.
3. Nella sezione Specificare il modello, seleziona Carica un file modello dal computer, quindi scegli Avanti.
4. Nella pagina Specificare i dettagli dello stack, imposta i seguenti parametri:
   a. Imposta il nome dello stack su TestStackMariadB.
   b. In Parametri, imposta le zone di disponibilità selezionando tre zone di disponibilità.
   c. Nella configurazione Linux Bastion Host, in Key Name, seleziona una coppia di chiavi per accedere alla tua istanza EC2.
   d. Nelle impostazioni di configurazione di Linux Bastion Host, imposta l'intervallo IP consentito sul tuo indirizzo IP. Per connetterti alle istanze EC2 nel tuo VPC utilizzando Secure Shell (SSH), determina il tuo indirizzo IP pubblico utilizzando il servizio all'indirizzo https://checkip.amazonaws.com. Un esempio di indirizzo IP è 192.0.2.1/32.

      > **Warning**
      >
      > Se utilizzi `0.0.0.0/0` per l'accesso SSH, consenti a tutti gli indirizzi IP di accedere alle istanze EC2 pubbliche utilizzando SSH. Questo approccio è accettabile per un breve periodo di tempo in un ambiente di test, ma non è sicuro per gli ambienti di produzione. In produzione, autorizza solo un determinato indirizzo IP o un intervallo di indirizzi per accedere alle istanze EC2 utilizzando SSH.

e. Nella configurazione generale del database, imposta la classe dell'istanza del database su db.t3.micro.

f. Imposta il nome del database su. **`database-test1`**

g. Per Nome utente principale del database, inserisci un nome per l'utente principale.

h. Imposta la password utente principale di Manage DB con Secrets Manager su `false` per questo tutorial.

i. Per la password del database, imposta una password a tua scelta. Ricorda questa password per ulteriori passaggi del tutorial.

j. In Configurazione dell'archiviazione del database, imposta il tipo di archiviazione del database su gp2.

k. Nella configurazione Database Monitoring, imposta Enable RDS Performance Insights su false.

l. Lascia tutte le altre impostazioni come valori predefiniti. Fate clic su Avanti per continuare.

5. Nella pagina Review stack, seleziona Invia dopo aver verificato le opzioni del database e dell'host Linux bastion.

Una volta completato il processo di creazione dello stack, visualizza gli stack con nomi BastionStacke RDSNS per annotare le informazioni necessarie per connetterti al database. Per ulteriori informazioni, vedere Visualizzazione dei dati e delle risorse AWS CloudFormation dello stack su. AWS Management Console

Fase 3: connessione a un'istanza database MariaDB

È possibile utilizzare qualsiasi applicazione client SQL standard per la connessione all'istanza database. In questo esempio, ti connetti a un'istanza database MariaDB utilizzando il client della linea di comando mysql.

Per eseguire la connessione a un'istanza database MariaDB

1. Individuare l'endpoint (nome DNS) e il numero di porta per l'istanza database.

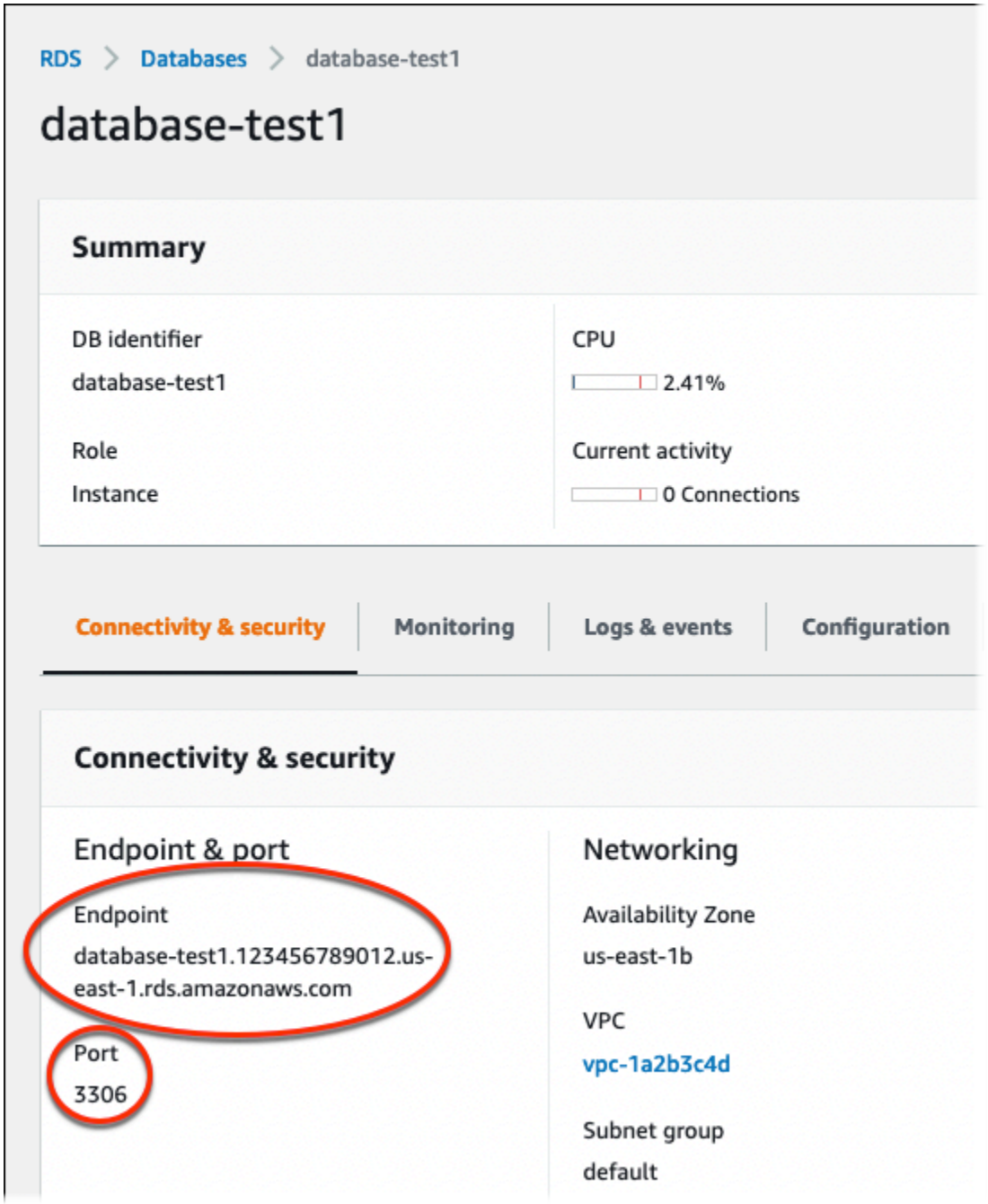
   a. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.

   b. Nell'angolo in alto a destra della console Amazon RDS, scegli l' Regione AWS istanza DB.

   c. Nel riquadro di navigazione, scegli Databases (Database).

   d. Scegliere il nome dell'istanza database MariaDB per visualizzarne i dettagli.

e. Nella scheda Connectivity & security (Connettività e sicurezza), copiare l'endpoint. Annotare anche il numero di porta. L'endpoint e il numero di porta sono necessari per la connessione all'istanza database.



2. Esegui la connessione all'istanza EC2 creata in precedenza seguendo la procedura riportata in Connessione all'istanza di Linux nella Guida per l'utente per istanze Linux di Amazon EC2.

   Ti consigliamo di connetterti all'istanza EC2 tramite SSH. Se l'utilità client SSH è installata su Windows, Linux o Mac, puoi connetterti all'istanza utilizzando il comando nel seguente formato:

```
ssh -i location_of_pem_file ec2-user@ec2-instance-public-dns-name
```

Ad esempio, supponi che `ec2-database-connect-key-pair.pem` sia archiviato in `/dir1` su Linux e che il DNS IPv4 pubblico per l'istanza EC2 sia `ec2-12-345-678-90.compute-1.amazonaws.com`. Il comando SSH sarà simile al seguente:

```
ssh -i /dir1/ec2-database-connect-key-pair.pem ec2-
user@ec2-12-345-678-90.compute-1.amazonaws.com
```

3. Ottieni le ultime correzioni di bug e gli aggiornamenti di sicurezza aggiornando il software sulla tua istanza EC2. A questo scopo, eseguire il comando seguente.

> **Note**
>
> L'opzione `-y` installa gli aggiornamenti senza chiedere conferma. Per esaminare gli aggiornamenti prima di installarli, omettere questa opzione.

```
sudo dnf update -y
```

4. Installa il client della linea di comando mysql da MariaDB.

   Per installare il client della linea di comando MariaDB su Amazon Linux 2023, esegui il comando seguente:

```
sudo dnf install mariadb105
```

5. Esegui la connessione all'istanza database MariaDB. Ad esempio, specifica il comando seguente: Questa azione consente di connetterti all'istanza database MariaDB utilizzando il client MySQL.

   Sostituisci l'endpoint dell'istanza database (nome DNS) per *`endpoint`* e il nome utente master utilizzato per *`admin`*. Devi fornire la password master utilizzata quando viene richiesta una password.

```
mysql -h endpoint -P 3306 -u admin -p
```

Dopo aver immesso la password per l'utente, l'output dovrebbe essere analogo a quanto mostrato di seguito.

```
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 156
Server version: 10.6.10-MariaDB-log managed by https://aws.amazon.com/rds/

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]>
```

Per ulteriori informazioni sulla connessione a un'istanza database MariaDB, consulta Connessione a un'istanza database che esegue il motore di database MariaDB. In caso di mancata connessione all'istanza database, consulta Impossibile connettersi all'istanza database di Amazon RDS.

Per motivi di sicurezza, la best practice è utilizzare connessioni crittografate. Usa una connessione MariaDB non crittografata solo quando il client e il server sono nello stesso VPC e la rete è attendibile. Per ulteriori informazioni sull'uso di connessioni crittografate, consulta Connessione dal client a riga di comando MySQL con SSL/TLS (crittografato).

6. Eseguire comandi SQL.

   Ad esempio, il seguente comando SQL mostra la data e l'ora correnti:

   ```
   SELECT CURRENT_TIMESTAMP;
   ```

Fase 4: eliminazione dell'istanza EC2 e dell'istanza database

Dopo la connessione e l'esplorazione dell'istanza EC2 e dell'istanza database di esempio che hai creato, eliminale per evitare di ricevere l'addebito dei relativi costi.

Se in passato AWS CloudFormation creavi risorse, salta questo passaggio e vai al passaggio successivo.

Per eliminare l'istanza EC2

1. Accedi AWS Management Console e apri la console Amazon EC2 all'indirizzo https://console.aws.amazon.com/ec2/.
2. Nel riquadro di navigazione, seleziona Istanze.
3. Seleziona l'istanza EC2 e scegli Stato istanza, Termina istanza.
4. Quando viene richiesta la conferma, seleziona Terminate (Interrompi).

Per ulteriori informazioni sull'eliminazione di un'istanza EC2, consulta Interruzione di un'istanza nella Guida per l'utente per istanze Linux di Amazon EC2.

Per eliminare l'istanza database senza snapshot database finale

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, scegliere Databases (Database).
3. Scegliere l'istanza database da eliminare.
4. In Actions (Azioni), selezionare Delete (Elimina).
5. Deseleziona Creare uno snapshot finale? e Conserva backup automatizzati.
6. Completa la conferma e scegli Elimina.

(Facoltativo) Elimina l'istanza EC2 e l'istanza DB create con CloudFormation

Se prima creavi AWS CloudFormation risorse, elimina lo CloudFormation stack dopo esserti connesso ed esplorato l'istanza EC2 e l'istanza DB di esempio, in modo che non ti vengano più addebitati costi.

Per eliminare le risorse CloudFormation

1. Apri la AWS CloudFormation console.
2. Nella pagina Stacks CloudFormationconsole, seleziona lo stack principale (lo stack senza il nome VPCStack o RDSNS). BastionStack
3. Scegli Elimina.
4. Seleziona Elimina stack quando viene richiesta la conferma.

Per ulteriori informazioni sull'eliminazione di uno stack in CloudFormation, consulta Eliminazione di uno stack sulla console nella Guida per l' AWS CloudFormationutente.AWS CloudFormation

(Facoltativo) Connessione dell'istanza database a una funzione Lambda

Puoi anche connettere l'istanza database RDS per MariaDB a una risorsa di calcolo serverless Lambda. Le funzioni Lambda consentono di eseguire il codice senza il provisioning o la gestione dell'infrastruttura. Una funzione Lambda consente inoltre di rispondere automaticamente alle richieste di esecuzione del codice su qualsiasi scala, da una dozzina di eventi al giorno a centinaia al secondo. Per ulteriori informazioni, consulta Connessione automatica di una funzione Lambda e di un'istanza database.

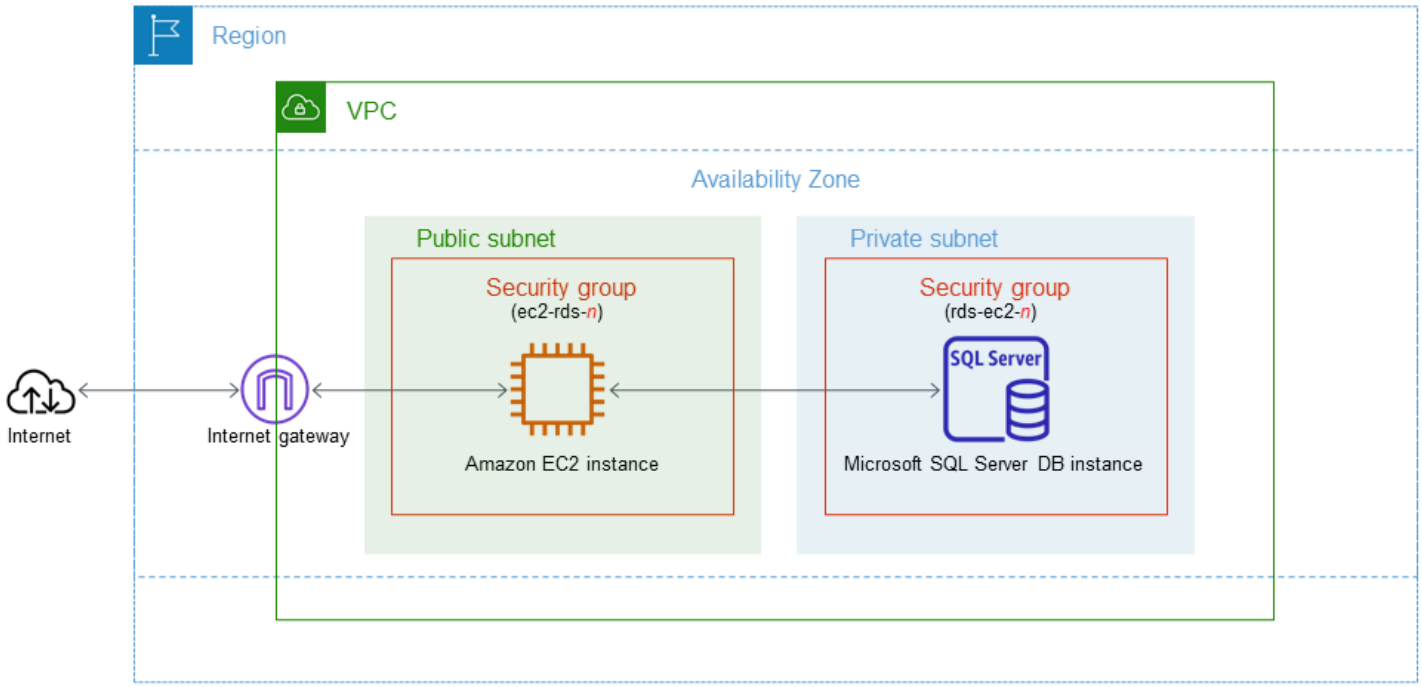
Creazione e connessione a un'istanza database Microsoft SQL Server

Questo tutorial illustra come creare un'istanza EC2 e un'istanza database RDS per Microsoft SQL Server. Il tutorial mostra come accedere all'istanza database dall'istanza EC2 utilizzando il client Microsoft SQL Server Management Studio. Come best practice, questo tutorial spiega come creare un'istanza database privata in un cloud privato virtuale (VPC). Nella maggior parte dei casi, le risorse presenti nello stesso VPC, come le istanze EC2, possono accedere all'istanza database, mentre le risorse esterne al VPC non possono accedervi.

Dopo aver completato il tutorial, è presente una sottorete pubblica e una privata in ogni zona di disponibilità del VPC. In una zona di disponibilità, l'istanza EC2 si trova nella sottorete pubblica mentre l'istanza database si trova nella sottorete privata.

> **Important**
>
> Non è previsto alcun costo per la creazione di un AWS account. Tuttavia, completando questo tutorial, potresti incorrere in costi per le AWS risorse che utilizzi. È possibile eliminare queste risorse dopo aver completato l'esercitazione se non sono più necessarie.

Il seguente diagramma illustra la configurazione al completamento del tutorial.



Questo tutorial ti consente di creare le tue risorse utilizzando uno dei seguenti metodi:

1. Usa AWS Management Console - Fase 2: creazione di un'istanza database SQL Server e Fase 1: creazione di un'istanza EC2
2. Utilizzare AWS CloudFormation per creare l'istanza del database e l'istanza EC2 - (Facoltativo) Crea VPC, istanza EC2 e istanza SQL Server utilizzando AWS CloudFormation

Il primo metodo utilizza Easy create per creare un'istanza DB privata di SQL Server con. AWS Management Console Qui, si specificano solo il tipo di motore DB, la dimensione dell'istanza DB e l'identificatore dell'istanza DB. Easy create (Creazione rapida) utilizza l'impostazione predefinita per altre opzioni di configurazione.

Se invece utilizzi Standard create, puoi specificare più opzioni di configurazione quando crei un'istanza DB. Queste opzioni includono impostazioni per la disponibilità, la sicurezza, i backup e la manutenzione. Per creare un'istanza database pubblica, è necessario utilizzare la Creazione standard. Per informazioni, consulta Creazione di un'istanza database Amazon RDS.

Argomenti

Prerequisiti

Prima di iniziare, completa le fasi descritte in questa sezione:

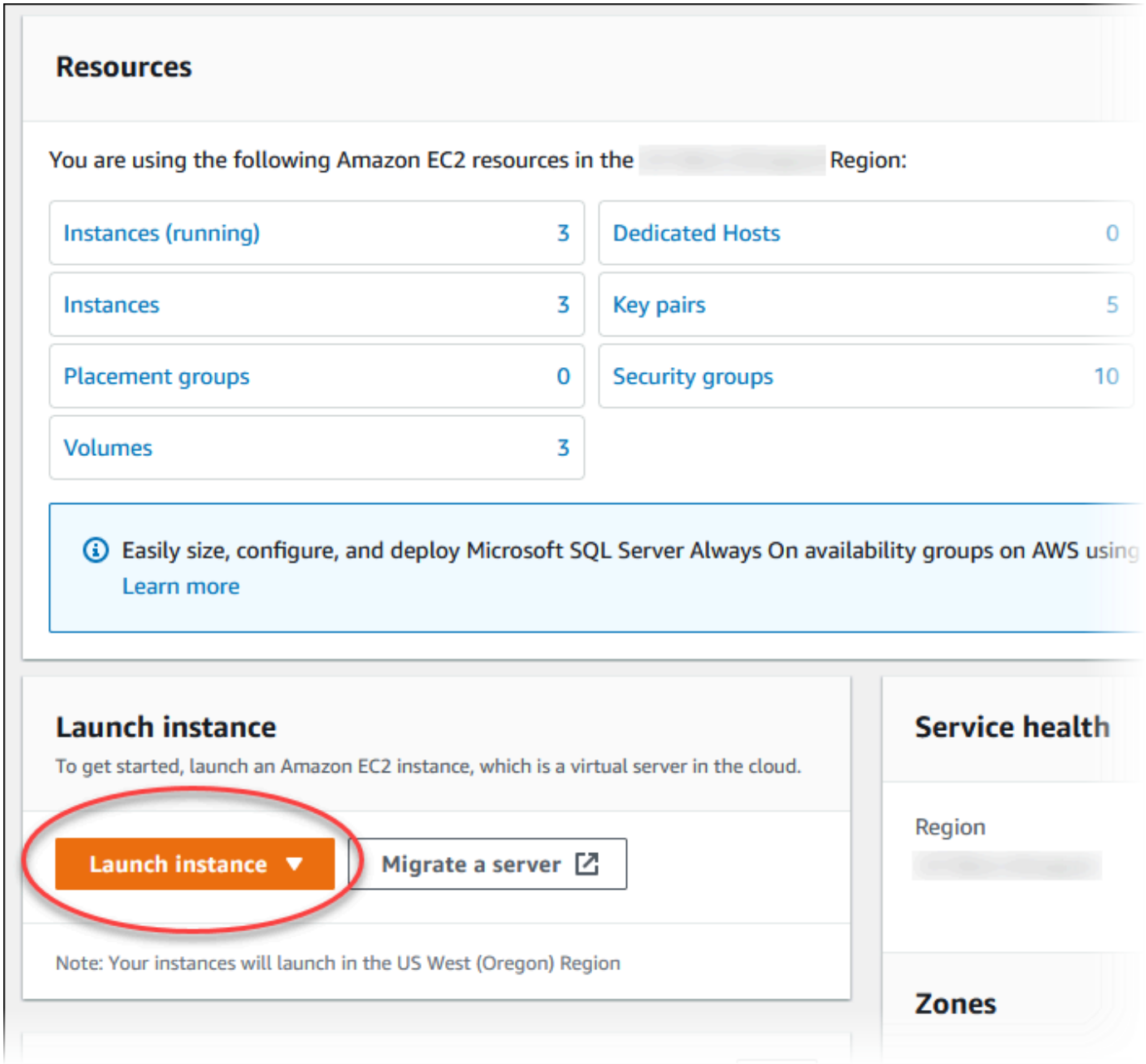
- Registrati per un Account AWS
- Crea un utente con accesso amministrativo

Fase 1: creazione di un'istanza EC2

Crea un'istanza Amazon EC2 da utilizzare per connetterti al database.

Per creare un'istanza EC2

1. Accedi AWS Management Console e apri la console Amazon EC2 all'indirizzo https://console.aws.amazon.com/ec2/.
2. Nell'angolo in alto a destra di AWS Management Console, scegli quello che Regione AWS hai usato in precedenza per il database.
3. Seleziona Pannello di controllo EC2, quindi Avvia istanza, come visualizzato di seguito.

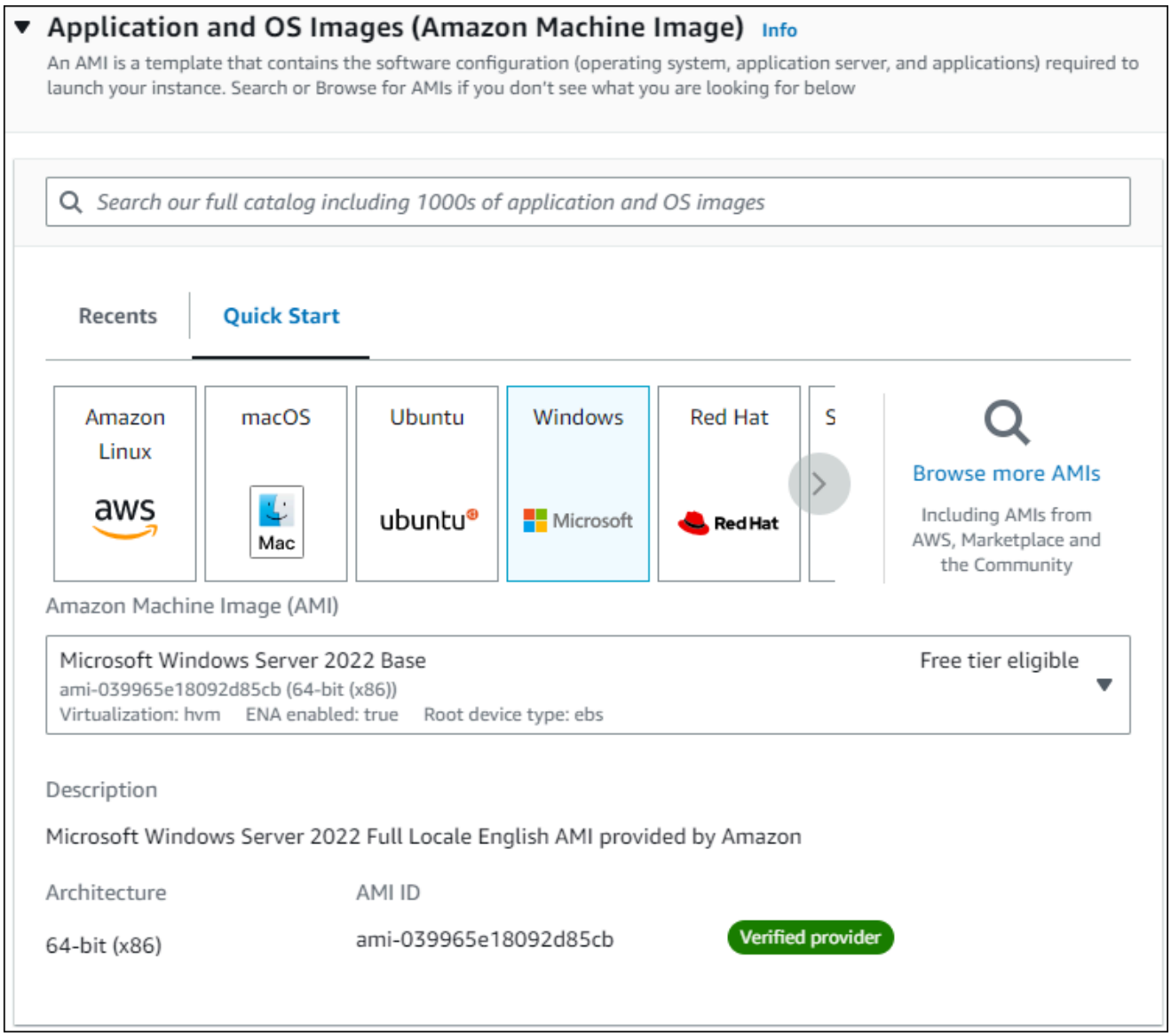


Viene visualizzata la pagina Avvia un'istanza.

4. Scegli le seguenti impostazioni nella pagina Avvia un'istanza.

   a. Nell'area Name and tags (Nome e tag), in Name (Nome) inserisci **`ec2-database-connect`**.

   b. In Immagini di applicazioni e sistema operativo (Amazon Machine Image), scegli Windows, quindi scegli Microsoft Windows Server 2022 Base. Mantieni le selezioni predefinite per le altre opzioni.



c. In Instance type (Tipo di istanza), scegli t2.micro.

d. In Key pair (login) (Coppia di chiavi (login), per Key pair name (Nome della coppia di chiavi), scegli una coppia di chiavi esistente. Per creare una nuova coppia di chiavi per l'istanza Amazon EC2, scegli Create new key pair (Crea nuova coppia di chiavi) e quindi utilizza la finestra Create key pair (Crea coppia di chiavi) per crearla.

   Per ulteriori informazioni sulla creazione di una nuova coppia di chiavi, consulta Creazione di una coppia di chiavi nella Guida per l'utente di Amazon EC2 per istanze Windows.

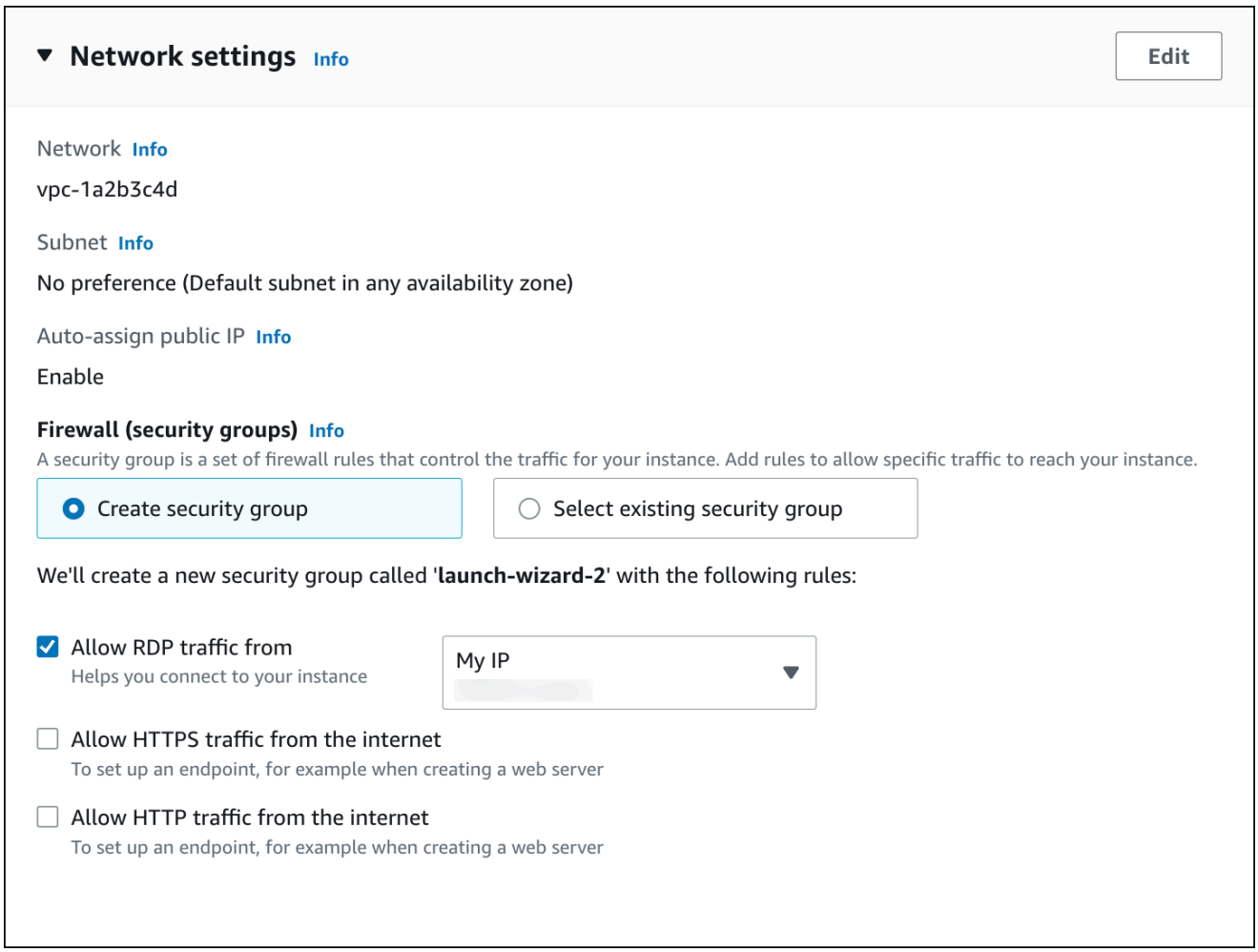
e. Nel campo Firewall (gruppi di sicurezza) nella sezione Impostazioni di rete, scegliere Consenti traffico RDP da per connettersi all'istanza EC2.

È possibile scegliere Il mio IP se l'indirizzo IP visualizzato è corretto per le connessioni RDP. In caso contrario, è possibile determinare l'indirizzo IP da utilizzare per connettersi alle istanze EC2 nel VPC utilizzando RDP. Per determinare l'indirizzo IP pubblico, in una finestra o una scheda del browser diversa, è possibile utilizzare il servizio all'indirizzo https://checkip.amazonaws.com. Un esempio di indirizzo IP è 192.0.2.1/32.

In molti casi, è possibile eseguire la connessione tramite un fornitore di servizi Internet (ISP) o con la protezione di un firewall senza un indirizzo IP statico. In tal caso, accertati di determinare l'intervallo di indirizzi IP utilizzati dai computer client.

> **Warning**
>
> Se utilizzi `0.0.0.0/0` per l'accesso RDP, consenti a tutti gli indirizzi IP di accedere alle istanze EC2 pubbliche utilizzando RDP. Questo approccio è accettabile per un breve periodo di tempo in un ambiente di test, ma non è sicuro per gli ambienti di produzione. In produzione, autorizza solo un determinato indirizzo IP o un intervallo di indirizzi per l'accesso alle istanze EC2 utilizzando RDP.

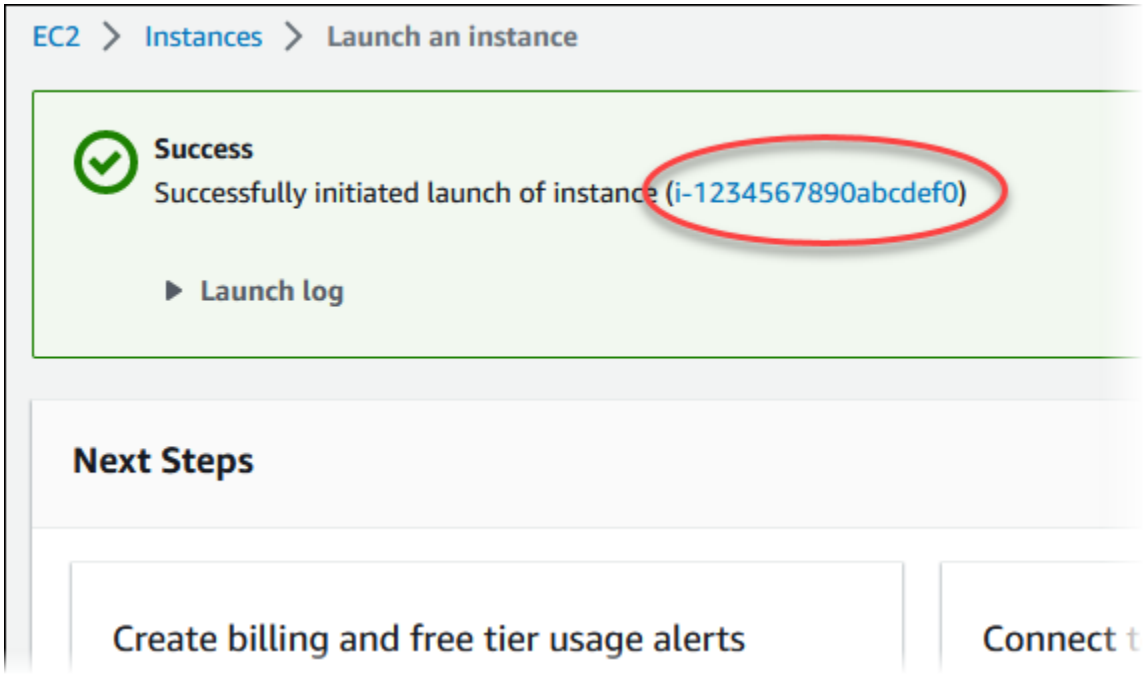
L'immagine seguente mostra un esempio della sezione Impostazioni di rete.



f. Non modificare i valori predefiniti per le sezioni rimanenti.

g. Analizza un riepilogo della configurazione dell'istanza EC2 nel pannello Riepilogo e, quando è tutto pronto, scegli Avvia istanza.

5. Nella pagina Stato avvio prendi nota dell'identificatore per la nuova istanza EC2, ad esempio: `i-1234567890abcdef0`.



6. Scegli l'identificatore dell'istanza EC2 per aprire l'elenco delle istanze EC2.
7. Attendi che Stato dell'istanza diventi In esecuzione per l'istanza EC2 prima di continuare.

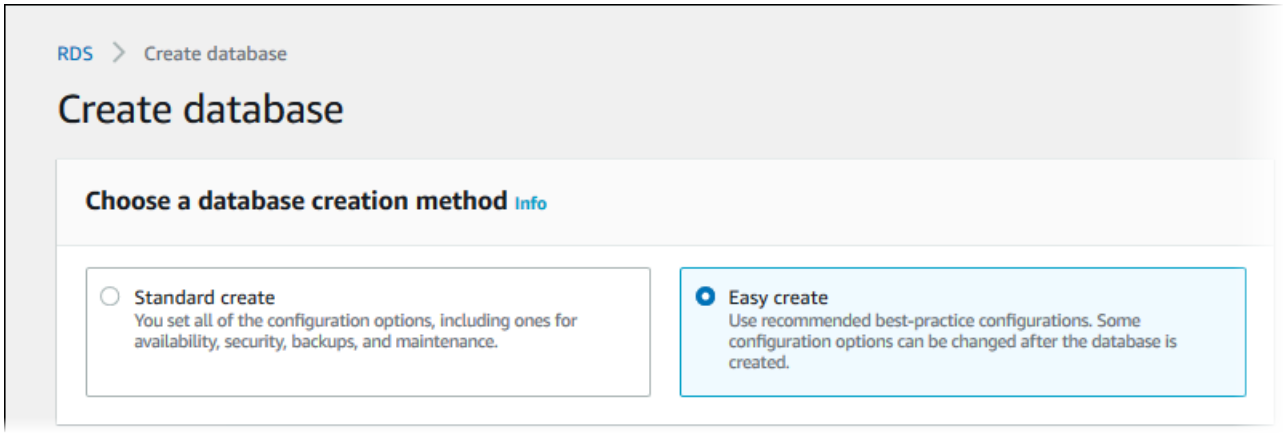
Fase 2: creazione di un'istanza database SQL Server

L'istanza database rappresenta l'elemento di base di Amazon RDS. Questo è l'ambiente dove esegui i tuoi database SQL Server.

In questo esempio, utilizzi Creazione semplice per creare una istanza database che esegue un motore di database SQL Server con una classe di istanza database db.t2.micro.

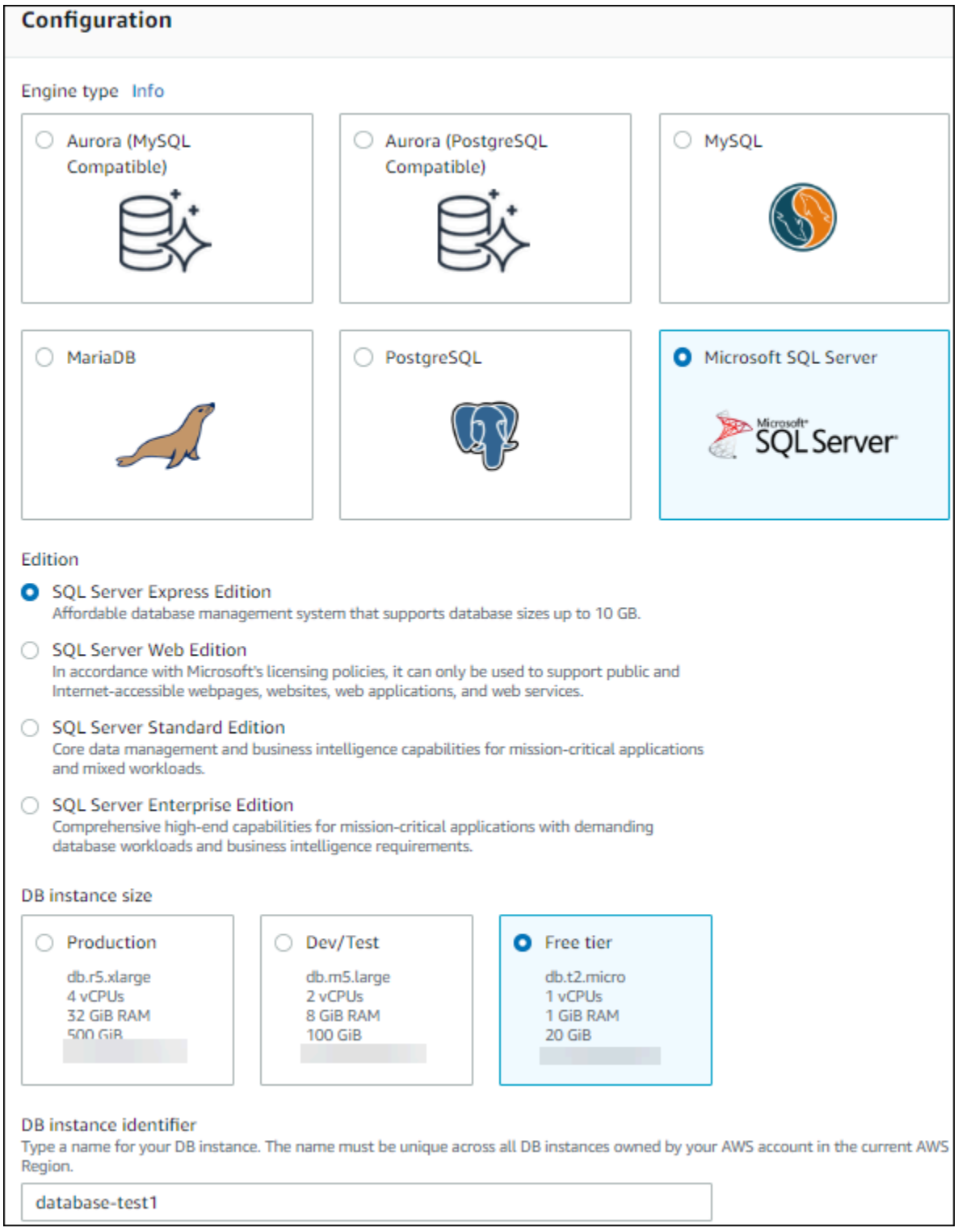
Per creare una istanza database Microsoft SQL Server con l'opzione Creazione rapida

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nell'angolo in alto a destra della console Amazon RDS, scegli l'istanza database Regione AWS in cui desideri creare l'istanza DB.
3. Nel riquadro di navigazione, scegliere Databases (Database).
4. Scegliere Create database (Crea database) e verificare che l'opzione Easy Create (Creazione rapida) sia selezionata.



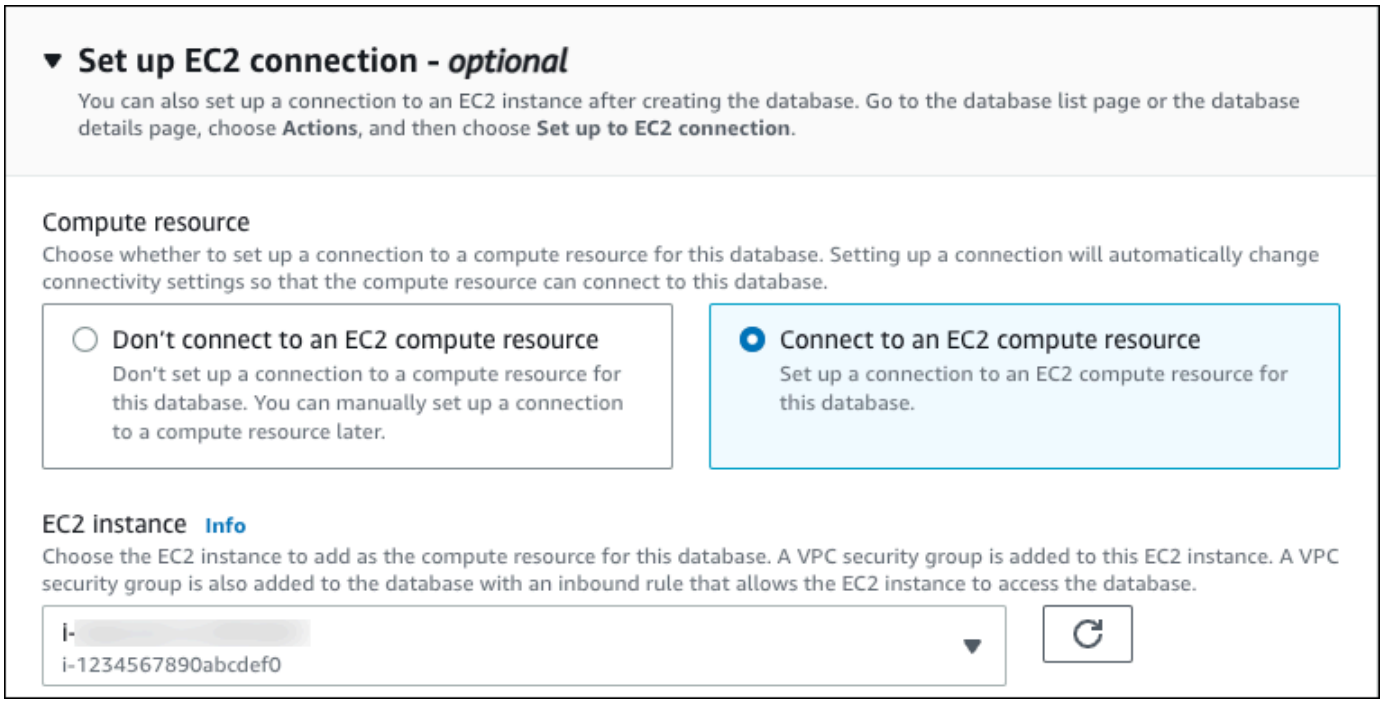
5. In Configuration (Configurazione), selezionare Microsoft SQL Server.
6. In Edizione, scegli SQL Server Express Edition.
7. Per DB instance size (Dimensione istanza database), seleziona Free tier (Piano gratuito).
8. Per l'identificatore dell'istanza DB, inserisci **`database-test1`**.

   La pagina Create database (Crea database) la pagina dovrebbe apparire simile alla seguente immagine.



9. Per Nome utente master, inserisci un nome per l'utente master o lascia il nome predefinito.
10. Per configurare una connessione con l'istanza EC2 creata in precedenza, apri Configura connessione EC2 - opzionale.

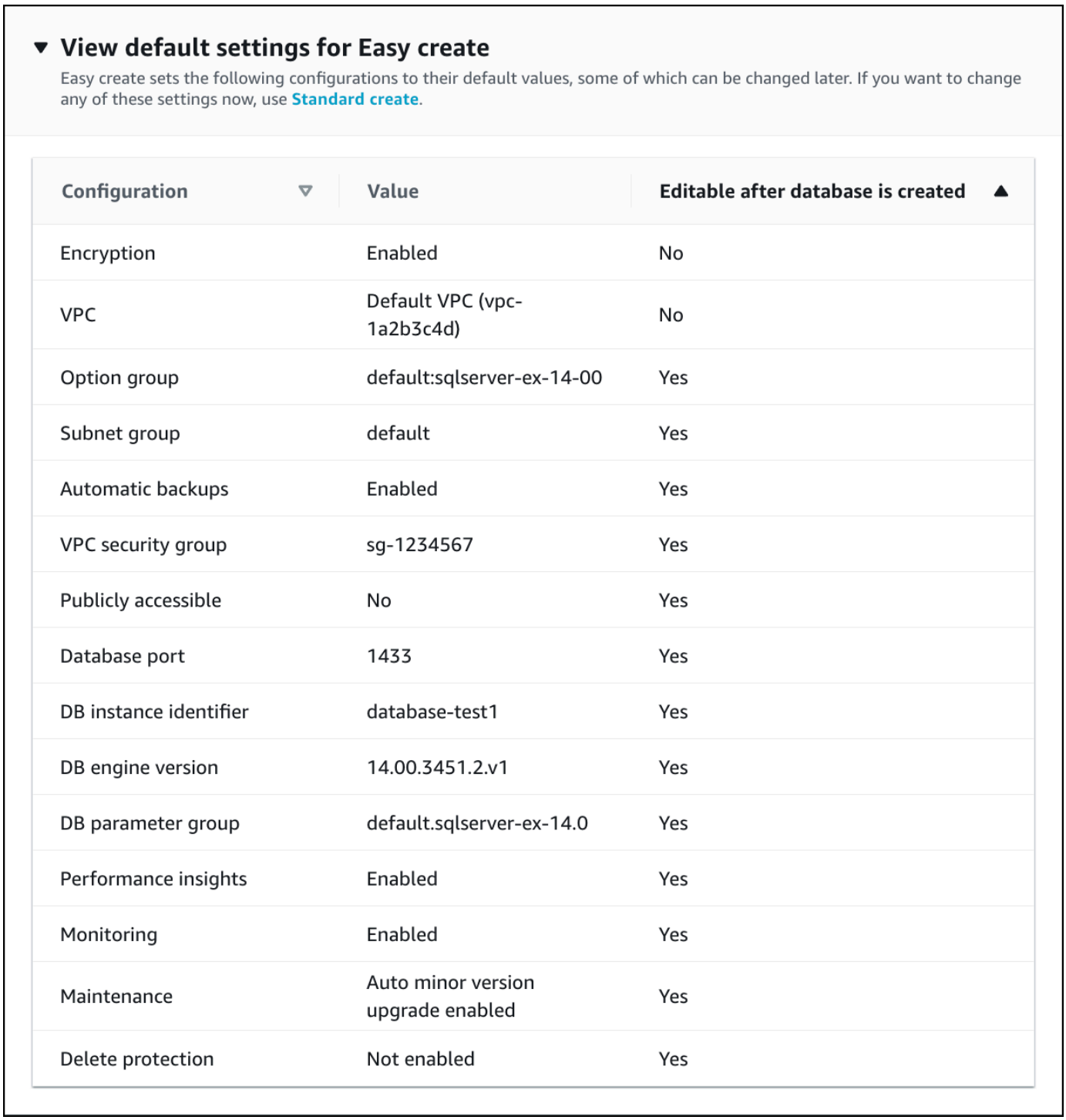
Seleziona Connetti a una risorsa di calcolo EC2. Scegli l'istanza EC2 creata in precedenza.



11. Per utilizzare una password master generata automaticamente per l'istanza database, seleziona la casella Genera automaticamente una password.

    Per inserire la password master, deseleziona la casella Auto generate a password (Genera automaticamente una password) e inserisci la stessa password in Master password (Password master) e Confirm password (Conferma password).

12. Apri Visualizza le impostazioni predefinite per la creazione Semplice.



View default settings for Easy create

Easy create sets the following configurations to their default values, some of which can be changed later. If you want to change any of these settings now, use **Standard create**.

Configuration	Value	Editable after database is created
Encryption	Enabled	No
VPC	Default VPC (vpc-1a2b3c4d)	No
Option group	default:sqlserver-ex-14-00	Yes
Subnet group	default	Yes
Automatic backups	Enabled	Yes
VPC security group	sg-1234567	Yes
Publicly accessible	No	Yes
Database port	1433	Yes
DB instance identifier	database-test1	Yes
DB engine version	14.00.3451.2.v1	Yes
DB parameter group	default.sqlserver-ex-14.0	Yes
Performance insights	Enabled	Yes
Monitoring	Enabled	Yes
Maintenance	Auto minor version upgrade enabled	Yes
Delete protection	Not enabled	Yes

Puoi esaminare le impostazioni predefinite utilizzate con Easy create (Creazione rapida). La colonna Modificabile dopo la creazione del database mostra le opzioni che puoi modificare dopo aver creato il database.

- Se un'impostazione contiene No in quella colonna e desideri cambiarla, puoi utilizzare la Creazione standard per creare l'istanza database.

- Se un'impostazione contiene Sì in quella colonna e desideri cambiarla, puoi utilizzare la Creazione standard per creare l'istanza database o modificare l'istanza database dopo averla creata per cambiare l'impostazione.

13. Scegliere Crea database.

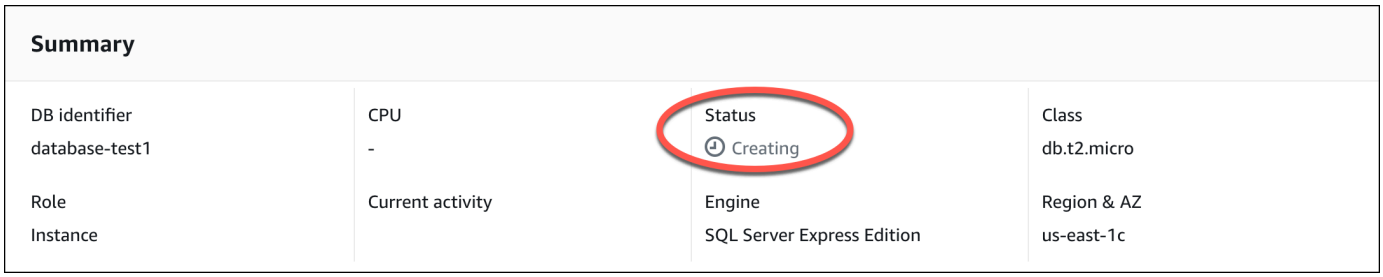
    Per vedere nome utente e password per l'istanza database, seleziona View credential details (Vedi dettagli delle credenziali).

    Per connetterti all'istanza database come utente principale, utilizza il nome utente e la password visualizzati.

    > **Important**
    >
    > Non potrai visualizzare di nuovo la password dell'utente principale. Se non la registri, potresti doverla modificare.
    > Se devi modificare la password dell'utente principale dopo che l'istanza database è disponibile, puoi modificare l'istanza database per eseguire tale operazione. Per ulteriori informazioni sulla modifica di un'istanza database , consulta Modifica di un'istanza database Amazon RDS.

14. Nell'elenco Database seleziona il nome della nuova istanza database SQL Server per visualizzarne i dettagli.

    L'istanza database ha lo stato Creazione in corso fino a quando non è pronta per essere utilizzata.



    Quando lo stato cambia in Available (Disponibile), puoi connettersi all'istanza database. A seconda della classe di istanza database e della quantità di storage, prima che la nuova istanza sia disponibile possono trascorrere fino a 20 minuti.

(Facoltativo) Crea VPC, istanza EC2 e istanza SQL Server utilizzando AWS CloudFormation

Invece di usare la console per creare il tuo VPC, l'istanza EC2 e l'istanza di SQL Server, puoi usarla AWS CloudFormation per fornire AWS risorse trattando l'infrastruttura come codice. Per aiutarti a organizzare AWS le tue risorse in unità più piccole e più gestibili, puoi utilizzare la funzionalità AWS CloudFormation nested stack. Per ulteriori informazioni, consulta Creazione di uno stack sulla AWS CloudFormation console e Utilizzo degli stack annidati.

> **Important**
>
> AWS CloudFormation è gratuito, ma le risorse che CloudFormation crea sono attive. Ti verranno addebitati i costi di utilizzo standard per queste risorse fino alla loro cessazione. L'addebito totale sarà minimo. Per informazioni su come ridurre al minimo gli addebiti, vai al AWS piano gratuito.

Per creare le tue risorse utilizzando la AWS CloudFormation console, completa i seguenti passaggi:

- Passaggio 1: scarica il CloudFormation modello
- Passaggio 2: configura le tue risorse utilizzando CloudFormation

Scarica il CloudFormation modello

Un CloudFormation modello è un file di testo JSON o YAML che contiene le informazioni di configurazione sulle risorse che desideri creare nello stack. Questo modello crea anche un VPC e un bastion host per te insieme all'istanza RDS.

Per scaricare il file modello, apri il seguente link, modello SQL Server. CloudFormation

Nella pagina Github, fai clic sul pulsante Scarica file raw per salvare il file YAML del modello.

Configura le tue risorse usando CloudFormation

> **Note**
>
> Prima di iniziare questo processo, assicurati di avere una coppia di chiavi per un'istanza EC2 nel tuo Account AWS. Per ulteriori informazioni, consulta Coppie di chiavi Amazon EC2 e istanze Linux.

Quando utilizzi il AWS CloudFormation modello, devi selezionare i parametri corretti per assicurarti che le risorse vengano create correttamente. Segui la procedura riportata di seguito:

1. Accedi AWS Management Console e apri la AWS CloudFormation console all'indirizzo https://console.aws.amazon.com/cloudformation.
2. Scegli Crea stack.
3. Nella sezione Specificare il modello, seleziona Carica un file modello dal computer, quindi scegli Avanti.
4. Nella pagina Specificare i dettagli dello stack, imposta i seguenti parametri:
   a. Imposta il nome dello stack su SQL. ServerTestStack
   b. In Parametri, imposta le zone di disponibilità selezionando tre zone di disponibilità.
   c. Nella configurazione Linux Bastion Host, in Key Name, seleziona una coppia di chiavi per accedere alla tua istanza EC2.
   d. Nelle impostazioni di configurazione di Linux Bastion Host, imposta l'intervallo IP consentito sul tuo indirizzo IP. Per connetterti alle istanze EC2 nel tuo VPC utilizzando Secure Shell (SSH), determina il tuo indirizzo IP pubblico utilizzando il servizio all'indirizzo https://checkip.amazonaws.com. Un esempio di indirizzo IP è 192.0.2.1/32.

      > **Warning**
      >
      > Se utilizzi `0.0.0.0/0` per l'accesso SSH, consenti a tutti gli indirizzi IP di accedere alle istanze EC2 pubbliche utilizzando SSH. Questo approccio è accettabile per un breve periodo di tempo in un ambiente di test, ma non è sicuro per gli ambienti di produzione. In produzione, autorizza solo un determinato indirizzo IP o un intervallo di indirizzi per accedere alle istanze EC2 utilizzando SSH.

e. Nella configurazione generale del database, imposta la classe dell'istanza del database su db.t3.micro.

f. Imposta il nome del database su. **`database-test1`**

g. Per Nome utente principale del database, inserisci un nome per l'utente principale.

h. Imposta la password utente principale di Manage DB con Secrets Manager su `false` per questo tutorial.

i. Per la password del database, imposta una password a tua scelta. Ricorda questa password per ulteriori passaggi del tutorial.

j. In Configurazione dell'archiviazione del database, imposta il tipo di archiviazione del database su gp2.

k. Nella configurazione Database Monitoring, imposta Enable RDS Performance Insights su false.

l. Lascia tutte le altre impostazioni come valori predefiniti. Fate clic su Avanti per continuare.

5. Nella pagina Configura le opzioni dello stack, lascia tutte le opzioni predefinite. Fai clic su Avanti per continuare.
6. Nella pagina Review stack, seleziona Invia dopo aver verificato le opzioni del database e dell'host Linux bastion.

Una volta completato il processo di creazione dello stack, visualizza gli stack con nomi BastionStacke RDSNS per annotare le informazioni necessarie per connetterti al database. Per ulteriori informazioni, vedere Visualizzazione dei dati e delle risorse AWS CloudFormation dello stack su. AWS Management Console

Fase 3: connessione all'istanza database SQL Server

Nella procedura seguente, eseguirai la connessione all'istanza database utilizzando Microsoft SQL Server Management Studio (SSMS).

Per connettersi all'istanza database RDS per SQL Server utilizzando SSMS

1. Individuare l'endpoint (nome DNS) e il numero di porta per l'istanza database.

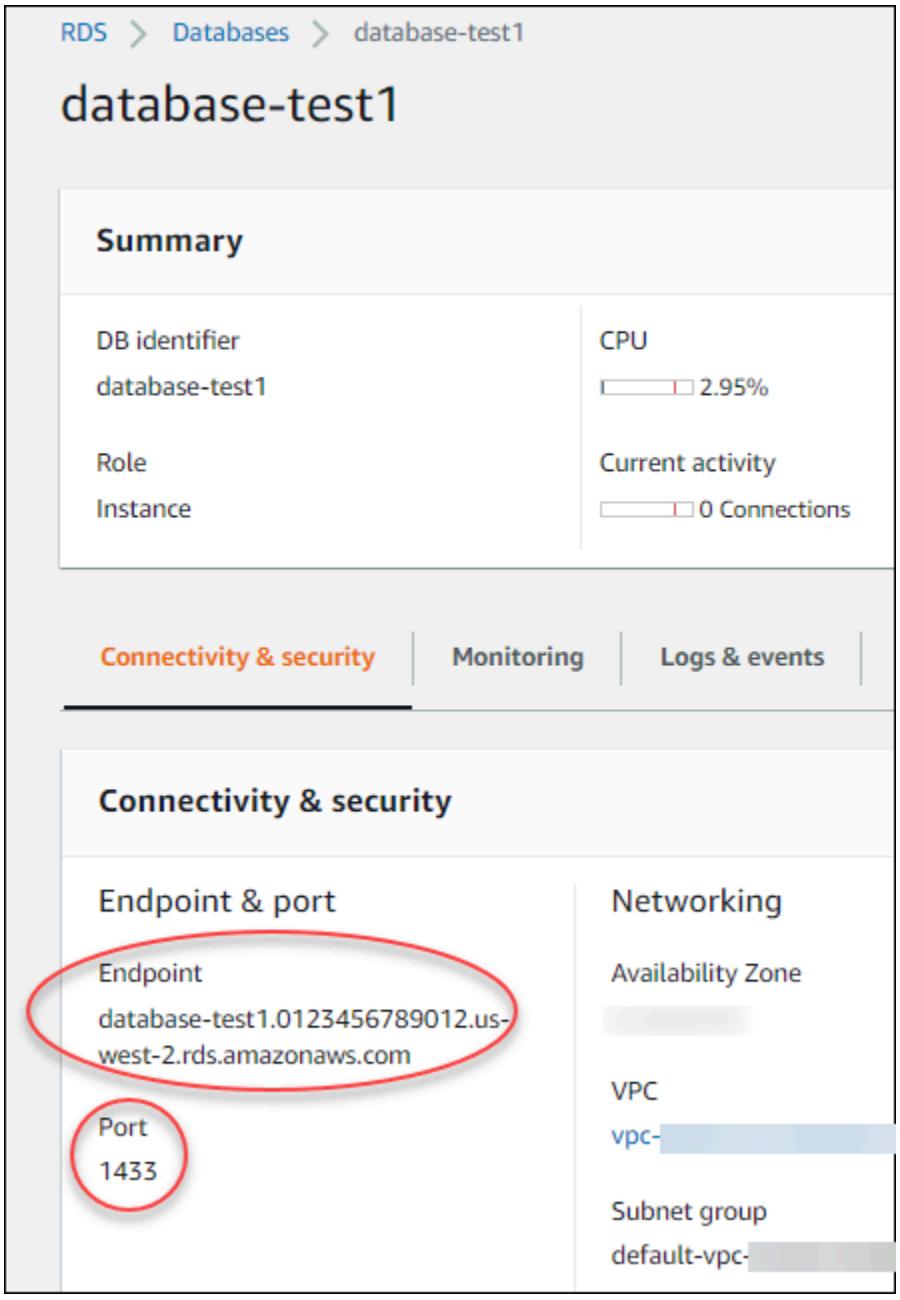
   a. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.

   b. Nell'angolo in alto a destra della console Amazon RDS, scegli l' Regione AWS istanza DB.

   c. Nel riquadro di navigazione, scegli Databases (Database).

d. Scegliere il nome dell'istanza database SQL Server per visualizzarne i dettagli.

e. Nella scheda Connectivity (Connettività), copiare l'endpoint. Annotare anche il numero di porta. L'endpoint e il numero di porta sono necessari per la connessione all'istanza database.



2. Esegui la connessione all'istanza EC2 creata in precedenza seguendo la procedura riportata in Connessione all'istanza Microsoft Windows nella Guida per l'utente per istanze Windows di Amazon EC2.

3. Installa il client SQL Server Management Studio (SSMS) di Microsoft.

Per scaricare una versione autonoma di SSMS nell'istanza EC2, consulta l'argomento relativo al download di SQL Server Management Studio (SSMS) nella documentazione di Microsoft.

a. Usa il menu Start per aprire Internet Explorer.

b. Usa Internet Explorer per scaricare e installare una versione autonoma di SSMS. Se viene visualizzato un messaggio indicante che il sito non è attendibile, aggiungi il sito all'elenco dei siti attendibili.

4. Avvia SQL Server Management Studio (SSMS).

   Viene visualizzata la finestra di dialogo Connect to Server (Connettiti al server).

5. Fornire le informazioni seguenti per l'istanza database di esempio:

   a. In Server type (Tipo di server) scegliere Database Engine (Motore di database).

   b. Per Server name (Nome server), inserire il nome DNS, seguito da una virgola e dal numero di porta (la porta predefinita è 1433). Ad esempio, il nome del server dovrebbe essere simile al seguente:

   ```
   database-test1.0123456789012.us-west-2.rds.amazonaws.com,1433
   ```

   c. In Authentication (Autenticazione) selezionare SQL Server Authentication (Autenticazione SQL Server).

   d. In Accesso, inserisci il nome utente scelto per l'istanza database di esempio. Questo è anche noto come nome utente master.

   e. In Password digitare la password scelta in precedenza per l'istanza database di esempio. Questa è anche nota come password utente master.

6. Scegliere Connetti.

   Dopo qualche secondo, SSMS effettua la connessione all'istanza database. Per motivi di sicurezza, la best practice è utilizzare connessioni crittografate. Utilizza una connessione SQL Server non crittografata solo quando il client e il server sono nello stesso VPC e la rete è attendibile. Per ulteriori informazioni sull'uso di connessioni crittografate, consulta Utilizzo di SSL con un'istanza database Microsoft SQL Server.

Per ulteriori informazioni sulla connessione a un'istanza database Microsoft SQL Server, consulta Connessione a un'istanza database che esegua il motore di database di Microsoft SQL Server.

Per informazioni sui problemi di connessione, consulta Impossibile connettersi all'istanza database di Amazon RDS.

Fase 4: esame dell'istanza database SQL Server di esempio

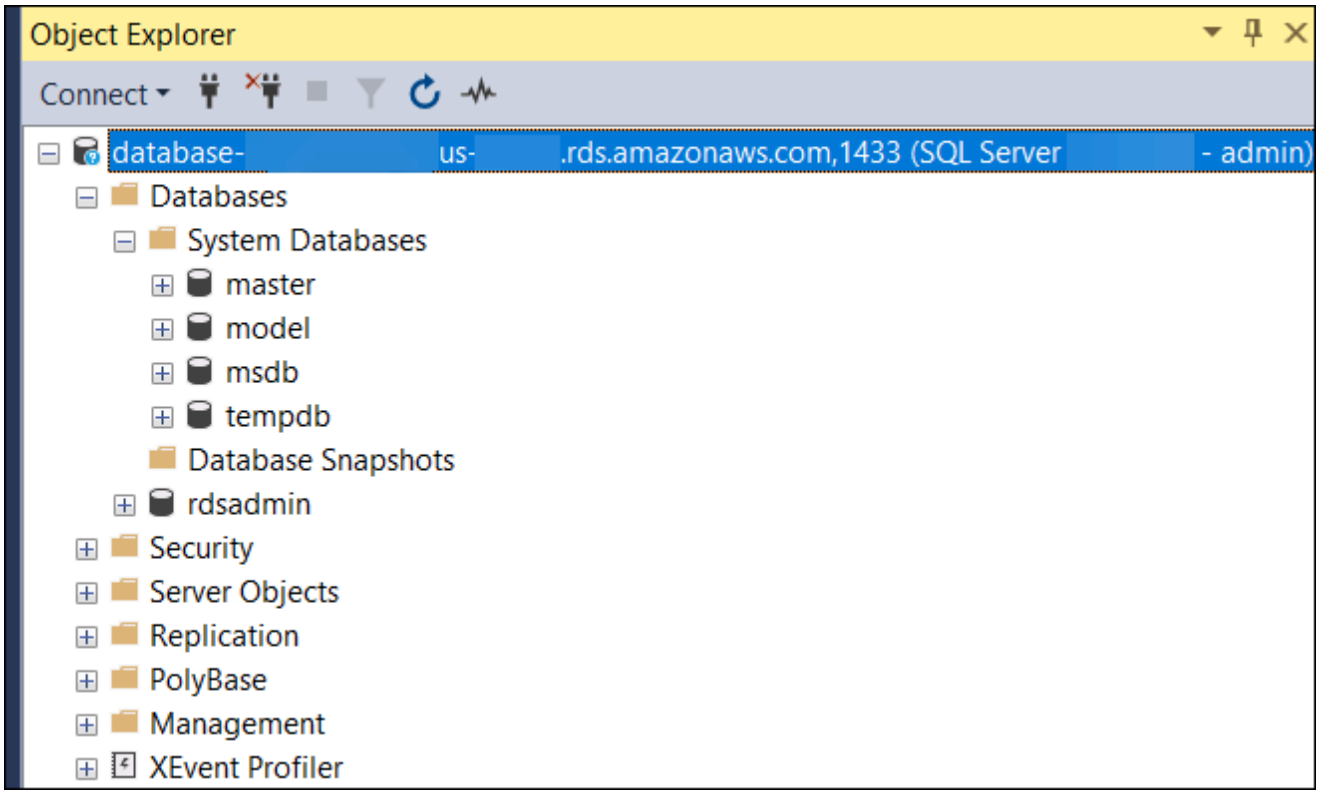
È possibile esplorare l'istanza database di esempio utilizzando Microsoft SQL Server Management Studio (SSMS).

Per esaminare un'istanza database utilizzando SSMS

1. L'istanza database SQL Server integra i database di sistema standard di SQL Server (master, model, msdb e tempdb). Per esaminare i database di sistema, procedere nel modo seguente:

   a. In SSMS, nel menu View (Visualizza), scegliere Object Explorer.

   b. Espandi l'istanza database, seleziona Database quindi Database di sistema come mostrato di seguito.



   L'istanza database di SQL Server viene inoltre fornita con un database denominato `rdsadmin`. Amazon RDS usa questo database per archiviare gli oggetti usati per gestire il database. Il database `rdsadmin` include anche le stored procedure che puoi eseguire per svolgere attività avanzate.

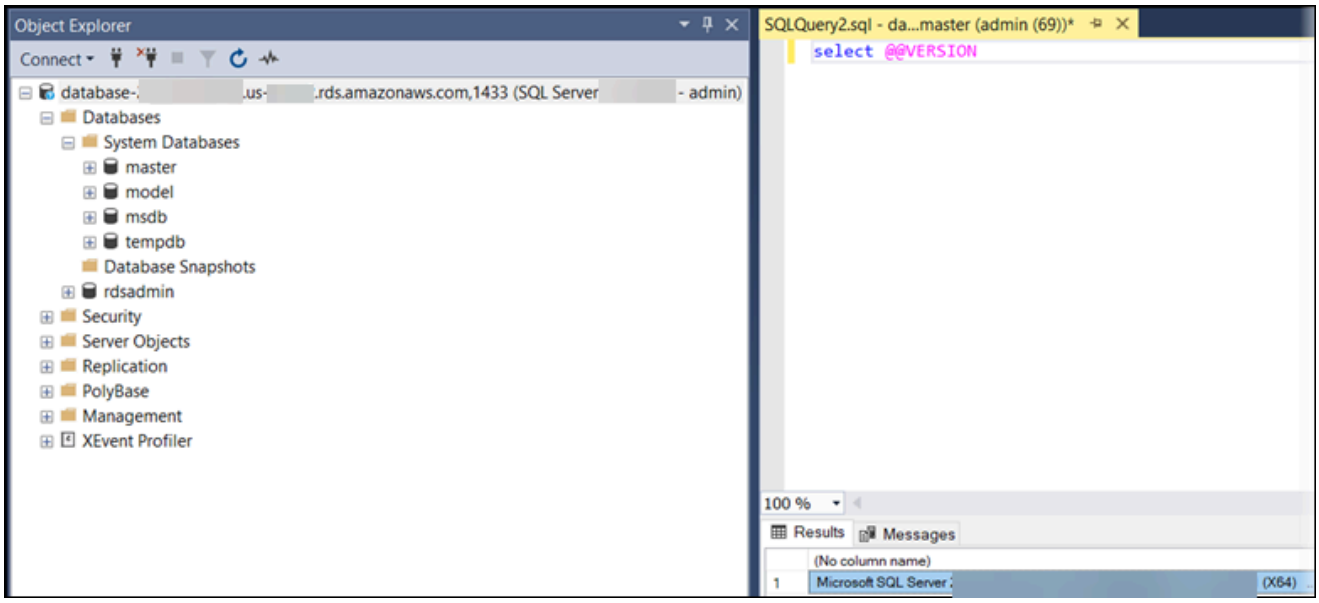
2. Puoi iniziare a creare database personali ed eseguire normalmente query sull'istanza e sui database. Per eseguire una query di test dell'istanza database di esempio, utilizzare la seguente procedura:

   a. In SSMS, nel menu File seleziona Nuovo, quindi scegliere Query con connessione corrente.

   b. Inserisci la query SQL seguente.

   ```
   select @@VERSION
   ```

   c. Eseguire la query. SSMS restituisce la versione di SQL Server dell'istanza database Amazon RDS.



Fase 5: eliminazione dell'istanza EC2 e dell'istanza database

Dopo la connessione e l'esplorazione dell'istanza EC2 e dell'istanza database di esempio che hai creato, eliminale per evitare di ricevere l'addebito dei relativi costi.

Se in passato AWS CloudFormation creavi risorse, salta questo passaggio e vai al passaggio successivo.

Per eliminare l'istanza EC2

1. Accedi AWS Management Console e apri la console Amazon EC2 all'indirizzo https://console.aws.amazon.com/ec2/.
2. Nel riquadro di navigazione, seleziona Istanze.

3. Seleziona l'istanza EC2 e scegli Stato istanza, Termina istanza.
4. Quando viene richiesta la conferma, seleziona Terminate (Interrompi).

Per ulteriori informazioni sull'eliminazione di un'istanza EC2, consulta Interruzione di un'istanza nella Guida per l'utente per istanze Windows.

Per eliminare l'istanza database senza snapshot database finale

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, scegliere Databases (Database).
3. Scegliere l'istanza database da eliminare.
4. In Actions (Azioni), selezionare Delete (Elimina).
5. Deseleziona Creare uno snapshot finale? e Conserva backup automatizzati.
6. Completa la conferma e scegli Elimina.

(Facoltativo) Elimina l'istanza EC2 e l'istanza DB create con CloudFormation

Se prima creavi AWS CloudFormation risorse, elimina lo CloudFormation stack dopo esserti connesso ed esplorato l'istanza EC2 e l'istanza DB di esempio, in modo che non ti vengano più addebitati costi.

Per eliminare le risorse CloudFormation

1. Apri la AWS CloudFormation console.
2. Nella pagina Stacks CloudFormationconsole, seleziona lo stack principale (lo stack senza il nome VPCStack o RDSNS). BastionStack
3. Scegli Elimina.
4. Seleziona Elimina stack quando viene richiesta la conferma.

Per ulteriori informazioni sull'eliminazione di uno stack in CloudFormation, consulta Eliminazione di uno stack sulla console nella Guida per l' AWS CloudFormationutente.AWS CloudFormation

(Facoltativo) Connessione dell'istanza database a una funzione Lambda

Puoi anche connettere la tua istanza database RDS per SQL Server a una risorsa di elaborazione serverless Lambda. Le funzioni Lambda consentono di eseguire il codice senza il provisioning o la gestione dell'infrastruttura. Una funzione Lambda consente inoltre di rispondere automaticamente alle richieste di esecuzione del codice su qualsiasi scala, da una dozzina di eventi al giorno a centinaia al secondo. Per ulteriori informazioni, consulta Connessione automatica di una funzione Lambda e di un'istanza database.

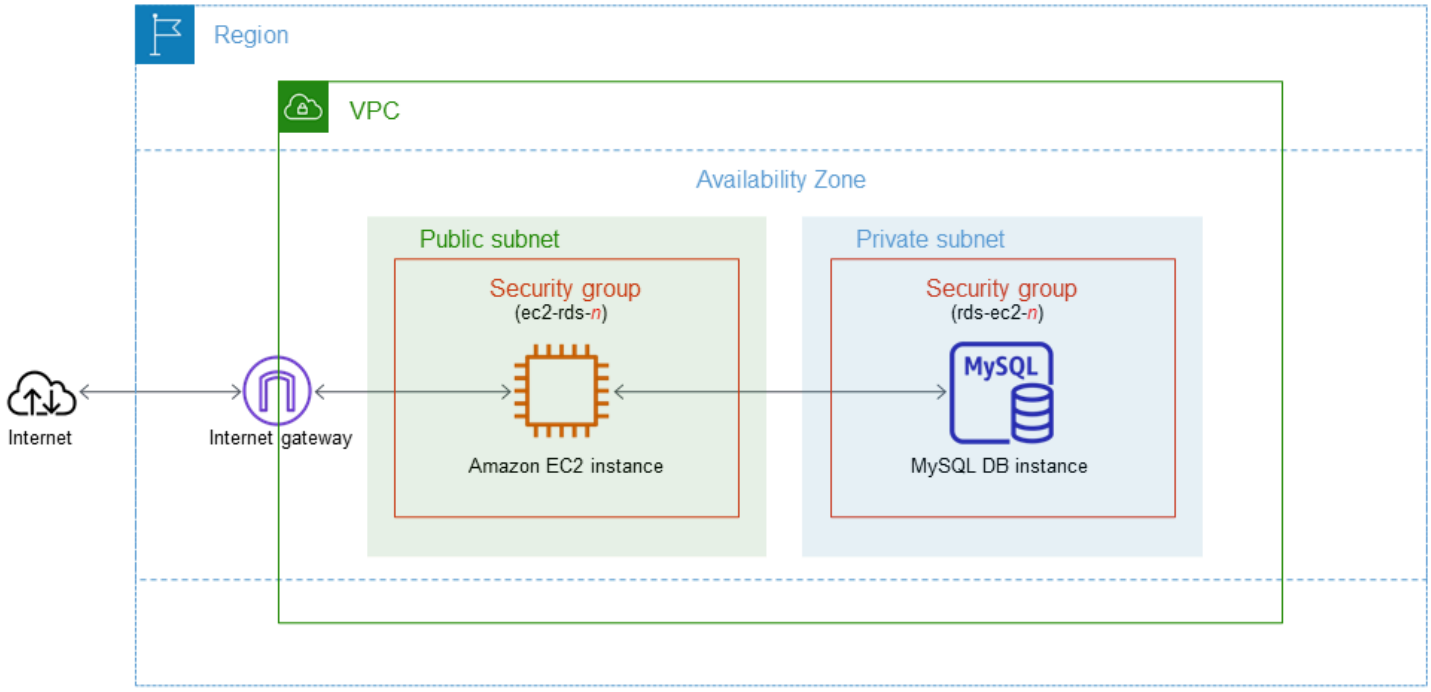
Creazione e connessione di un'istanza database MySQL

Questo tutorial illustra come creare un'istanza EC2 e un'istanza database RDS per MySQL. Il tutorial mostra come accedere all'istanza database dall'istanza EC2 utilizzando il client MySQL standard. Come best practice, questo tutorial spiega come creare un'istanza database privata in un cloud privato virtuale (VPC). Nella maggior parte dei casi, le risorse presenti nello stesso VPC, come le istanze EC2, possono accedere all'istanza database, mentre le risorse esterne al VPC non possono accedervi.

Dopo aver completato il tutorial, è presente una sottorete pubblica e una privata in ogni zona di disponibilità del VPC. In una zona di disponibilità, l'istanza EC2 si trova nella sottorete pubblica mentre l'istanza database si trova nella sottorete privata.

> **Important**
>
> Non è previsto alcun costo per la creazione di un AWS account. Tuttavia, completando questo tutorial, potresti incorrere in costi per le AWS risorse che utilizzi. È possibile eliminare queste risorse dopo aver completato l'esercitazione se non sono più necessarie.

Il seguente diagramma illustra la configurazione al completamento del tutorial.



Questo tutorial ti consente di creare le tue risorse utilizzando uno dei seguenti metodi:

1. Usa AWS Management Console - Fase 2: creazione di un'istanza database MySQL e Fase 1: creazione di un'istanza EC2
2. Utilizzare AWS CloudFormation per creare l'istanza del database e l'istanza EC2 - (Facoltativo) Crea VPC, istanza EC2 e istanza MySQL utilizzando AWS CloudFormation

Il primo metodo utilizza Easy create per creare un'istanza DB MySQL privata con. AWS Management Console Qui, si specificano solo il tipo di motore DB, la dimensione dell'istanza DB e l'identificatore dell'istanza DB. Easy create (Creazione rapida) utilizza l'impostazione predefinita per altre opzioni di configurazione.

Se invece utilizzi Standard create, puoi specificare più opzioni di configurazione quando crei un'istanza DB. Queste opzioni includono impostazioni per la disponibilità, la sicurezza, i backup e la manutenzione. Per creare un'istanza database pubblica, è necessario utilizzare la Creazione standard. Per informazioni, consulta Creazione di un'istanza database Amazon RDS.

Argomenti

Prerequisiti

Prima di iniziare, completa le fasi descritte in questa sezione:

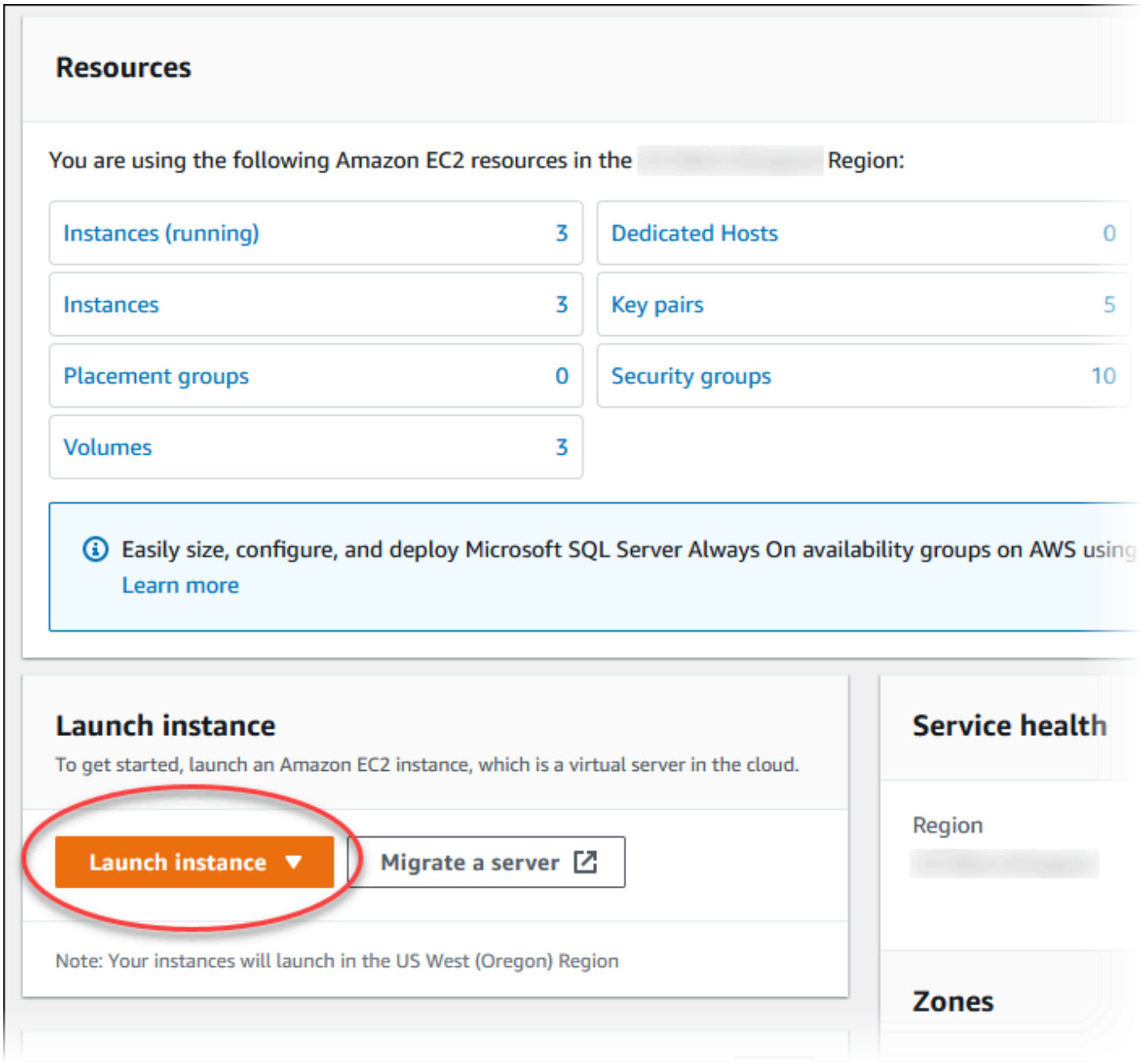
- Registrati per un Account AWS
- Crea un utente con accesso amministrativo

Fase 1: creazione di un'istanza EC2

Crea un'istanza Amazon EC2 da utilizzare per connetterti al database.

Per creare un'istanza EC2

1. Accedi AWS Management Console e apri la console Amazon EC2 all'indirizzo https://console.aws.amazon.com/ec2/.
2. Nell'angolo in alto a destra di AWS Management Console, scegli l'istanza EC2 Regione AWS in cui desideri creare l'istanza EC2.
3. Seleziona Pannello di controllo EC2, quindi Avvia istanza, come visualizzato di seguito.

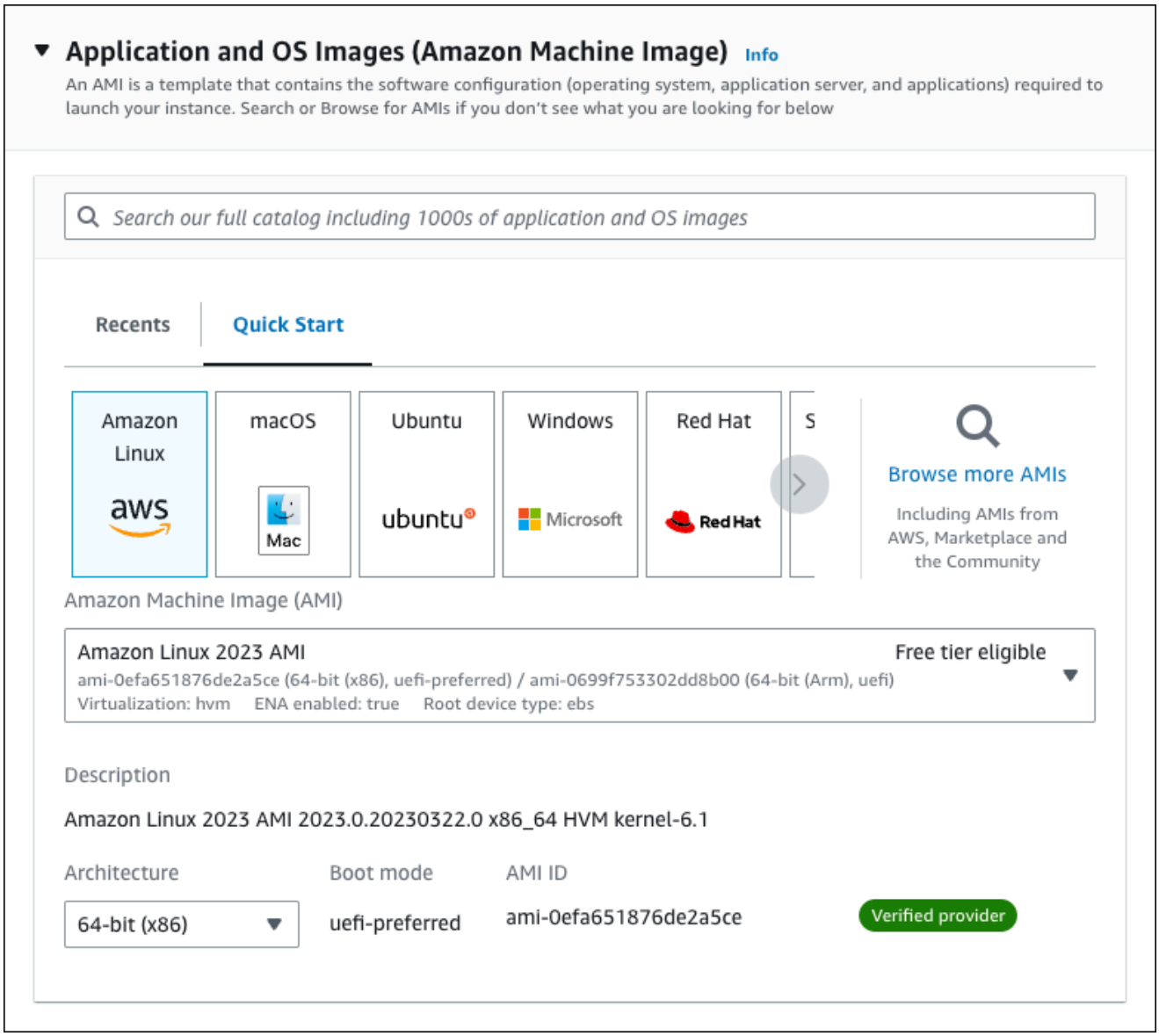


Viene visualizzata la pagina Avvia un'istanza.

4. Scegli le seguenti impostazioni nella pagina Avvia un'istanza.

   a. Nell'area Name and tags (Nome e tag), in Name (Nome) inserisci **`ec2-database-connect`**.

   b. In Immagini applicazione e sistema operativo (Amazon Machine Image), scegli Amazon Linux, quindi AMI Amazon Linux 2023. Mantieni le selezioni predefinite per le altre opzioni.



   c. In Instance type (Tipo di istanza), scegli t2.micro.

   d. In Key pair (login) (Coppia di chiavi (login), per Key pair name (Nome della coppia di chiavi), scegli una coppia di chiavi esistente. Per creare una nuova coppia di chiavi per l'istanza Amazon EC2, scegli Create new key pair (Crea nuova coppia di chiavi) e quindi utilizza la finestra Create key pair (Crea coppia di chiavi) per crearla.

Per ulteriori informazioni sulla creazione di una nuova coppia di chiavi, consulta Creazione di una coppia di chiavi nella Guida per l'utente di Amazon EC2 per istanze Linux.

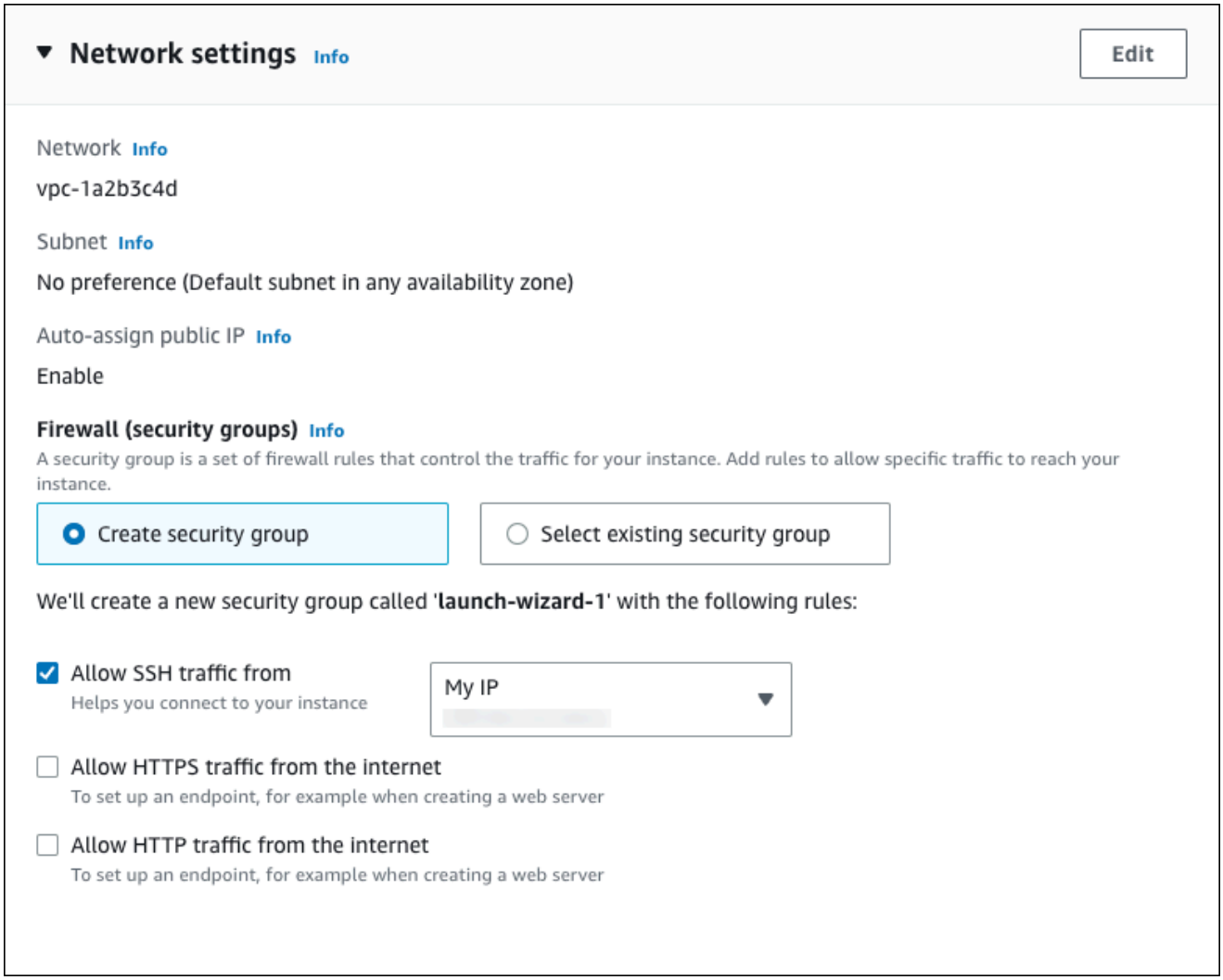
e. In Consenti traffico SSH, nell'area Impostazioni di rete scegliere l'origine delle connessioni SSH all'istanza EC2.

   È possibile scegliere My IP (Il mio IP) se l'indirizzo IP visualizzato è corretto per le connessioni SSH. In caso contrario, è possibile determinare l'indirizzo IP da utilizzare per connettersi alle istanze EC2 nel VPC utilizzando Secure Shell (SSH). Per determinare l'indirizzo IP pubblico, in una finestra o una scheda del browser diversa, è possibile utilizzare il servizio all'indirizzo https://checkip.amazonaws.com. Un esempio di indirizzo IP è 192.0.2.1/32.

   In molti casi, è possibile eseguire la connessione tramite un fornitore di servizi Internet (ISP) o con la protezione di un firewall senza un indirizzo IP statico. In tal caso, accertati di determinare l'intervallo di indirizzi IP utilizzati dai computer client.

   > **Warning**
   >
   > Se utilizzi `0.0.0.0/0` per l'accesso SSH, consenti a tutti gli indirizzi IP di accedere alle istanze EC2 pubbliche utilizzando SSH. Questo approccio è accettabile per un breve periodo di tempo in un ambiente di test, ma non è sicuro per gli ambienti di produzione. In produzione, autorizza solo un determinato indirizzo IP o un intervallo di indirizzi per accedere alle istanze EC2 utilizzando SSH.

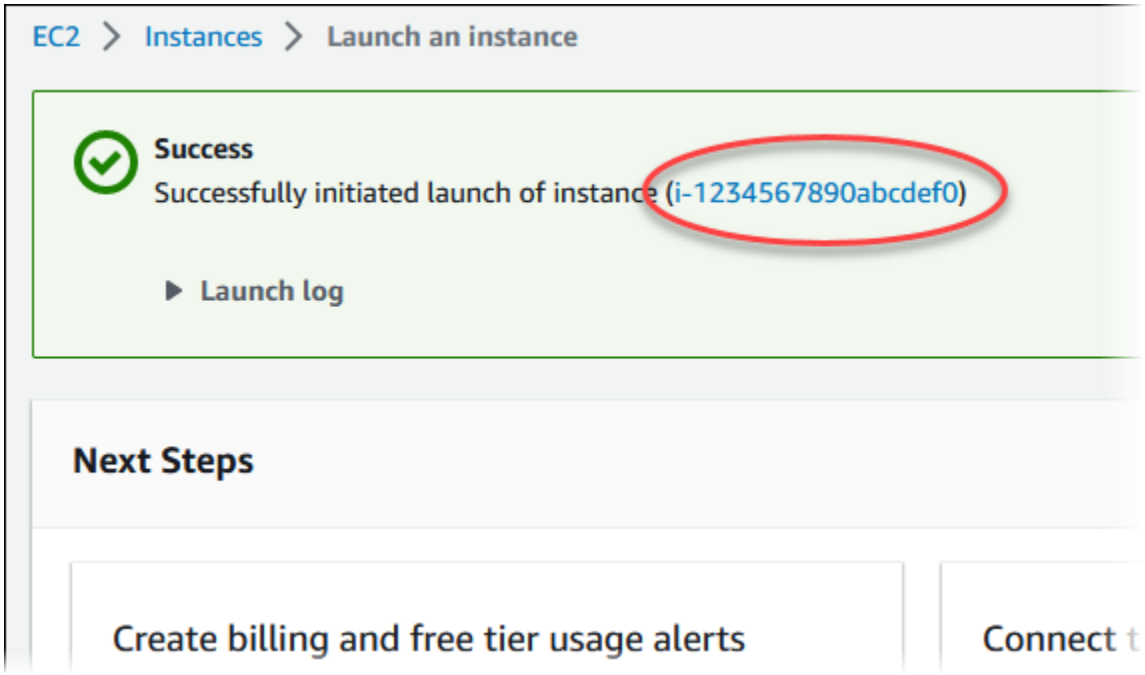
   L'immagine seguente mostra un esempio della sezione Impostazioni di rete.



f. Lascia i valori predefiniti per le sezioni rimanenti.

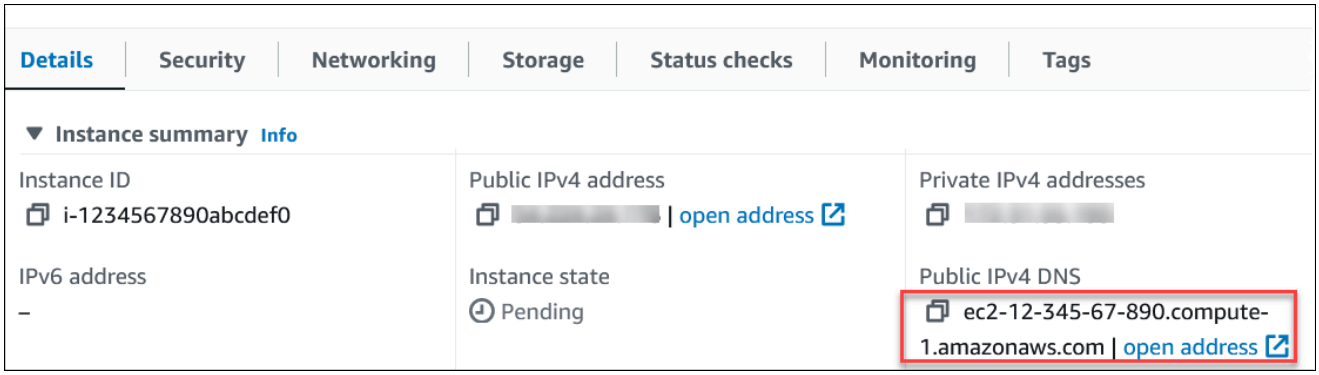
g. Analizza un riepilogo della configurazione dell'istanza EC2 nel pannello Riepilogo e, quando è tutto pronto, scegli Avvia istanza.

5. Nella pagina Stato avvio prendi nota dell'identificatore per la nuova istanza EC2, ad esempio: `i-1234567890abcdef0`.

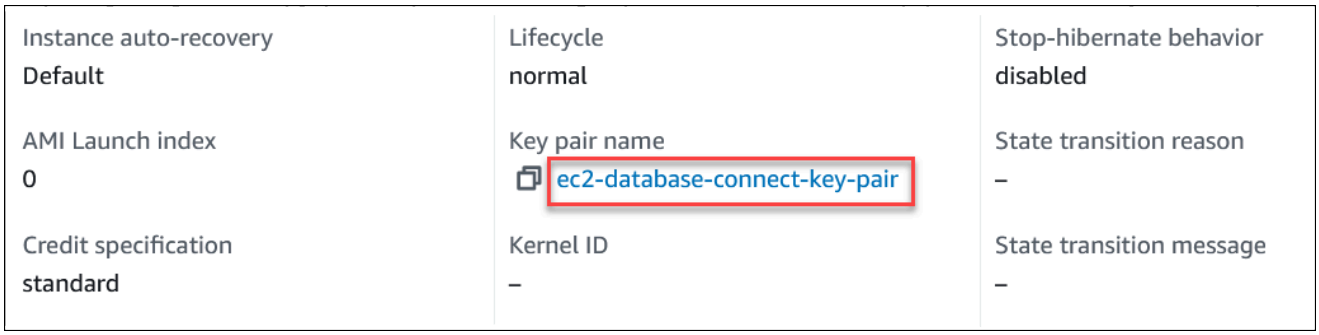


6. Scegli l'identificatore dell'istanza EC2 per aprire l'elenco delle istanze EC2, quindi seleziona l'istanza EC2.
7. Nella scheda Dettagli, annota i seguenti valori, necessari quando ti connetti tramite SSH:

   a. In Riepilogo istanza, annota il valore visualizzato in DNS IPv4 pubblico.



   b. In Dettagli istanza, annota il valore visualizzato in Nome coppia di chiavi.



8. Attendi che Stato dell'istanza diventi In esecuzione per l'istanza EC2 prima di continuare.

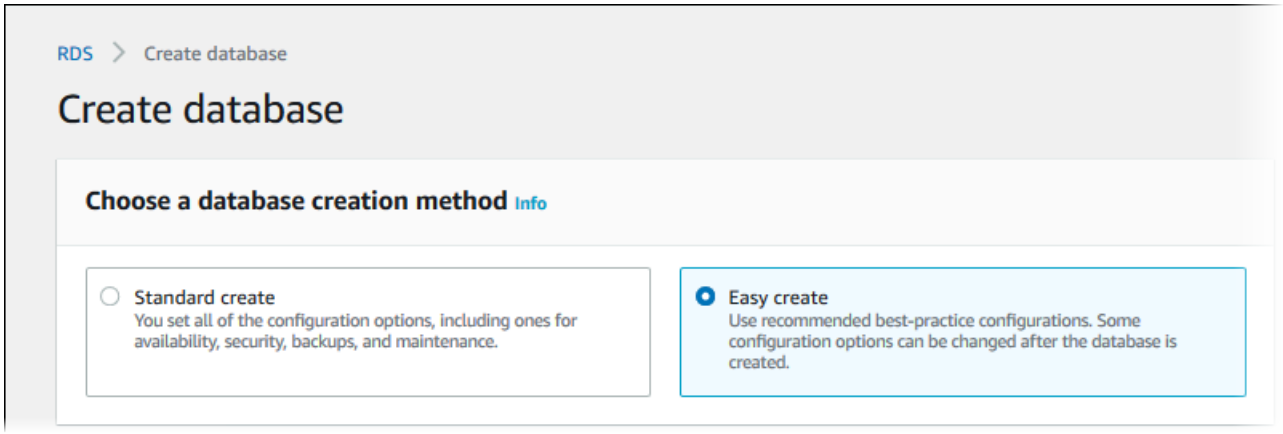
Fase 2: creazione di un'istanza database MySQL

L'istanza database rappresenta l'elemento di base di Amazon RDS. Questo è l'ambiente dove esegui i tuoi database MySQL.

In questo esempio, utilizzi la Creazione semplice per creare un'istanza database che esegue un motore di database MySQL con una classe di istanza database db.t3.micro.

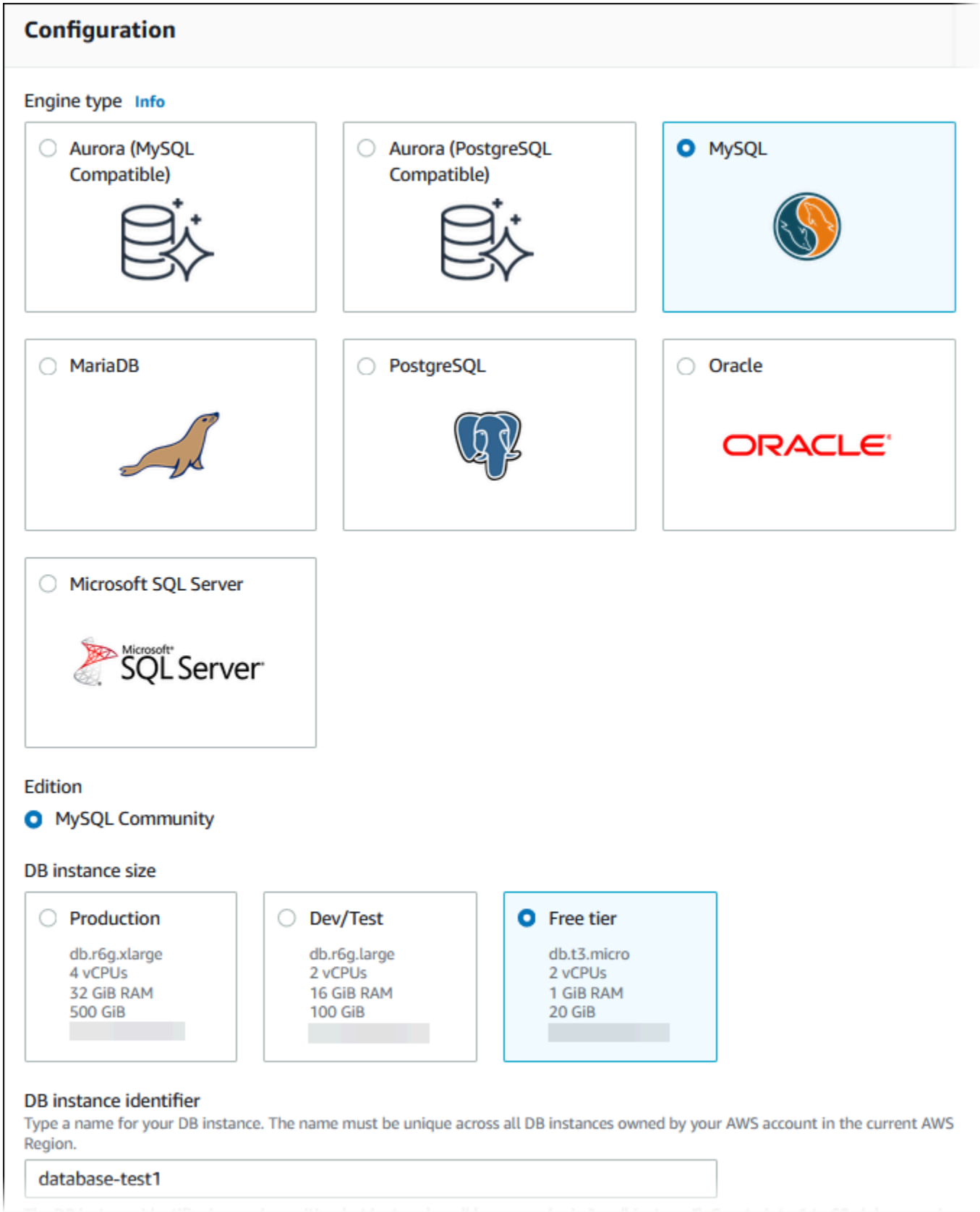
Per creare una istanza database MySQL con Easy Create (Creazione rapida)

1. Accedi AWS Management Console e apri la console Amazon RDS all'[indirizzo https://console.aws.amazon.com/rds/](https://console.aws.amazon.com/rds/).
2. Nell'angolo in alto a destra della console Amazon RDS, scegli quella che Regione AWS hai usato in precedenza per l'istanza EC2.
3. Nel riquadro di navigazione, scegliere Databases (Database).
4. Scegliere Create database (Crea database) e verificare che l'opzione Easy Create (Creazione rapida) sia selezionata.



5. In Configuration (Configurazione), seleziona MySQL.
6. Per DB instance size (Dimensione istanza database), seleziona Free tier (Piano gratuito).
7. Per l'identificatore dell'istanza DB, inserisci **`database-test1`**.
8. Per Nome utente master, inserisci un nome per l'utente master o lascia il nome predefinito.

   La pagina Create database (Crea database) la pagina dovrebbe apparire simile alla seguente immagine.

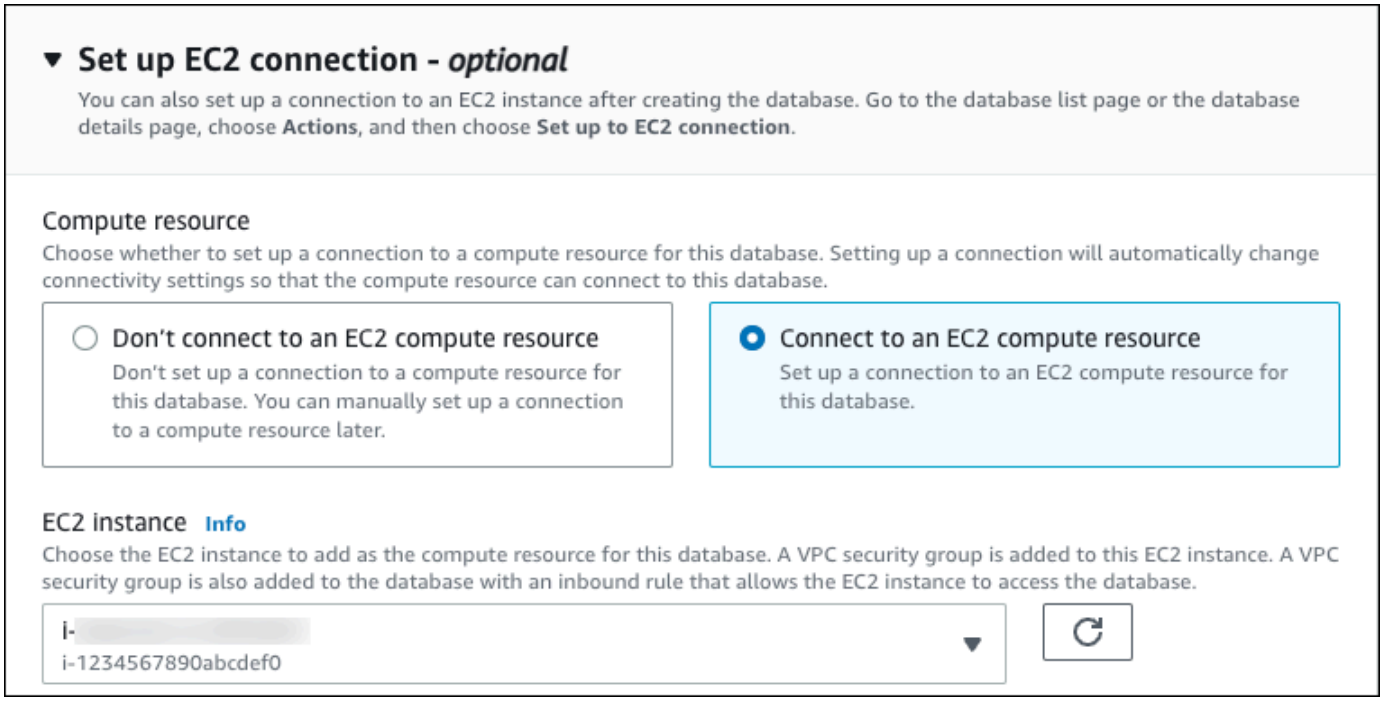

Configuration
Engine type Info
Aurora (MySQL Compatible)
Aurora (PostgreSQL Compatible)
MySQL
MariaDB
PostgreSQL
Oracle
ORACLE
Microsoft SQL Server
Microsoft SQL Server
Edition
MySQL Community
DB instance size
Production
db.r6g.xlarge
4 vCPUs
32 GiB RAM
500 GiB
Dev/Test
db.r6g.large
2 vCPUs
16 GiB RAM
100 GiB
Free tier
db.t3.micro
2 vCPUs
1 GiB RAM
20 GiB
DB instance identifier
Type a name for your DB instance. The name must be unique across all DB instances owned by your AWS account in the current AWS Region.
database-test1

9. Per utilizzare una password master generata automaticamente per l'istanza database, seleziona Genera automaticamente una password.

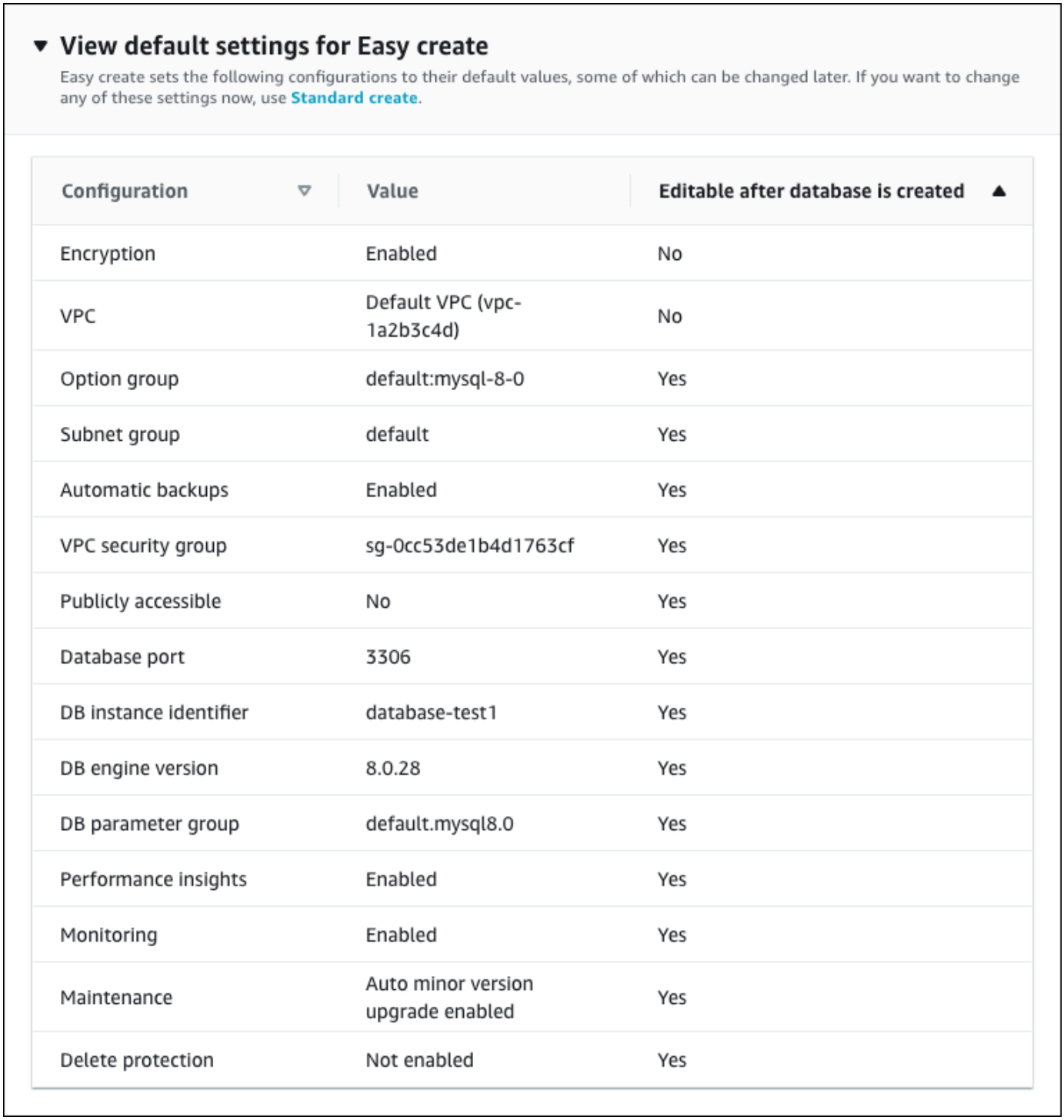
   Per inserire la password master, deseleziona la casella Genera automaticamente una password e inserisci la stessa password in Password master e Conferma password.

10. Per configurare una connessione con l'istanza EC2 creata in precedenza, apri Configura connessione EC2 - opzionale.

    Seleziona Connetti a una risorsa di calcolo EC2. Scegli l'istanza EC2 creata in precedenza.



11. (Facoltativo) Aprire View default settings for Easy create (Visualizza impostazioni predefinite per Creazione rapida).



View default settings for Easy create

Easy create sets the following configurations to their default values, some of which can be changed later. If you want to change any of these settings now, use **Standard create**.

Configuration	Value	Editable after database is created
Encryption	Enabled	No
VPC	Default VPC (vpc-1a2b3c4d)	No
Option group	default:mysql-8-0	Yes
Subnet group	default	Yes
Automatic backups	Enabled	Yes
VPC security group	sg-0cc53de1b4d1763cf	Yes
Publicly accessible	No	Yes
Database port	3306	Yes
DB instance identifier	database-test1	Yes
DB engine version	8.0.28	Yes
DB parameter group	default.mysql8.0	Yes
Performance insights	Enabled	Yes
Monitoring	Enabled	Yes
Maintenance	Auto minor version upgrade enabled	Yes
Delete protection	Not enabled	Yes

Puoi esaminare le impostazioni predefinite utilizzate con Easy create (Creazione rapida). La colonna Modificabile dopo la creazione del database mostra le opzioni che puoi modificare dopo aver creato il database.

- Se un'impostazione contiene No in quella colonna e desideri cambiarla, puoi utilizzare la Creazione standard per creare l'istanza database.
- Se un'impostazione contiene Sì in quella colonna e desideri cambiarla, puoi utilizzare la Creazione standard per creare l'istanza database o modificare l'istanza database dopo averla creata per cambiare l'impostazione.

12. Scegliere Crea database.

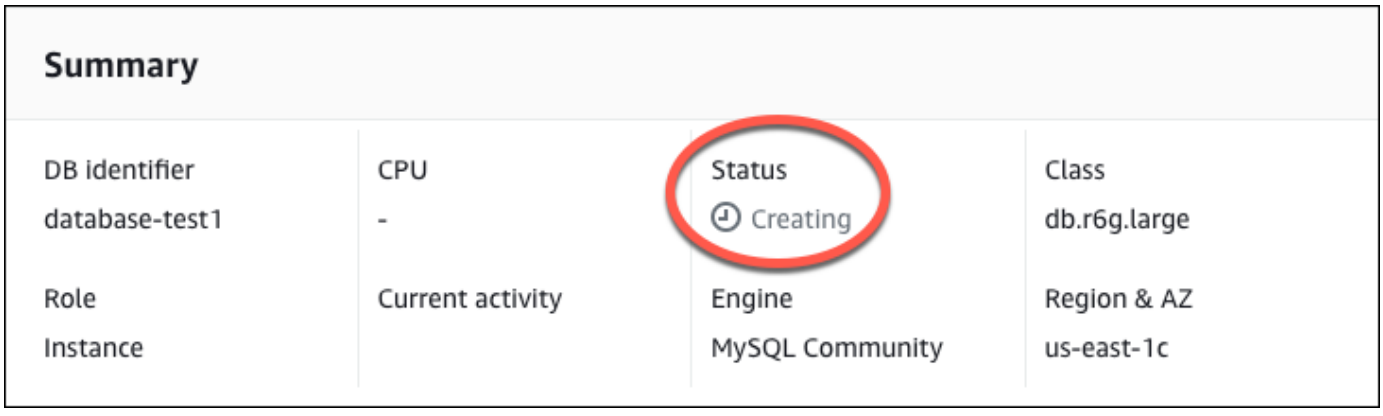
    Per vedere nome utente e password per l'istanza database, seleziona View credential details (Vedi dettagli delle credenziali).

    Per connetterti all'istanza database come utente principale, utilizza il nome utente e la password visualizzati.

    > **Important**
    >
    > Non potrai visualizzare di nuovo la password dell'utente principale. Se non la registri, potresti doverla modificare.
    > Se devi modificare la password dell'utente principale dopo che l'istanza database è disponibile, puoi modificare l'istanza database per eseguire tale operazione. Per ulteriori informazioni sulla modifica di un'istanza database , consulta Modifica di un'istanza database Amazon RDS.

13. Nell'elenco Database seleziona il nome della nuova istanza database MySQL per visualizzarne i dettagli.

    L'istanza database ha lo stato Creazione in corso fino a quando non è pronta per essere utilizzata.



Quando lo stato cambia in Available (Disponibile), puoi connettersi all'istanza database. A seconda della classe di istanza database e della quantità di storage, prima che la nuova istanza sia disponibile possono trascorrere fino a 20 minuti.

(Facoltativo) Crea VPC, istanza EC2 e istanza MySQL utilizzando AWS CloudFormation

Invece di usare la console per creare il tuo VPC, l'istanza EC2 e l'istanza MySQL, puoi usarla AWS CloudFormation per fornire AWS risorse trattando l'infrastruttura come codice. Per aiutarti a organizzare AWS le tue risorse in unità più piccole e più gestibili, puoi utilizzare la funzionalità nested stack. AWS CloudFormation Per ulteriori informazioni, consulta Creare uno stack sulla AWS CloudFormation console e Lavorare con gli stack annidati.

> **Important**
>
> AWS CloudFormation è gratuito, ma le risorse che CloudFormation crea sono attive. Ti verranno addebitati i costi di utilizzo standard per queste risorse fino alla loro cessazione. L'addebito totale sarà minimo. Per informazioni su come ridurre al minimo gli addebiti, vai al AWS piano gratuito.

Per creare le tue risorse utilizzando la AWS CloudFormation console, completa i seguenti passaggi:

- Passaggio 1: scarica il CloudFormation modello
- Passaggio 2: configura le tue risorse utilizzando CloudFormation

Scarica il CloudFormation modello

Un CloudFormation modello è un file di testo JSON o YAML che contiene le informazioni di configurazione sulle risorse che desideri creare nello stack. Questo modello crea anche un VPC e un bastion host per te insieme all'istanza RDS.

Per scaricare il file modello, apri il seguente link, modello CloudFormation MySQL.

Nella pagina Github, fai clic sul pulsante Scarica file raw per salvare il file YAML del modello.

Configura le tue risorse usando CloudFormation

> **Note**
>
> Prima di iniziare questo processo, assicurati di avere una coppia di chiavi per un'istanza EC2 nel tuo Account AWS. Per ulteriori informazioni, consulta Coppie di chiavi Amazon EC2 e istanze Linux.

Quando utilizzi il AWS CloudFormation modello, devi selezionare i parametri corretti per assicurarti che le risorse vengano create correttamente. Segui la procedura riportata di seguito:

1. Accedi AWS Management Console e apri la AWS CloudFormation console all'indirizzo https://console.aws.amazon.com/cloudformation.
2. Scegli Crea stack.
3. Nella sezione Specificare il modello, seleziona Carica un file modello dal computer, quindi scegli Avanti.
4. Nella pagina Specificare i dettagli dello stack, imposta i seguenti parametri:
   a. Imposta il nome dello stack su MySQL TestStack.
   b. In Parametri, imposta le zone di disponibilità selezionando tre zone di disponibilità.
   c. Nella configurazione Linux Bastion Host, in Key Name, seleziona una coppia di chiavi per accedere alla tua istanza EC2.
   d. Nelle impostazioni di configurazione di Linux Bastion Host, imposta l'intervallo IP consentito sul tuo indirizzo IP. Per connetterti alle istanze EC2 nel tuo VPC utilizzando Secure Shell (SSH), determina il tuo indirizzo IP pubblico utilizzando il servizio all'indirizzo https://checkip.amazonaws.com. Un esempio di indirizzo IP è 192.0.2.1/32.

> **Warning**
>
> Se utilizzi `0.0.0.0/0` per l'accesso SSH, consenti a tutti gli indirizzi IP di accedere alle istanze EC2 pubbliche utilizzando SSH. Questo approccio è accettabile per un breve periodo di tempo in un ambiente di test, ma non è sicuro per gli ambienti di produzione. In produzione, autorizza solo un determinato indirizzo IP o un intervallo di indirizzi per accedere alle istanze EC2 utilizzando SSH.

e. Nella configurazione generale del database, imposta la classe dell'istanza del database su db.t3.micro.

f. Imposta il nome del database su. **`database-test1`**

g. Per Nome utente principale del database, inserisci un nome per l'utente principale.

h. Imposta la password utente principale di Manage DB con Secrets Manager su `false` per questo tutorial.

i. Per la password del database, imposta una password a tua scelta. Ricorda questa password per ulteriori passaggi del tutorial.

j. In Configurazione dell'archiviazione del database, imposta il tipo di archiviazione del database su gp2.

k. Nella configurazione Database Monitoring, imposta Enable RDS Performance Insights su false.

l. Lascia tutte le altre impostazioni come valori predefiniti. Fate clic su Avanti per continuare.

5. Nella pagina Configura le opzioni dello stack, lascia tutte le opzioni predefinite. Fai clic su Avanti per continuare.
6. Nella pagina Review stack, seleziona Invia dopo aver verificato le opzioni del database e dell'host Linux bastion.

Una volta completato il processo di creazione dello stack, visualizza gli stack con nomi BastionStacke RDSNS per annotare le informazioni necessarie per connetterti al database. Per ulteriori informazioni, vedere Visualizzazione dei dati e delle risorse AWS CloudFormation dello stack su. AWS Management Console

Fase 3: connessione a un'istanza database MySQL

È possibile utilizzare qualsiasi applicazione client SQL standard per la connessione all'istanza database. In questo esempio, ti connetti a un'istanza database MySQL utilizzando il client della linea di comando mysql.

Per eseguire la connessione a un'istanza database MySQL

1. Individuare l'endpoint (nome DNS) e il numero di porta per l'istanza database.

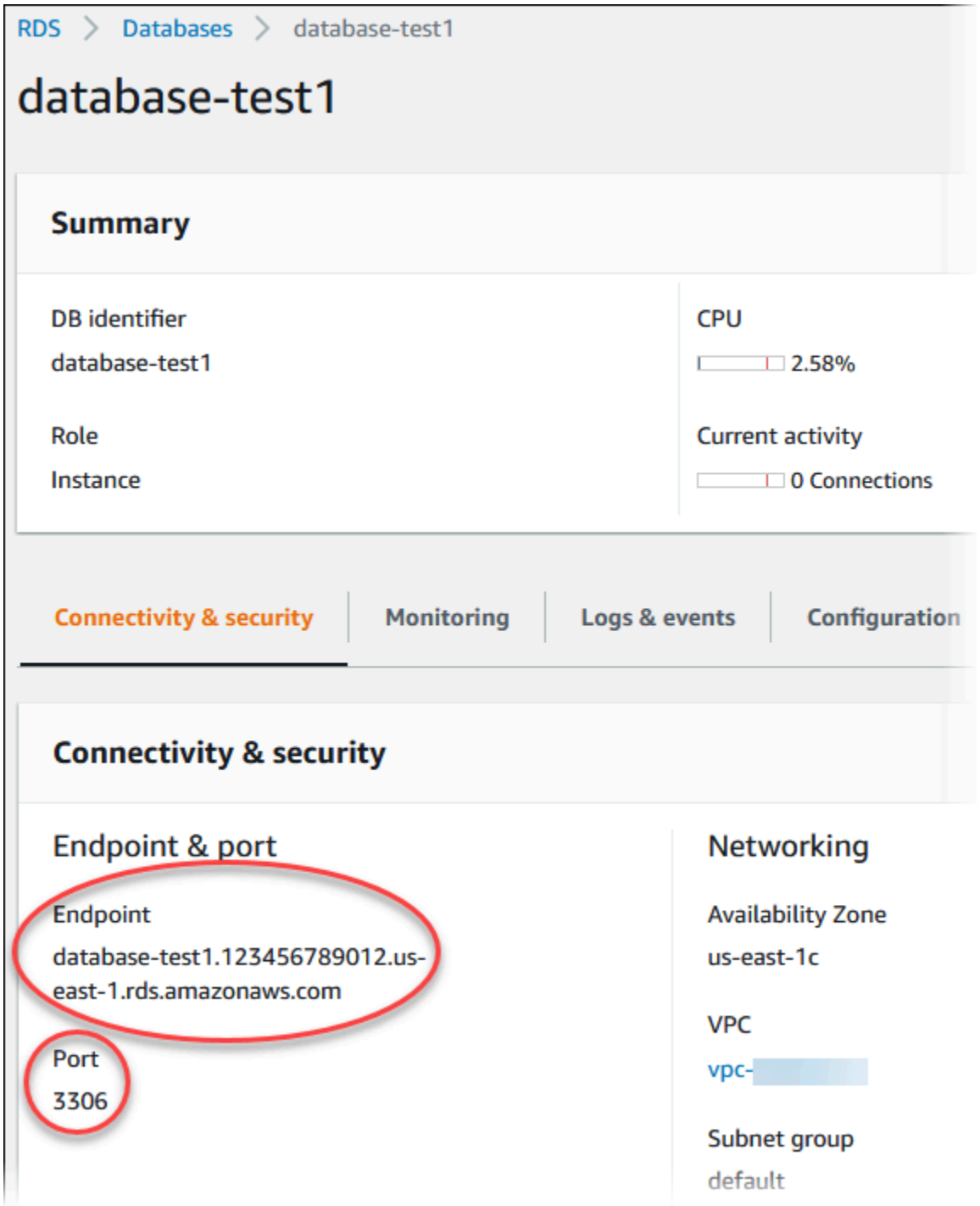
   a. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.

   b. Nell'angolo in alto a destra della console Amazon RDS, scegli l' Regione AWS istanza DB.

c. Nel riquadro di navigazione, scegliere Databases (Database).

d. Scegliere il nome dell'istanza database MySQL per visualizzarne i dettagli.

e. Nella scheda Connectivity & security (Connettività e sicurezza), copiare l'endpoint. Annotare anche il numero di porta. L'endpoint e il numero di porta sono necessari per la connessione all'istanza database.



2. Esegui la connessione all'istanza EC2 creata in precedenza seguendo la procedura riportata in Connessione all'istanza di Linux nella Guida per l'utente per istanze Linux di Amazon EC2.

   Ti consigliamo di connetterti all'istanza EC2 tramite SSH. Se l'utilità client SSH è installata su Windows, Linux o Mac, puoi connetterti all'istanza utilizzando il comando nel seguente formato:

   ```
   ssh -i location_of_pem_file ec2-user@ec2-instance-public-dns-name
   ```

   Ad esempio, supponi che `ec2-database-connect-key-pair.pem` sia archiviato in `/dir1` su Linux e che il DNS IPv4 pubblico per l'istanza EC2 sia `ec2-12-345-678-90.compute-1.amazonaws.com`. Il comando SSH sarà simile al seguente:

   ```
   ssh -i /dir1/ec2-database-connect-key-pair.pem ec2-
   user@ec2-12-345-678-90.compute-1.amazonaws.com
   ```

3. Ottieni le ultime correzioni di bug e gli aggiornamenti di sicurezza aggiornando il software sulla tua istanza EC2. A questo scopo, eseguire il comando seguente.

   > **Note**
   >
   > L'opzione `-y` installa gli aggiornamenti senza chiedere conferma. Per esaminare gli aggiornamenti prima di installarli, omettere questa opzione.

   ```
   sudo dnf update -y
   ```

4. Per installare il client della linea di comando MySQL da MariaDB su Amazon Linux 2023, esegui il comando seguente:

   ```
   sudo dnf install mariadb105
   ```

5. Effettua la connessione all'istanza database MySQL. Ad esempio, specifica il comando seguente: Questa azione consente di connetterti all'istanza database MySQL utilizzando il client MySQL.

   Sostituisci l'endpoint dell'istanza database (nome DNS) per *`endpoint`* e il nome utente master utilizzato per *`admin`*. Devi fornire la password master utilizzata quando viene richiesta una password.

```
mysql -h endpoint -P 3306 -u admin -p
```

Dopo aver immesso la password per l'utente, l'output dovrebbe essere analogo a quanto mostrato di seguito.

```
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 3082
Server version: 8.0.28 Source distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]>
```

Per ulteriori informazioni sulla connessione a un'istanza database di MySQL, consulta Connessione a un'istanza database che esegue il motore di database di MySQL. In caso di mancata connessione all'istanza database, consulta Impossibile connettersi all'istanza database di Amazon RDS.

Per motivi di sicurezza, la best practice è utilizzare connessioni crittografate. Utilizzare una connessione MySQL non crittografata solo quando il client e il server sono nello stesso VPC e la rete è attendibile. Per ulteriori informazioni sull'uso di connessioni crittografate, consulta Connessione dal client a riga di comando MySQL con SSL/TLS (crittografato).

6. Eseguire comandi SQL.

   Ad esempio, il seguente comando SQL mostra la data e l'ora correnti:

```
SELECT CURRENT_TIMESTAMP;
```

Fase 4: eliminazione dell'istanza EC2 e dell'istanza database

Dopo la connessione e l'esplorazione dell'istanza EC2 e dell'istanza database di esempio che hai creato, eliminale per evitare di ricevere l'addebito dei relativi costi.

Se in passato AWS CloudFormation creavi risorse, salta questo passaggio e vai al passaggio successivo.

Per eliminare l'istanza EC2

1. Accedi AWS Management Console e apri la console Amazon EC2 all'indirizzo https://console.aws.amazon.com/ec2/.
2. Nel riquadro di navigazione, seleziona Istanze.
3. Seleziona l'istanza EC2 e scegli Stato istanza, Termina istanza.
4. Quando viene richiesta la conferma, seleziona Terminate (Interrompi).

Per ulteriori informazioni sull'eliminazione di un'istanza EC2, consulta Interruzione di un'istanza nella Guida per l'utente per istanze Linux di Amazon EC2.

Per eliminare l'istanza database senza snapshot database finale

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, scegliere Databases (Database).
3. Scegliere l'istanza database da eliminare.
4. In Actions (Azioni), selezionare Delete (Elimina).
5. Deseleziona Creare uno snapshot finale? e Conserva backup automatizzati.
6. Completa la conferma e scegli Elimina.

(Facoltativo) Elimina l'istanza EC2 e l'istanza DB create con CloudFormation

Se prima creavi AWS CloudFormation risorse, elimina lo CloudFormation stack dopo esserti connesso ed esplorato l'istanza EC2 e l'istanza DB di esempio, in modo che non ti vengano più addebitati costi.

Per eliminare le risorse CloudFormation

1. Apri la AWS CloudFormation console.
2. Nella pagina Stacks CloudFormationconsole, seleziona lo stack principale (lo stack senza il nome VPCStack o RDSNS). BastionStack
3. Scegli Elimina.
4. Seleziona Elimina stack quando viene richiesta la conferma.

Per ulteriori informazioni sull'eliminazione di uno stack in CloudFormation, consulta Eliminazione di uno stack sulla console nella Guida per l' AWS CloudFormationutente.AWS CloudFormation

(Facoltativo) Connessione dell'istanza database a una funzione Lambda

Puoi anche connettere la tua istanza database RDS per MySQL a una risorsa di elaborazione serverless Lambda. Le funzioni Lambda consentono di eseguire il codice senza il provisioning o la gestione dell'infrastruttura. Una funzione Lambda consente inoltre di rispondere automaticamente alle richieste di esecuzione del codice su qualsiasi scala, da una dozzina di eventi al giorno a centinaia al secondo. Per ulteriori informazioni, consulta Connessione automatica di una funzione Lambda e di un'istanza database.

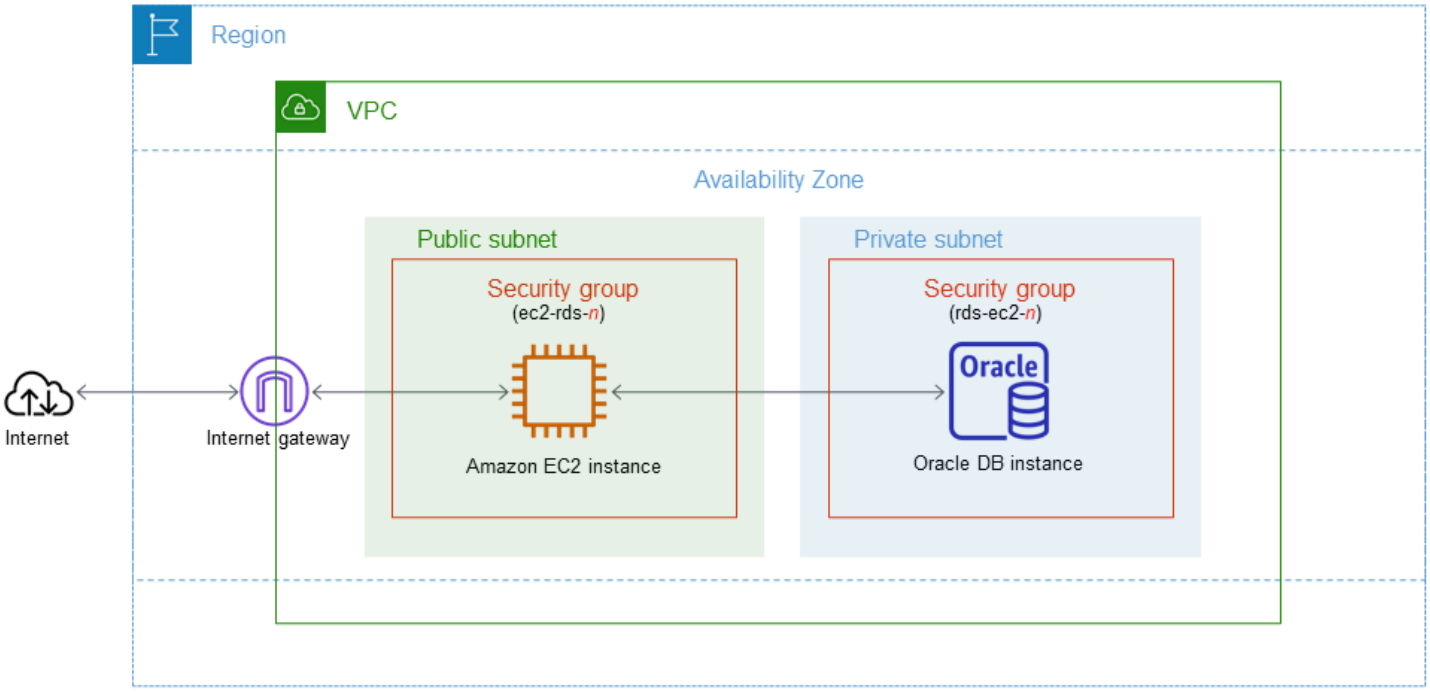
Creazione e connessione a un'istanza database Oracle

Questo tutorial illustra come creare un'istanza EC2 e un'istanza database RDS per Oracle. Il tutorial mostra come accedere all'istanza database dall'istanza EC2 utilizzando il client Oracle standard. Come best practice, questo tutorial spiega come creare un'istanza database privata in un cloud privato virtuale (VPC). Nella maggior parte dei casi, le risorse presenti nello stesso VPC, come le istanze EC2, possono accedere all'istanza database, mentre le risorse esterne al VPC non possono accedervi.

Dopo aver completato il tutorial, è presente una sottorete pubblica e una privata in ogni zona di disponibilità del VPC. In una zona di disponibilità, l'istanza EC2 si trova nella sottorete pubblica mentre l'istanza database si trova nella sottorete privata.

> **Important**
>
> Non è previsto alcun costo per la creazione di un AWS account. Tuttavia, completando questo tutorial, potresti incorrere in costi per le AWS risorse che utilizzi. È possibile eliminare queste risorse dopo aver completato l'esercitazione se non sono più necessarie.

Il seguente diagramma illustra la configurazione al completamento del tutorial.



Questo tutorial ti consente di creare le tue risorse utilizzando uno dei seguenti metodi:

1. Usa AWS Management Console - Fase 2: creazione di un'istanza database Oracle e Fase 1: creazione di un'istanza EC2
2. Utilizzare AWS CloudFormation per creare l'istanza del database e l'istanza EC2 - (Facoltativo) Crea VPC, istanza EC2 e istanza Oracle DB utilizzando AWS CloudFormation

Il primo metodo utilizza Easy create per creare un'istanza Oracle DB privata con. AWS Management Console Qui, si specificano solo il tipo di motore DB, la dimensione dell'istanza DB e l'identificatore dell'istanza DB. Easy create (Creazione rapida) utilizza l'impostazione predefinita per altre opzioni di configurazione.

Se invece utilizzi Standard create, puoi specificare più opzioni di configurazione quando crei un'istanza DB. Queste opzioni includono impostazioni per la disponibilità, la sicurezza, i backup e la manutenzione. Per creare un'istanza database pubblica, è necessario utilizzare la Creazione standard. Per informazioni, consulta Creazione di un'istanza database Amazon RDS.

Argomenti

Prerequisiti

Prima di iniziare, completa le fasi descritte in questa sezione:

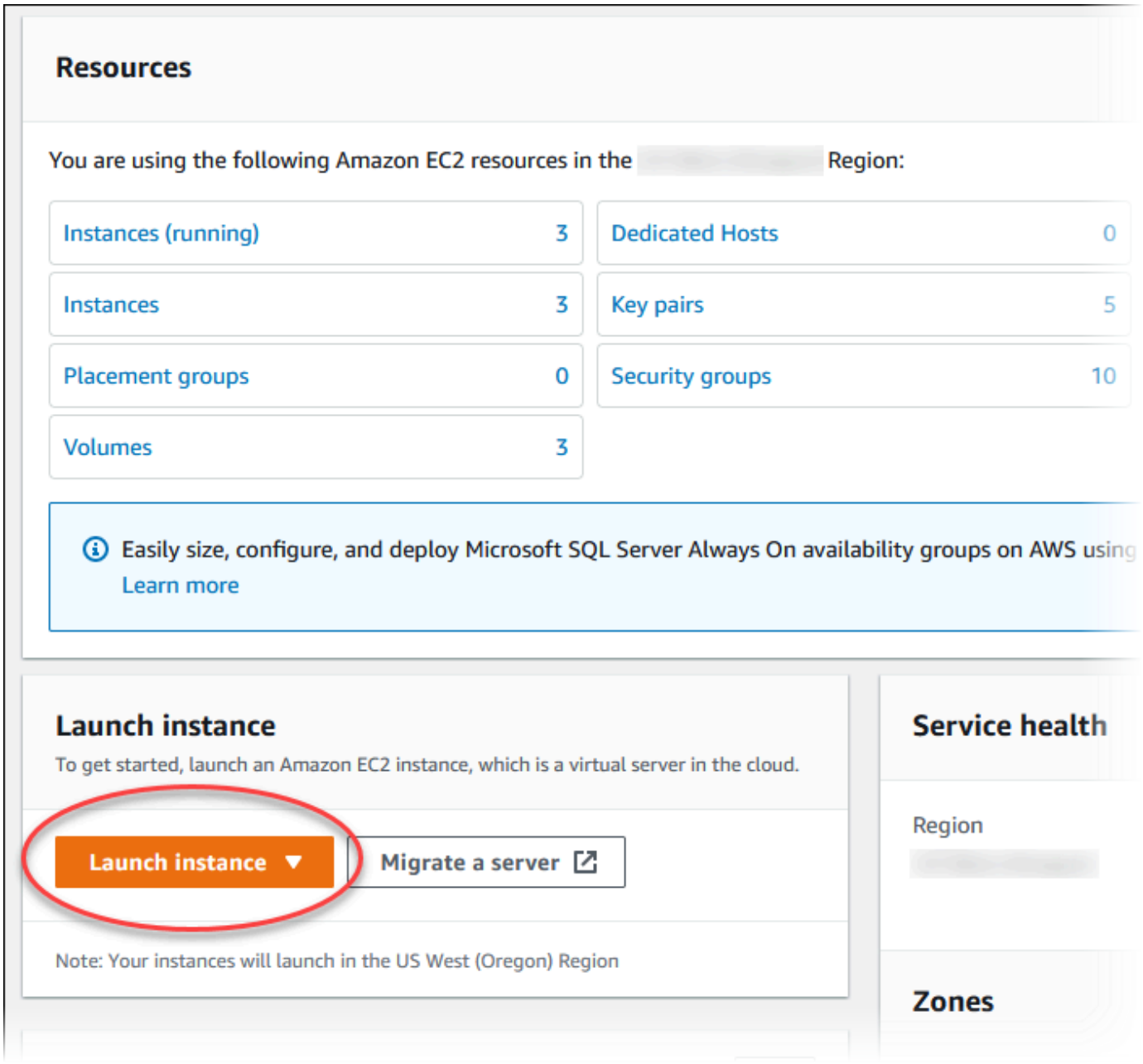
- Registrati per un Account AWS
- Crea un utente con accesso amministrativo

Fase 1: creazione di un'istanza EC2

Crea un'istanza Amazon EC2 da utilizzare per connetterti al database.

Per creare un'istanza EC2

1. Accedi AWS Management Console e apri la console Amazon EC2 all'indirizzo https://console.aws.amazon.com/ec2/.
2. Nell'angolo in alto a destra di AWS Management Console, scegli l'istanza EC2 Regione AWS in cui desideri creare l'istanza EC2.
3. Seleziona Pannello di controllo EC2, quindi Avvia istanza, come visualizzato di seguito.

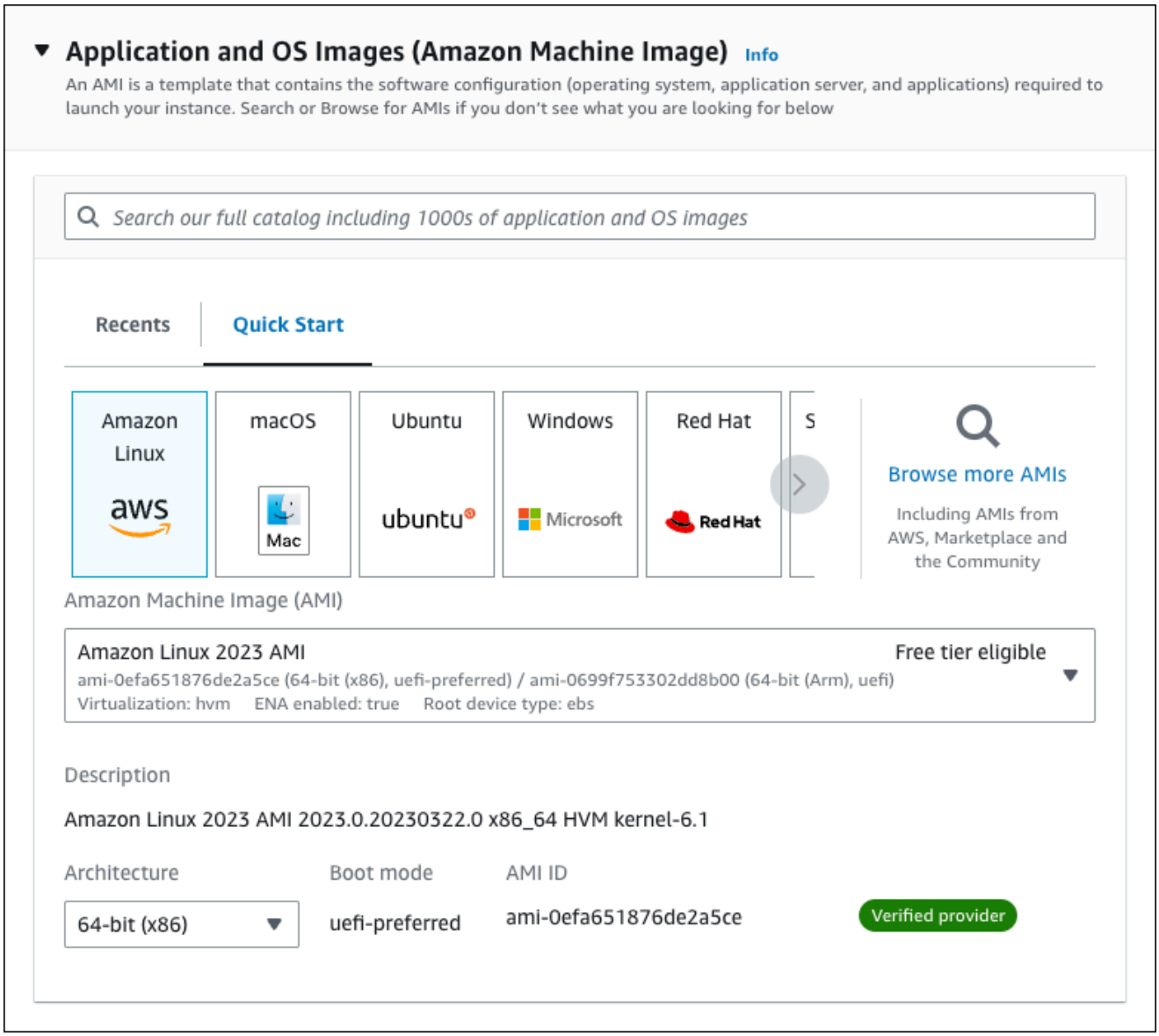


Viene visualizzata la pagina Avvia un'istanza.

4. Scegli le seguenti impostazioni nella pagina Avvia un'istanza.

   a. Nell'area Name and tags (Nome e tag), in Name (Nome) inserisci **`ec2-database-connect`**.

   b. In Immagini applicazione e sistema operativo (Amazon Machine Image), scegli Amazon Linux, quindi AMI Amazon Linux 2023. Mantieni le selezioni predefinite per le altre opzioni.



   c. In Instance type (Tipo di istanza), scegli t2.micro.

   d. In Key pair (login) (Coppia di chiavi (login), per Key pair name (Nome della coppia di chiavi), scegli una coppia di chiavi esistente. Per creare una nuova coppia di chiavi per l'istanza Amazon EC2, scegli Create new key pair (Crea nuova coppia di chiavi) e quindi utilizza la finestra Create key pair (Crea coppia di chiavi) per crearla.

Per ulteriori informazioni sulla creazione di una nuova coppia di chiavi, consulta Creazione di una coppia di chiavi nella Guida per l'utente di Amazon EC2 per istanze Linux.

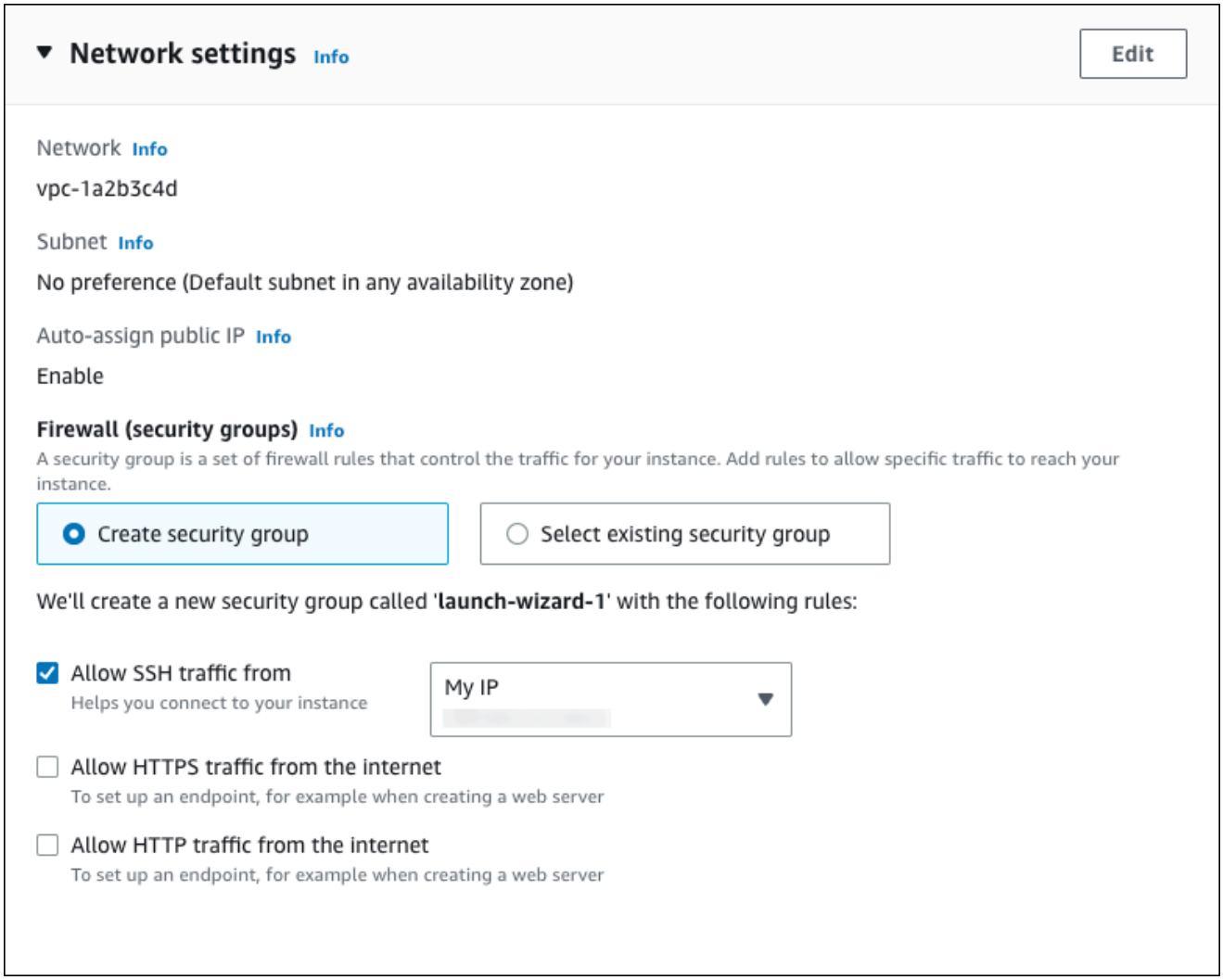
e. In Consenti traffico SSH, nell'area Impostazioni di rete scegliere l'origine delle connessioni SSH all'istanza EC2.

   È possibile scegliere My IP (Il mio IP) se l'indirizzo IP visualizzato è corretto per le connessioni SSH. In caso contrario, è possibile determinare l'indirizzo IP da utilizzare per connettersi alle istanze EC2 nel VPC utilizzando Secure Shell (SSH). Per determinare l'indirizzo IP pubblico, in una finestra o una scheda del browser diversa, è possibile utilizzare il servizio all'indirizzo https://checkip.amazonaws.com. Un esempio di indirizzo IP è 192.0.2.1/32.

   In molti casi, è possibile eseguire la connessione tramite un fornitore di servizi Internet (ISP) o con la protezione di un firewall senza un indirizzo IP statico. In tal caso, accertati di determinare l'intervallo di indirizzi IP utilizzati dai computer client.

   > **Warning**
   >
   > Se utilizzi `0.0.0.0/0` per l'accesso SSH, consenti a tutti gli indirizzi IP di accedere alle istanze EC2 pubbliche utilizzando SSH. Questo approccio è accettabile per un breve periodo di tempo in un ambiente di test, ma non è sicuro per gli ambienti di produzione. In produzione, autorizza solo un determinato indirizzo IP o un intervallo di indirizzi per accedere alle istanze EC2 utilizzando SSH.

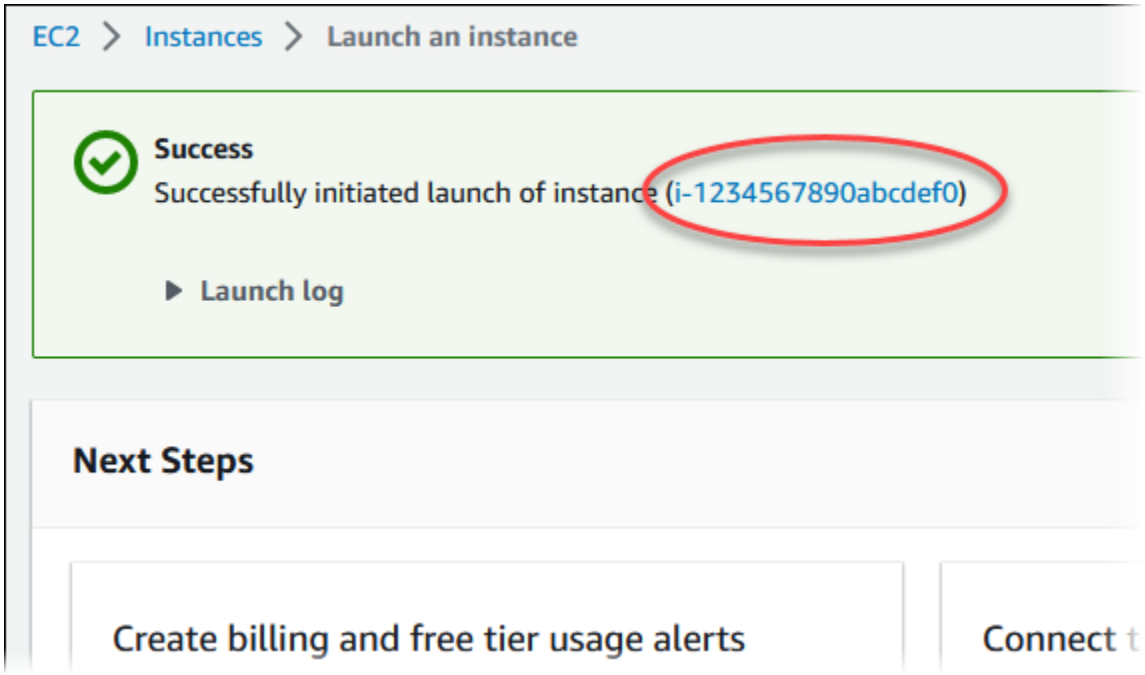
   L'immagine seguente mostra un esempio della sezione Impostazioni di rete.



f. Lascia i valori predefiniti per le sezioni rimanenti.

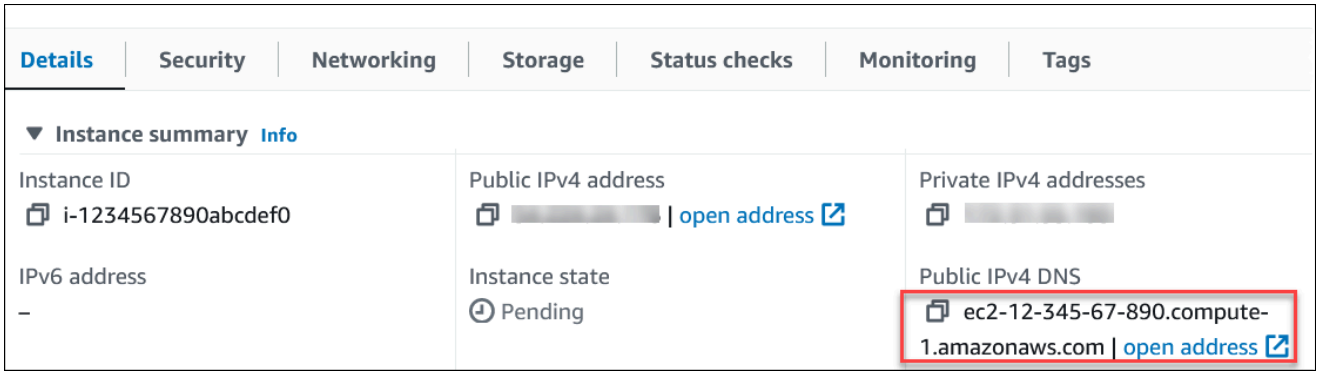
g. Analizza un riepilogo della configurazione dell'istanza EC2 nel pannello Riepilogo e, quando è tutto pronto, scegli Avvia istanza.

5. Nella pagina Stato avvio prendi nota dell'identificatore per la nuova istanza EC2, ad esempio: `i-1234567890abcdef0`.

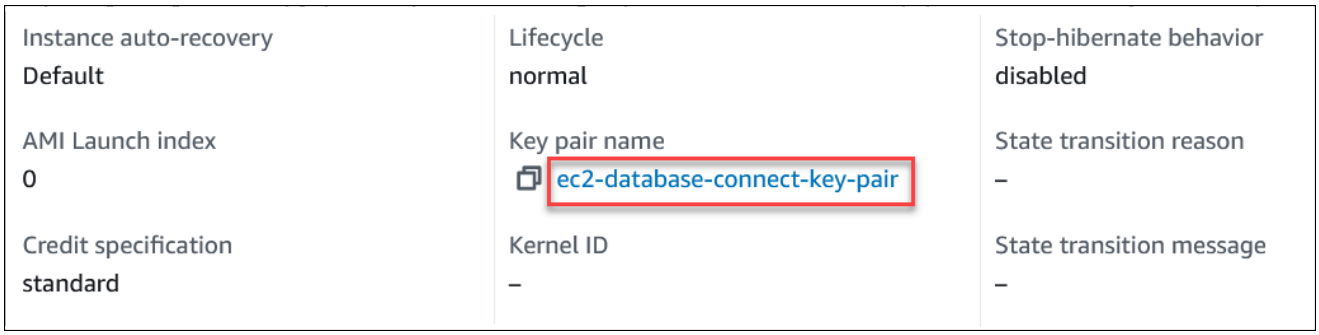


6. Scegli l'identificatore dell'istanza EC2 per aprire l'elenco delle istanze EC2, quindi seleziona l'istanza EC2.
7. Nella scheda Dettagli, annota i seguenti valori, necessari quando ti connetti tramite SSH:

   a. In Riepilogo istanza, annota il valore visualizzato in DNS IPv4 pubblico.



   b. In Dettagli istanza, annota il valore visualizzato in Nome coppia di chiavi.



8. Attendi che Stato dell'istanza diventi In esecuzione per l'istanza EC2 prima di continuare.

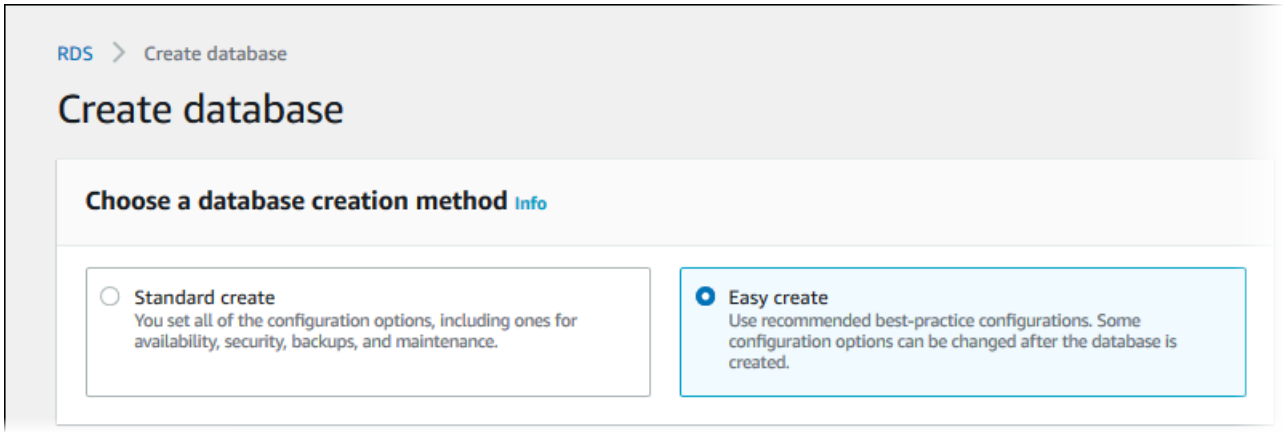
Fase 2: creazione di un'istanza database Oracle

L'istanza database rappresenta l'elemento di base di Amazon RDS. Questo è l'ambiente dove esegui i tuoi database Oracle.

Per questo esempio, utilizza Creazione semplice per creare una istanza database che esegue un motore di database Oracle con una classe di istanza database db.m5.large.

Per creare una istanza database Oracle con Creazione semplice

1. Accedi AWS Management Console e apri la console Amazon RDS all'[indirizzo https://console.aws.amazon.com/rds/](https://console.aws.amazon.com/rds/).
2. Nell'angolo in alto a destra della console Amazon RDS, scegli l'istanza database Regione AWS in cui desideri creare l'istanza DB.
3. Nel riquadro di navigazione, scegliere Databases (Database).
4. Scegliere Create database (Crea database) e verificare che l'opzione Easy Create (Creazione rapida) sia selezionata.



5. In Configuration (Configurazione), seleziona Oracle.
6. Per DB instance size (Dimensione istanza database), seleziona Dev/Test.
7. Per l'identificatore dell'istanza DB, inserisci **`database-test1`**.
8. Per Nome utente master, inserisci un nome per l'utente master o lascia il nome predefinito.

   La pagina Create database (Crea database) la pagina dovrebbe apparire simile alla seguente immagine.

Configuration

Engine type Info

- Aurora (MySQL Compatible)
- Aurora (PostgreSQL Compatible)
- MySQL
- MariaDB
- PostgreSQL
- Oracle
- Microsoft SQL Server

Edition

- Oracle Enterprise Edition
  Affordable and full-featured database management system supporting up to 16 vCPUs.
- Oracle Standard Edition Two
  Affordable and full-featured database management system supporting up to 16 vCPUs. Oracle Database Standard Edition Two is a replacement for Standard Edition and Standard Edition One.

DB instance size

- Production
  db.r5.large
  2 vCPUs
  16 GiB RAM
  500 GiB
- Dev/Test
  db.m5.large
  2 vCPUs
  8 GiB RAM
  100 GiB

DB instance identifier

Type a name for your DB instance. The name must be unique across all DB instances owned by your AWS account in the current AWS Region.

database-test1

The DB instance identifier is case-insensitive, but is stored as all lowercase (as in "mydbinstance"). Constraints: 1 to 60 alphanumeric characters or hyphens. First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.

Master username Info

Type a login ID for the master user of your DB instance.

admin

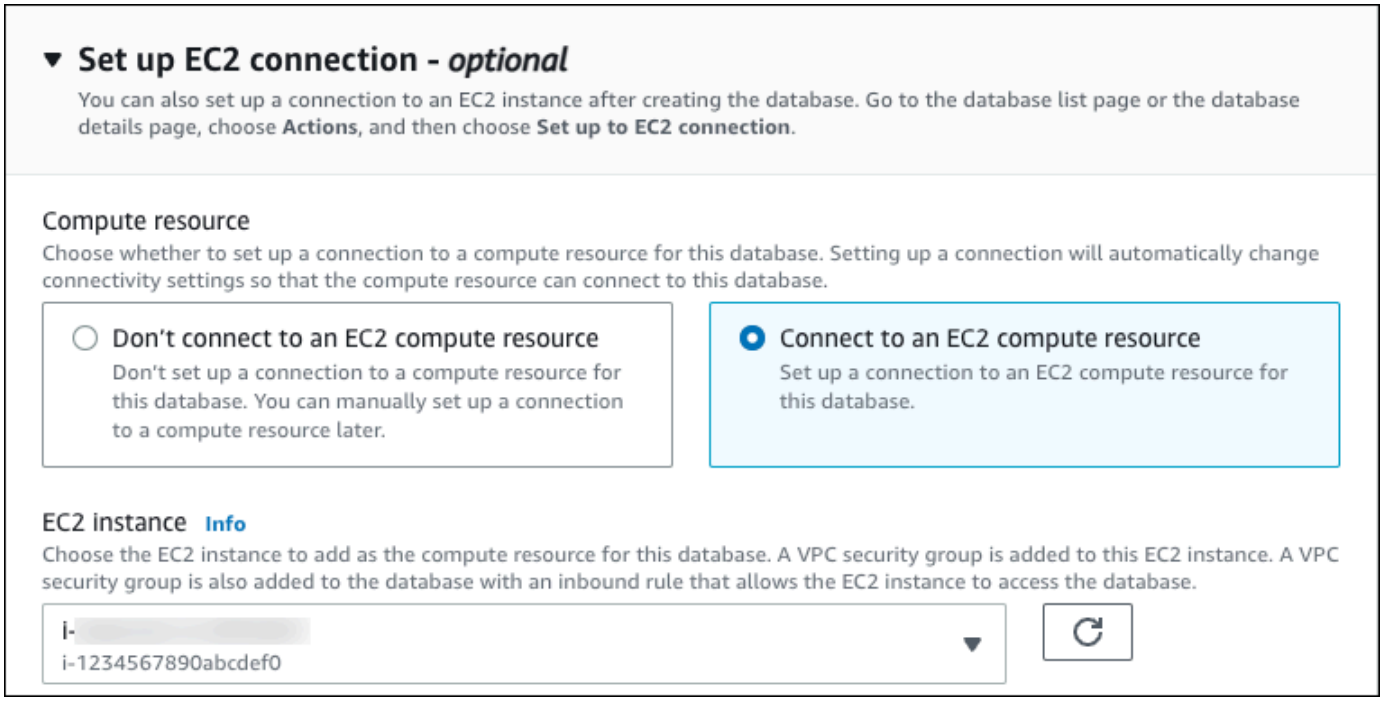
1 to 16 alphanumeric characters. First character must be a letter.

9. Per utilizzare una password master generata automaticamente per l'istanza database, seleziona Genera automaticamente una password.

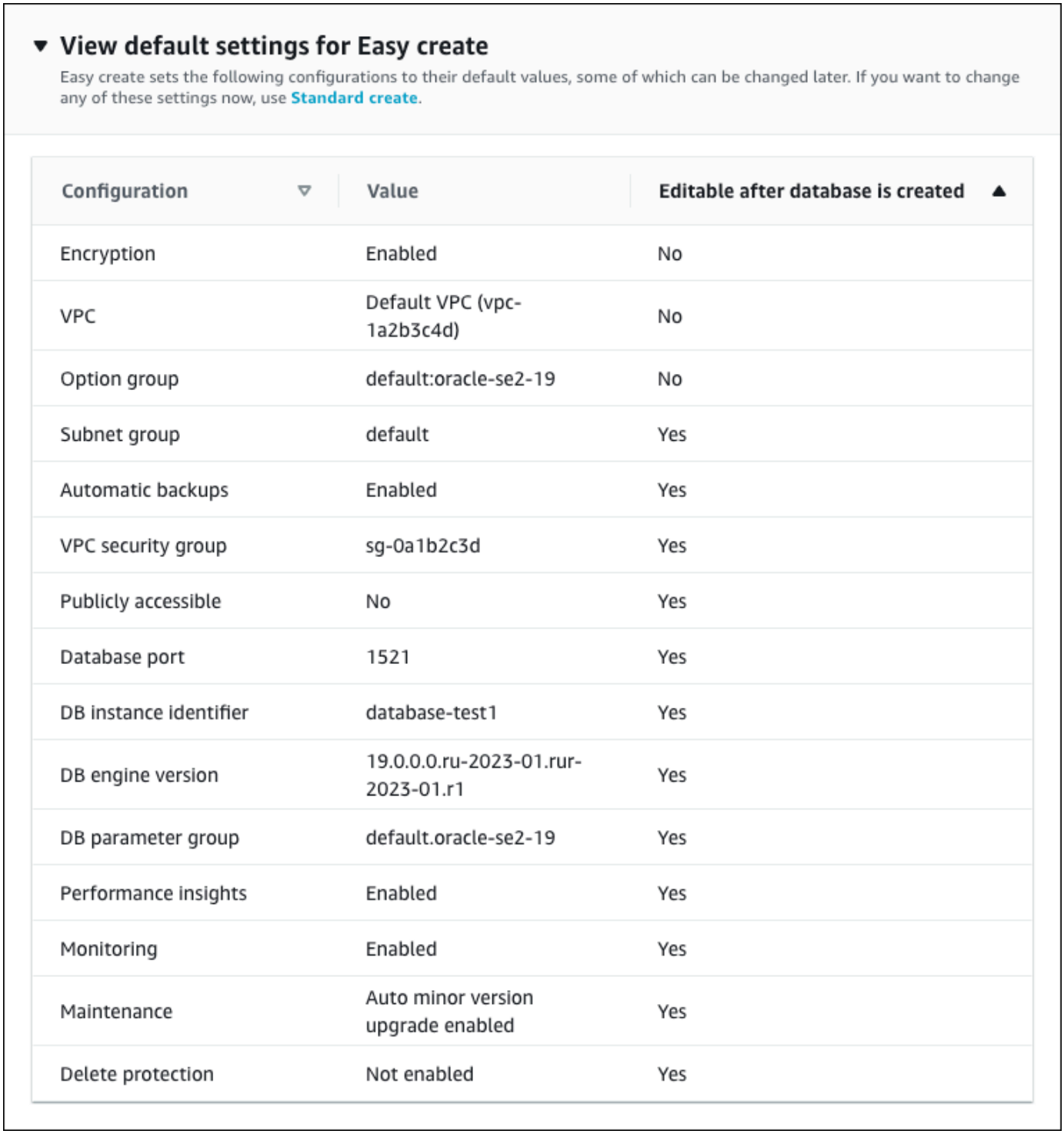
   Per inserire la password master, deseleziona la casella Genera automaticamente una password e inserisci la stessa password in Password master e Conferma password.

10. Per configurare una connessione con l'istanza EC2 creata in precedenza, apri Configura connessione EC2 - opzionale.

    Seleziona Connetti a una risorsa di calcolo EC2. Scegli l'istanza EC2 creata in precedenza.



11. Apri Visualizza le impostazioni predefinite per la creazione Semplice.



View default settings for Easy create

Easy create sets the following configurations to their default values, some of which can be changed later. If you want to change any of these settings now, use **Standard create**.

Configuration	Value	Editable after database is created
Encryption	Enabled	No
VPC	Default VPC (vpc-1a2b3c4d)	No
Option group	default:oracle-se2-19	No
Subnet group	default	Yes
Automatic backups	Enabled	Yes
VPC security group	sg-0a1b2c3d	Yes
Publicly accessible	No	Yes
Database port	1521	Yes
DB instance identifier	database-test1	Yes
DB engine version	19.0.0.0.ru-2023-01.rur-2023-01.r1	Yes
DB parameter group	default.oracle-se2-19	Yes
Performance insights	Enabled	Yes
Monitoring	Enabled	Yes
Maintenance	Auto minor version upgrade enabled	Yes
Delete protection	Not enabled	Yes

Puoi esaminare le impostazioni predefinite utilizzate con Easy create (Creazione rapida). La colonna Modificabile dopo la creazione del database mostra le opzioni che puoi modificare dopo aver creato il database.

- Se un'impostazione contiene No in quella colonna e desideri cambiarla, puoi utilizzare la Creazione standard per creare l'istanza database.
- Se un'impostazione contiene Sì in quella colonna e desideri cambiarla, puoi utilizzare la Creazione standard per creare l'istanza database o modificare l'istanza database dopo averla creata per cambiare l'impostazione.

12. Scegliere Crea database.

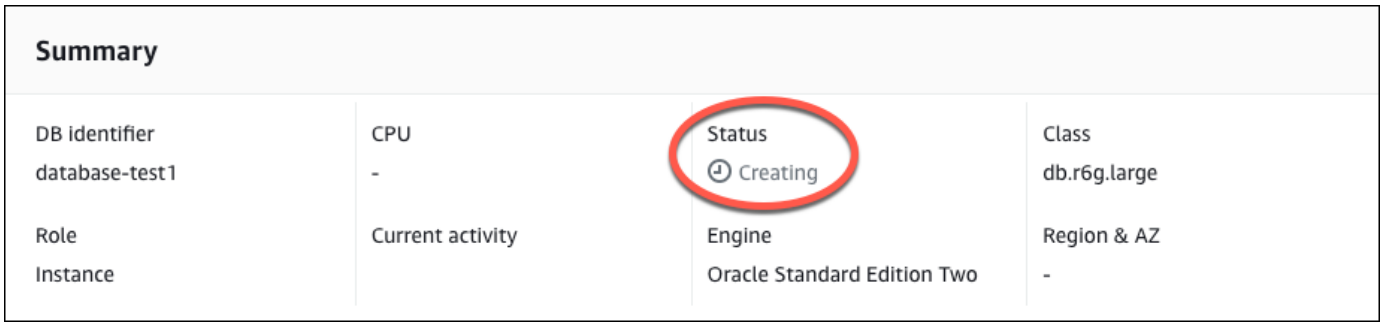
    Per vedere nome utente e password per l'istanza database, seleziona View credential details (Vedi dettagli delle credenziali).

    Per connetterti all'istanza database come utente principale, utilizza il nome utente e la password visualizzati.

    > **Important**
    >
    > Non potrai visualizzare di nuovo la password dell'utente principale. Se non la registri, potresti doverla modificare.
    > Se devi modificare la password dell'utente principale dopo che l'istanza database è disponibile, puoi modificare l'istanza database per eseguire tale operazione. Per ulteriori informazioni sulla modifica di un'istanza database , consulta Modifica di un'istanza database Amazon RDS.

13. Nell'elenco Database seleziona il nome della nuova istanza database Oracle per visualizzarne i dettagli.

    L'istanza database ha lo stato Creazione in corso fino a quando non è pronta per essere utilizzata.



    Quando lo stato cambia in Available (Disponibile), puoi connettersi all'istanza database. A seconda della classe di istanza database e della quantità di storage, prima che la nuova istanza

sia disponibile possono trascorrere fino a 20 minuti. Durante la creazione dell'istanza database, puoi passare alla fase successiva e creare un'istanza EC2.

(Facoltativo) Crea VPC, istanza EC2 e istanza Oracle DB utilizzando AWS CloudFormation

Invece di utilizzare la console per creare il tuo VPC, l'istanza EC2 e l'istanza Oracle DB, puoi utilizzarla AWS CloudFormation per fornire AWS risorse trattando l'infrastruttura come codice. Per aiutarti a organizzare AWS le tue risorse in unità più piccole e più gestibili, puoi utilizzare la funzionalità AWS CloudFormation nested stack. Per ulteriori informazioni, consulta Creare uno stack sulla AWS CloudFormation console e Lavorare con gli stack annidati.

> **Important**
>
> AWS CloudFormation è gratuito, ma le risorse che CloudFormation crea sono attive. Ti verranno addebitati i costi di utilizzo standard per queste risorse fino alla loro cessazione. L'addebito totale sarà minimo. Per informazioni su come ridurre al minimo gli addebiti, vai al AWS piano gratuito.

Per creare le tue risorse utilizzando la AWS CloudFormation console, completa i seguenti passaggi:

- Passaggio 1: scarica il CloudFormation modello
- Passaggio 2: configura le tue risorse utilizzando CloudFormation

Scarica il CloudFormation modello

Un CloudFormation modello è un file di testo JSON o YAML che contiene le informazioni di configurazione sulle risorse che desideri creare nello stack. Questo modello crea anche un VPC e un bastion host per te insieme all'istanza RDS.

Per scaricare il file modello, apri il seguente link, Oracle template. CloudFormation

Nella pagina Github, fai clic sul pulsante Scarica file raw per salvare il file YAML del modello.

Configura le tue risorse usando CloudFormation

> **Note**
>
> Prima di iniziare questo processo, assicurati di avere una coppia di chiavi per un'istanza EC2 nel tuo Account AWS. Per ulteriori informazioni, consulta Coppie di chiavi Amazon EC2 e istanze Linux.

Quando utilizzi il AWS CloudFormation modello, devi selezionare i parametri corretti per assicurarti che le risorse vengano create correttamente. Segui la procedura riportata di seguito:

1. Accedi AWS Management Console e apri la AWS CloudFormation console all'indirizzo https://console.aws.amazon.com/cloudformation.
2. Scegli Crea stack.
3. Nella sezione Specificare il modello, seleziona Carica un file modello dal computer, quindi scegli Avanti.
4. Nella pagina Specificare i dettagli dello stack, imposta i seguenti parametri:
   a. Imposta il nome dello stack su. OracleTestStack
   b. In Parametri, imposta le zone di disponibilità selezionando tre zone di disponibilità.
   c. Nella configurazione Linux Bastion Host, in Key Name, seleziona una coppia di chiavi per accedere alla tua istanza EC2.
   d. Nelle impostazioni di configurazione di Linux Bastion Host, imposta l'intervallo IP consentito sul tuo indirizzo IP. Per connetterti alle istanze EC2 nel tuo VPC utilizzando Secure Shell (SSH), determina il tuo indirizzo IP pubblico utilizzando il servizio all'indirizzo https://checkip.amazonaws.com. Un esempio di indirizzo IP è 192.0.2.1/32.

> **Warning**
>
> Se utilizzi `0.0.0.0/0` per l'accesso SSH, consenti a tutti gli indirizzi IP di accedere alle istanze EC2 pubbliche utilizzando SSH. Questo approccio è accettabile per un breve periodo di tempo in un ambiente di test, ma non è sicuro per gli ambienti di produzione. In produzione, autorizza solo un determinato indirizzo IP o un intervallo di indirizzi per accedere alle istanze EC2 utilizzando SSH.

e. Nella configurazione generale del database, imposta la classe dell'istanza del database su db.t3.micro.

f. Imposta il nome del database su. **`database-test1`**

g. Per Nome utente principale del database, inserisci un nome per l'utente principale.

h. Imposta la password utente principale di Manage DB con Secrets Manager su `false` per questo tutorial.

i. Per la password del database, imposta una password a tua scelta. Ricorda questa password per ulteriori passaggi del tutorial.

j. In Configurazione dell'archiviazione del database, imposta il tipo di archiviazione del database su gp2.

k. Nella configurazione Database Monitoring, imposta Enable RDS Performance Insights su false.

l. Lascia tutte le altre impostazioni come valori predefiniti. Fate clic su Avanti per continuare.

5. Nella pagina Configura le opzioni dello stack, lascia tutte le opzioni predefinite. Fai clic su Avanti per continuare.
6. Nella pagina Review stack, seleziona Invia dopo aver verificato le opzioni del database e dell'host Linux bastion.

Una volta completato il processo di creazione dello stack, visualizza gli stack con nomi BastionStacke RDSNS per annotare le informazioni necessarie per connetterti al database. Per ulteriori informazioni, vedere Visualizzazione dei dati e delle risorse AWS CloudFormation dello stack su. AWS Management Console

Fase 3: connessione del client SQL a un'istanza database Oracle.

È possibile utilizzare qualsiasi applicazione client SQL standard per la connessione all'istanza database. In questo esempio, ti connetti a un'istanza database Oracle utilizzando il client della linea di comando Oracle.

Per connettersi a un'istanza database Oracle

1. Individuare l'endpoint (nome DNS) e il numero di porta per l'istanza database.

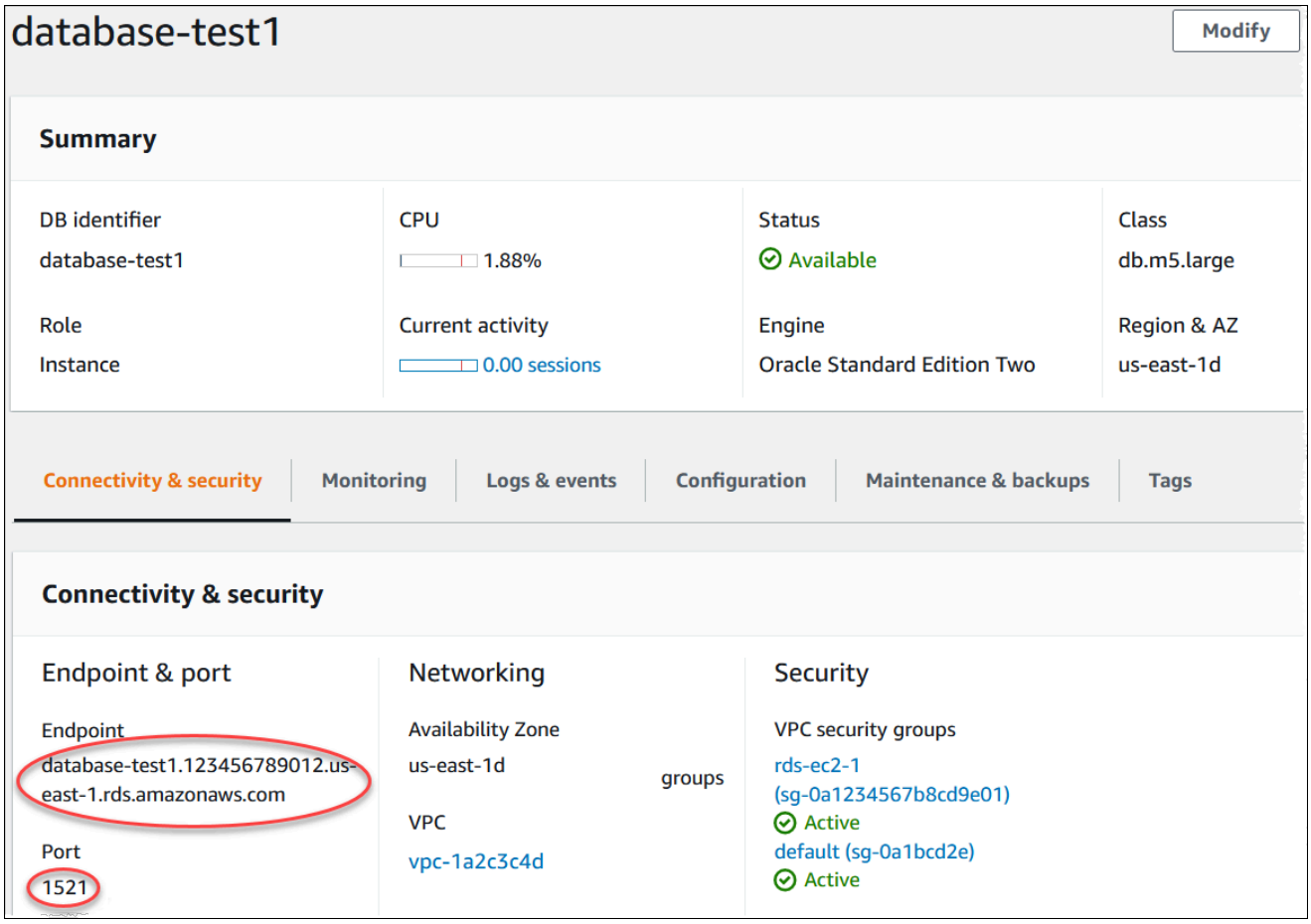
   a. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.

   b. Nell'angolo superiore destro della console Amazon RDS, scegli la Regione AWS dell'istanza database.

c. Nel riquadro di navigazione, scegli Databases (Database).

d. Scegliere il nome dell'istanza database Oracle per visualizzarne i dettagli.

e. Nella scheda Connectivity & security (Connettività e sicurezza), copiare l'endpoint. Annotare anche il numero di porta. L'endpoint e il numero di porta sono necessari per la connessione all'istanza database.



2. Esegui la connessione all'istanza EC2 creata in precedenza seguendo la procedura riportata in Connessione all'istanza di Linux nella Guida per l'utente per istanze Linux di Amazon EC2.

   Ti consigliamo di connetterti all'istanza EC2 tramite SSH. Se l'utilità client SSH è installata su Windows, Linux o Mac, puoi connetterti all'istanza utilizzando il comando nel seguente formato:

   ```
   ssh -i location_of_pem_file ec2-user@ec2-instance-public-dns-name
   ```

   Ad esempio, supponi che `ec2-database-connect-key-pair.pem` sia archiviato in `/dir1` su Linux e che il DNS IPv4 pubblico per l'istanza EC2 sia

`ec2-12-345-678-90.compute-1.amazonaws.com`. Il comando SSH sarà simile al seguente:

```
ssh -i /dir1/ec2-database-connect-key-pair.pem ec2-
user@ec2-12-345-678-90.compute-1.amazonaws.com
```

3. Ottieni le ultime correzioni di bug e gli aggiornamenti di sicurezza aggiornando il software sulla tua istanza EC2. A tale scopo, utilizzare il comando seguente.

> **Note**
>
> L'opzione `-y` installa gli aggiornamenti senza chiedere conferma. Per esaminare gli aggiornamenti prima di installarli, omettere questa opzione.

```
sudo dnf update -y
```

4. In un browser web, passa a https://www.oracle.com/database/technologies/instant-client/linux-x86-64-downloads.html.
5. Per la versione più recente del database visualizzata nella pagina web, copia i collegamenti .rpm (non i collegamenti .zip) per il pacchetto Instant Client Basic e il pacchetto SQL*Plus. Ad esempio, i seguenti link si riferiscono alla versione 21.9 di Oracle Database:
   - https://download.oracle.com/otn_software/linux/instantclient/219000/ -21.9.0.0.0-1.el8.x86_64.rpm oracle-instantclient-basic
   - https://download.oracle.com/otn_software/linux/instantclient/219000/ oracle-instantclient-sqlplus -21.9.0.0.0-1.el8.x86_64.rpm
6. Nella sessione SSH, esegui il comando `wget` per scaricare i file .rpm dai collegamenti che hai ottenuto nel passaggio precedente. L'esempio seguente scarica i file .rpm per Oracle Database versione 21.9:

```
wget https://download.oracle.com/otn_software/linux/instantclient/219000/oracle-
instantclient-basic-21.9.0.0.0-1.el8.x86_64.rpm
wget https://download.oracle.com/otn_software/linux/instantclient/219000/oracle-
instantclient-sqlplus-21.9.0.0.0-1.el8.x86_64.rpm
```

7. Installa i pacchetti eseguendo il comando `dnf` come segue:

```
sudo dnf install oracle-instantclient-*.rpm
```

8. Avvia SQL*Plus e stabilisci una connessione all'istanza database Oracle. Ad esempio, specifica il comando seguente:

   Sostituisci l'endpoint dell'istanza database (nome DNS) per *oracle-db-instance-endpoint* e il nome utente master utilizzato per *admin*. Quando usi Creazione semplice per Oracle, il nome del database èDATABASE. Devi fornire la password master utilizzata quando viene richiesta una password.

   ```
   sqlplus admin@oracle-db-instance-endpoint:1521/DATABASE
   ```

   Dopo aver immesso la password per l'utente, l'output dovrebbe essere analogo a quanto mostrato di seguito.

   ```
   SQL*Plus: Release 21.0.0.0.0 - Production on Wed Mar 1 16:41:28 2023
   Version 21.9.0.0.0

   Copyright (c) 1982, 2022, Oracle.  All rights reserved.

   Enter password:
   Last Successful login time: Wed Mar 01 2023 16:30:52 +00:00

   Connected to:
   Oracle Database 19c Standard Edition 2 Release 19.0.0.0.0 - Production
   Version 19.18.0.0.0

   SQL>
   ```

   Per ulteriori informazioni sulla connessione a un'istanza database RDS per Oracle, consulta Connessione all'istanza database RDS per Oracle. In caso di mancata connessione all'istanza database, consulta Impossibile connettersi all'istanza database di Amazon RDS.

   Per motivi di sicurezza, la best practice è utilizzare connessioni crittografate. Utilizza una connessione Oracle non crittografata solo quando il client e il server sono nello stesso VPC e la rete è attendibile. Per ulteriori informazioni sull'uso di connessioni crittografate, consulta Protezione delle connessioni di istanze database di Oracle.

9. Eseguire comandi SQL.

Ad esempio, il seguente comando SQL mostra la data corrente:

```
SELECT SYSDATE FROM DUAL;
```

Fase 4: eliminazione dell'istanza EC2 e dell'istanza database

Dopo la connessione e l'esplorazione dell'istanza EC2 e dell'istanza database di esempio che hai creato, eliminale per evitare di ricevere l'addebito dei relativi costi.

Se in passato creavi risorse, salta questo passaggio e vai AWS CloudFormation al passaggio successivo.

Per eliminare l'istanza EC2

1. Accedi AWS Management Console e apri la console Amazon EC2 all'indirizzo https://console.aws.amazon.com/ec2/.
2. Nel riquadro di navigazione, seleziona Istanze.
3. Seleziona l'istanza EC2 e scegli Stato istanza, Termina istanza.
4. Quando viene richiesta la conferma, seleziona Terminate (Interrompi).

Per ulteriori informazioni sull'eliminazione di un'istanza EC2, consulta Interruzione di un'istanza nella Guida per l'utente per istanze Linux di Amazon EC2.

Per eliminare l'istanza database senza snapshot database finale

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, scegliere Databases (Database).
3. Scegliere l'istanza database da eliminare.
4. In Actions (Azioni), selezionare Delete (Elimina).
5. Deseleziona Creare uno snapshot finale? e Conserva backup automatizzati.
6. Completa la conferma e scegli Elimina.

(Facoltativo) Elimina l'istanza EC2 e l'istanza DB create con CloudFormation

Se prima creavi AWS CloudFormation risorse, elimina lo CloudFormation stack dopo esserti connesso ed esplorato l'istanza EC2 e l'istanza DB di esempio, in modo che non ti vengano più addebitati costi.

Per eliminare le risorse CloudFormation

1. Apri la AWS CloudFormation console.
2. Nella pagina Stacks CloudFormationconsole, seleziona lo stack principale (lo stack senza il nome VPCStack o RDSNS). BastionStack
3. Scegli Elimina.
4. Seleziona Elimina stack quando viene richiesta la conferma.

Per ulteriori informazioni sull'eliminazione di uno stack in CloudFormation, consulta Eliminazione di uno stack sulla console nella Guida per l' AWS CloudFormationutente.AWS CloudFormation

(Facoltativo) Connessione dell'istanza database a una funzione Lambda

Puoi anche connettere la tua istanza database RDS per Oracle a una risorsa di elaborazione serverless Lambda. Le funzioni Lambda consentono di eseguire il codice senza il provisioning o la gestione dell'infrastruttura. Una funzione Lambda consente inoltre di rispondere automaticamente alle richieste di esecuzione del codice su qualsiasi scala, da una dozzina di eventi al giorno a centinaia al secondo. Per ulteriori informazioni, consulta Connessione automatica di una funzione Lambda e di un'istanza database.

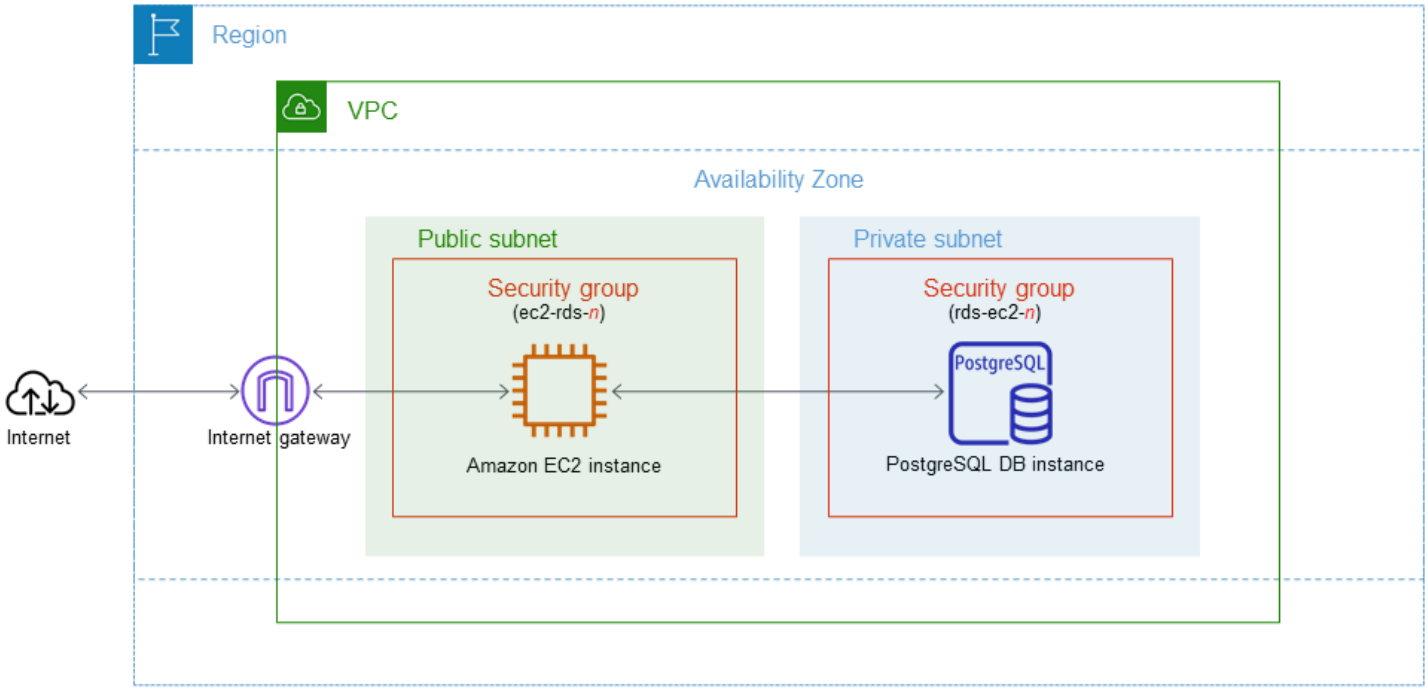
Creazione e connessione di un'istanza database PostgreSQL

Questo tutorial illustra come creare un'istanza EC2 e un'istanza database RDS per PostgreSQL. Il tutorial mostra come accedere all'istanza database dall'istanza EC2 utilizzando il client PostgreSQL standard. Come best practice, questo tutorial spiega come creare un'istanza database privata in un cloud privato virtuale (VPC). Nella maggior parte dei casi, le risorse presenti nello stesso VPC, come le istanze EC2, possono accedere all'istanza database, mentre le risorse esterne al VPC non possono accedervi.

Dopo aver completato il tutorial, è presente una sottorete pubblica e una privata in ogni zona di disponibilità del VPC. In una zona di disponibilità, l'istanza EC2 si trova nella sottorete pubblica mentre l'istanza database si trova nella sottorete privata.

> **Important**
>
> La creazione di un AWS account è gratuita. Tuttavia, completando questo tutorial, potresti incorrere in costi per le AWS risorse che utilizzi. È possibile eliminare queste risorse dopo aver completato l'esercitazione se non sono più necessarie.

Il seguente diagramma illustra la configurazione al completamento del tutorial.



Questo tutorial ti consente di creare le tue risorse utilizzando uno dei seguenti metodi:

1. Usa AWS Management Console - Fase 1: creazione di un'istanza EC2 e Fase 2: creazione di un'istanza database PostgreSQL
2. Utilizzare AWS CloudFormation per creare l'istanza del database e l'istanza EC2 - (Facoltativo) Crea VPC, istanza EC2 e istanza PostgreSQL utilizzando AWS CloudFormation

Il primo metodo utilizza Easy create per creare un'istanza database PostgreSQL privata con. AWS Management Console Qui, si specificano solo il tipo di motore DB, la dimensione dell'istanza DB e l'identificatore dell'istanza DB. Easy create (Creazione rapida) utilizza l'impostazione predefinita per altre opzioni di configurazione.

Se invece utilizzi Standard create, puoi specificare più opzioni di configurazione quando crei un'istanza DB. Queste opzioni includono impostazioni per la disponibilità, la sicurezza, i backup e la manutenzione. Per creare un'istanza database pubblica, è necessario utilizzare la Creazione standard. Per informazioni, consulta Creazione di un'istanza database Amazon RDS.

Argomenti

Prerequisiti

Prima di iniziare, completa le fasi descritte in questa sezione:

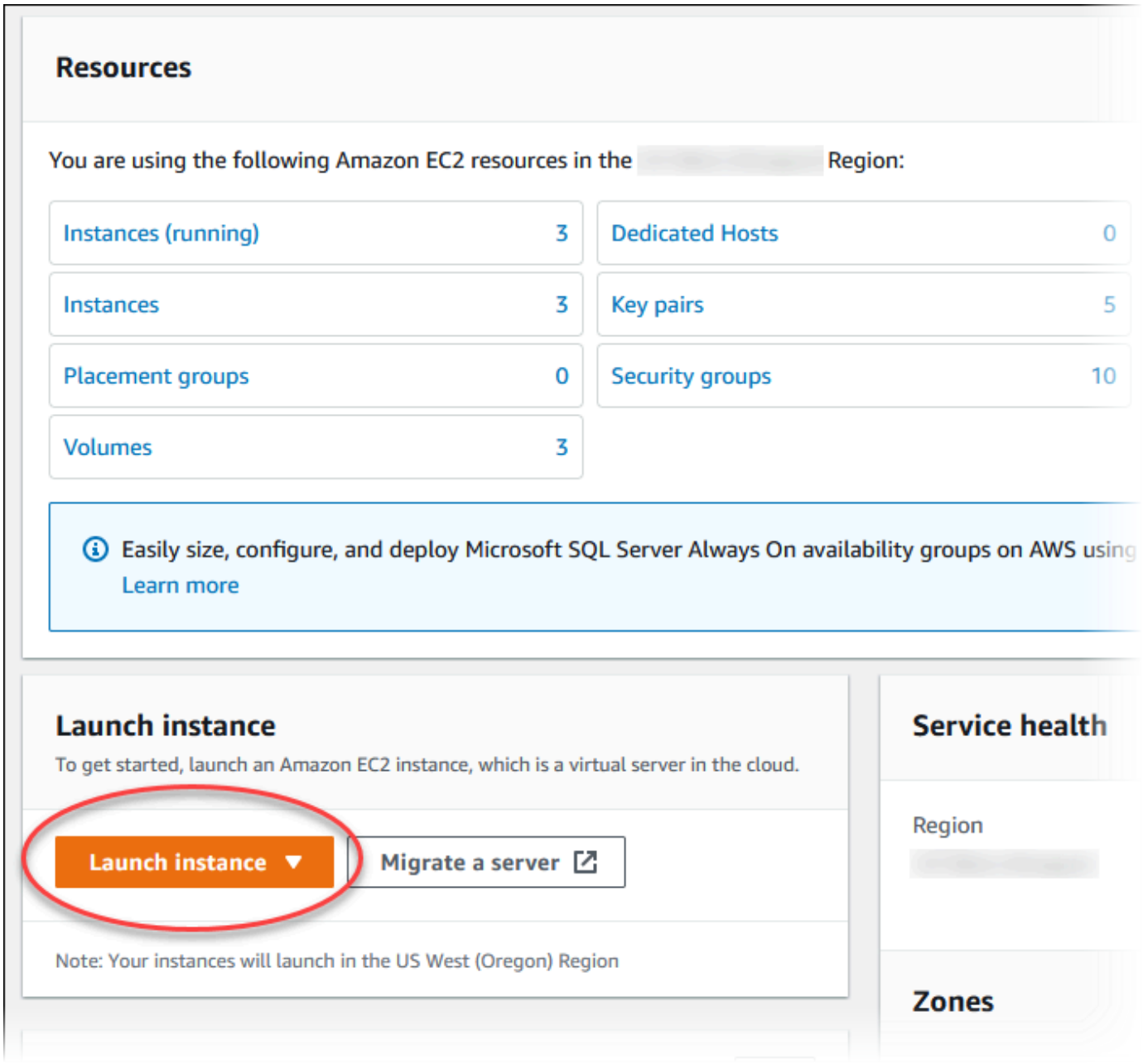
- Registrati per un Account AWS
- Crea un utente con accesso amministrativo

Fase 1: creazione di un'istanza EC2

Crea un'istanza Amazon EC2 da utilizzare per connetterti al database.

Per creare un'istanza EC2

1. Accedi AWS Management Console e apri la console Amazon EC2 all'indirizzo https://console.aws.amazon.com/ec2/.
2. Nell'angolo in alto a destra di AWS Management Console, scegli l'istanza EC2 Regione AWS in cui desideri creare l'istanza EC2.
3. Seleziona Pannello di controllo EC2, quindi Avvia istanza, come visualizzato di seguito.

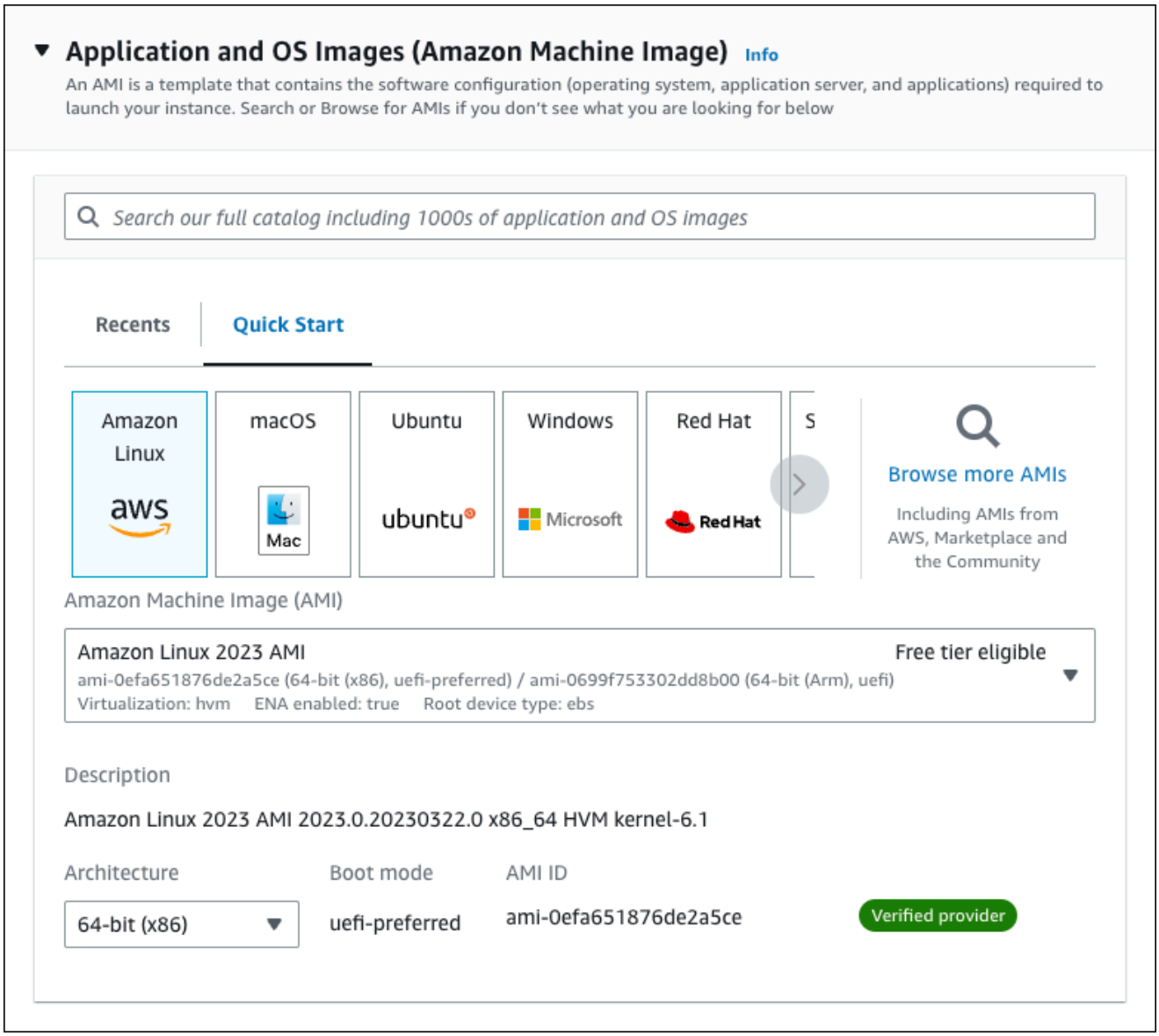


Viene visualizzata la pagina Avvia un'istanza.

4. Scegli le seguenti impostazioni nella pagina Avvia un'istanza.

   a. Nell'area Name and tags (Nome e tag), in Name (Nome) inserisci **`ec2-database-connect`**.

   b. In Immagini applicazione e sistema operativo (Amazon Machine Image), scegli Amazon Linux, quindi AMI Amazon Linux 2023. Mantieni le selezioni predefinite per le altre opzioni.



   c. In Instance type (Tipo di istanza), scegli t2.micro.

   d. In Key pair (login) (Coppia di chiavi (login), per Key pair name (Nome della coppia di chiavi), scegli una coppia di chiavi esistente. Per creare una nuova coppia di chiavi per l'istanza Amazon EC2, scegli Create new key pair (Crea nuova coppia di chiavi) e quindi utilizza la finestra Create key pair (Crea coppia di chiavi) per crearla.

Per ulteriori informazioni sulla creazione di una nuova coppia di chiavi, consulta Creazione di una coppia di chiavi nella Guida per l'utente di Amazon EC2 per istanze Linux.

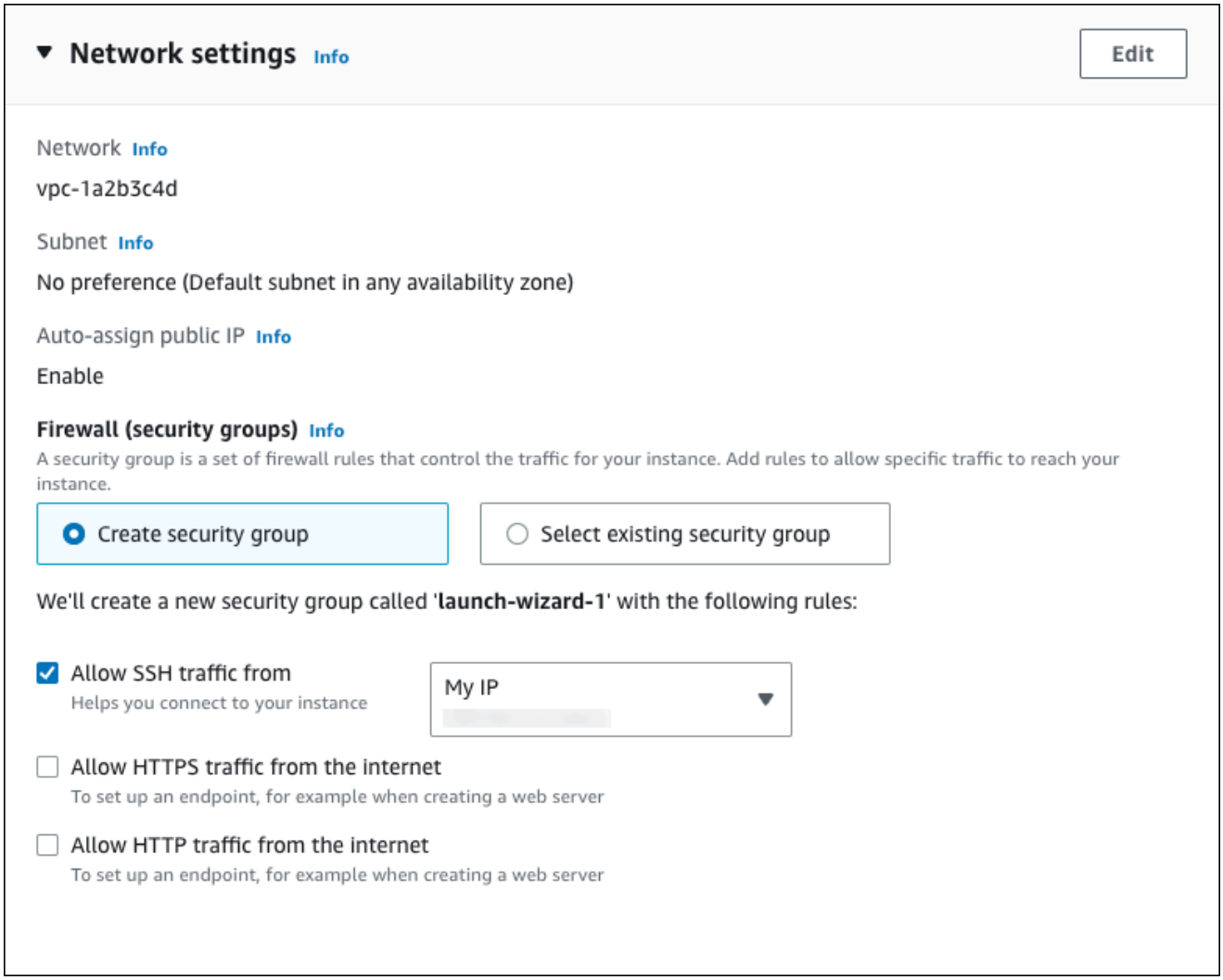
e. In Consenti traffico SSH, nell'area Impostazioni di rete scegliere l'origine delle connessioni SSH all'istanza EC2.

   È possibile scegliere My IP (Il mio IP) se l'indirizzo IP visualizzato è corretto per le connessioni SSH. In caso contrario, è possibile determinare l'indirizzo IP da utilizzare per connettersi alle istanze EC2 nel VPC utilizzando Secure Shell (SSH). Per determinare l'indirizzo IP pubblico, in una finestra o una scheda del browser diversa, è possibile utilizzare il servizio all'indirizzo https://checkip.amazonaws.com. Un esempio di indirizzo IP è 192.0.2.1/32.

   In molti casi, è possibile eseguire la connessione tramite un fornitore di servizi Internet (ISP) o con la protezione di un firewall senza un indirizzo IP statico. In tal caso, accertati di determinare l'intervallo di indirizzi IP utilizzati dai computer client.

   > **Warning**
   >
   > Se utilizzi `0.0.0.0/0` per l'accesso SSH, consenti a tutti gli indirizzi IP di accedere alle istanze EC2 pubbliche utilizzando SSH. Questo approccio è accettabile per un breve periodo di tempo in un ambiente di test, ma non è sicuro per gli ambienti di produzione. In produzione, autorizza solo un determinato indirizzo IP o un intervallo di indirizzi per accedere alle istanze EC2 utilizzando SSH.

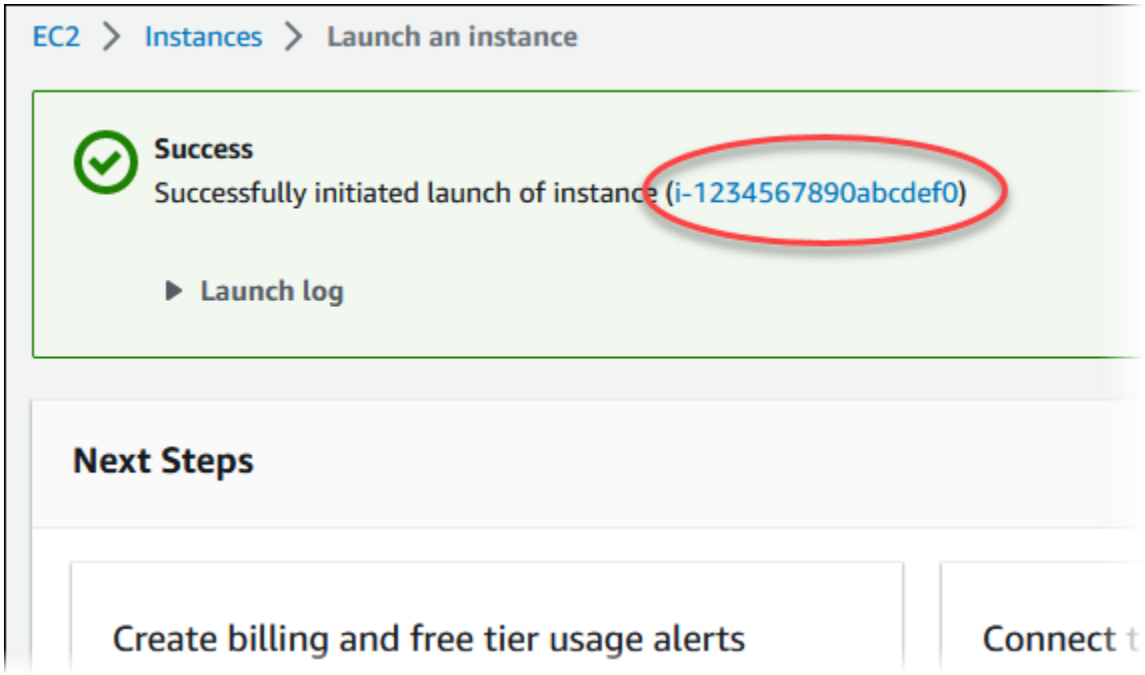
   L'immagine seguente mostra un esempio della sezione Impostazioni di rete.



f. Lascia i valori predefiniti per le sezioni rimanenti.

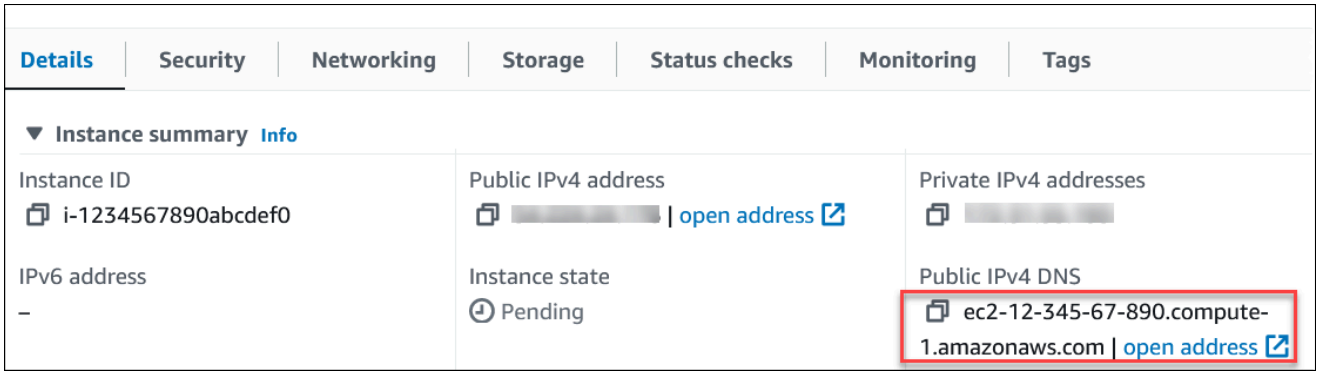
g. Analizza un riepilogo della configurazione dell'istanza EC2 nel pannello Riepilogo e, quando è tutto pronto, scegli Avvia istanza.

5. Nella pagina Stato avvio prendi nota dell'identificatore per la nuova istanza EC2, ad esempio: `i-1234567890abcdef0`.

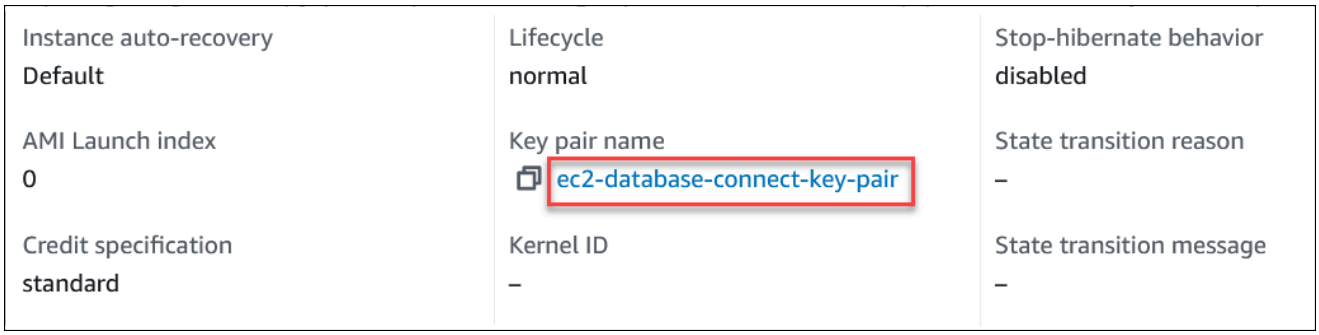


6. Scegli l'identificatore dell'istanza EC2 per aprire l'elenco delle istanze EC2, quindi seleziona l'istanza EC2.
7. Nella scheda Dettagli, annota i seguenti valori, necessari quando ti connetti tramite SSH:

   a. In Riepilogo istanza, annota il valore visualizzato in DNS IPv4 pubblico.



   b. In Dettagli istanza, annota il valore visualizzato in Nome coppia di chiavi.



8. Attendi che Stato dell'istanza diventi In esecuzione per l'istanza EC2 prima di continuare.

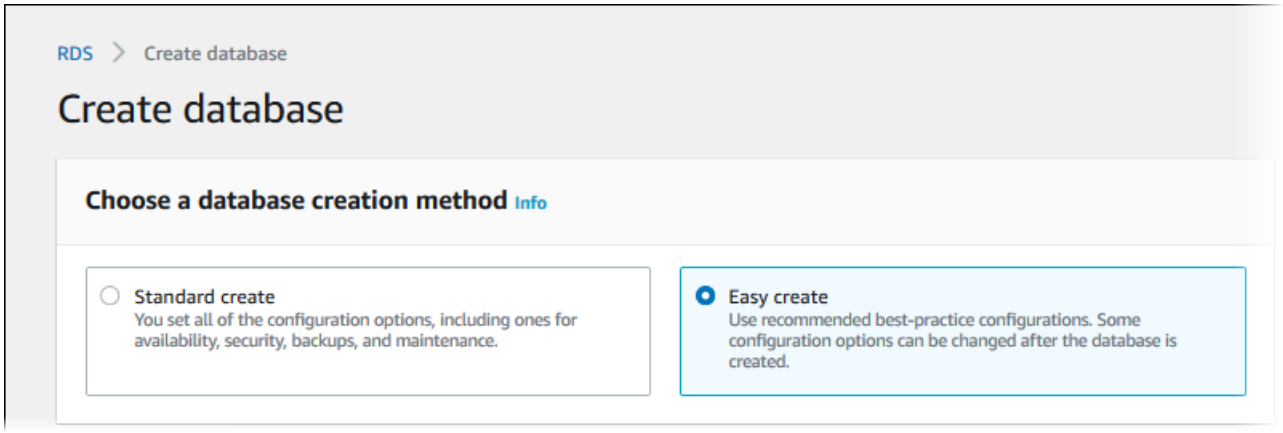
Fase 2: creazione di un'istanza database PostgreSQL

L'istanza database rappresenta l'elemento di base di Amazon RDS. Questo è l'ambiente dove esegui i tuoi database PostgreSQL.

In questo esempio, utilizzi la funzionalità Creazione semplice per creare un'istanza database che esegue un motore di database PostgreSQL con una classe di istanza database db.t3.micro.

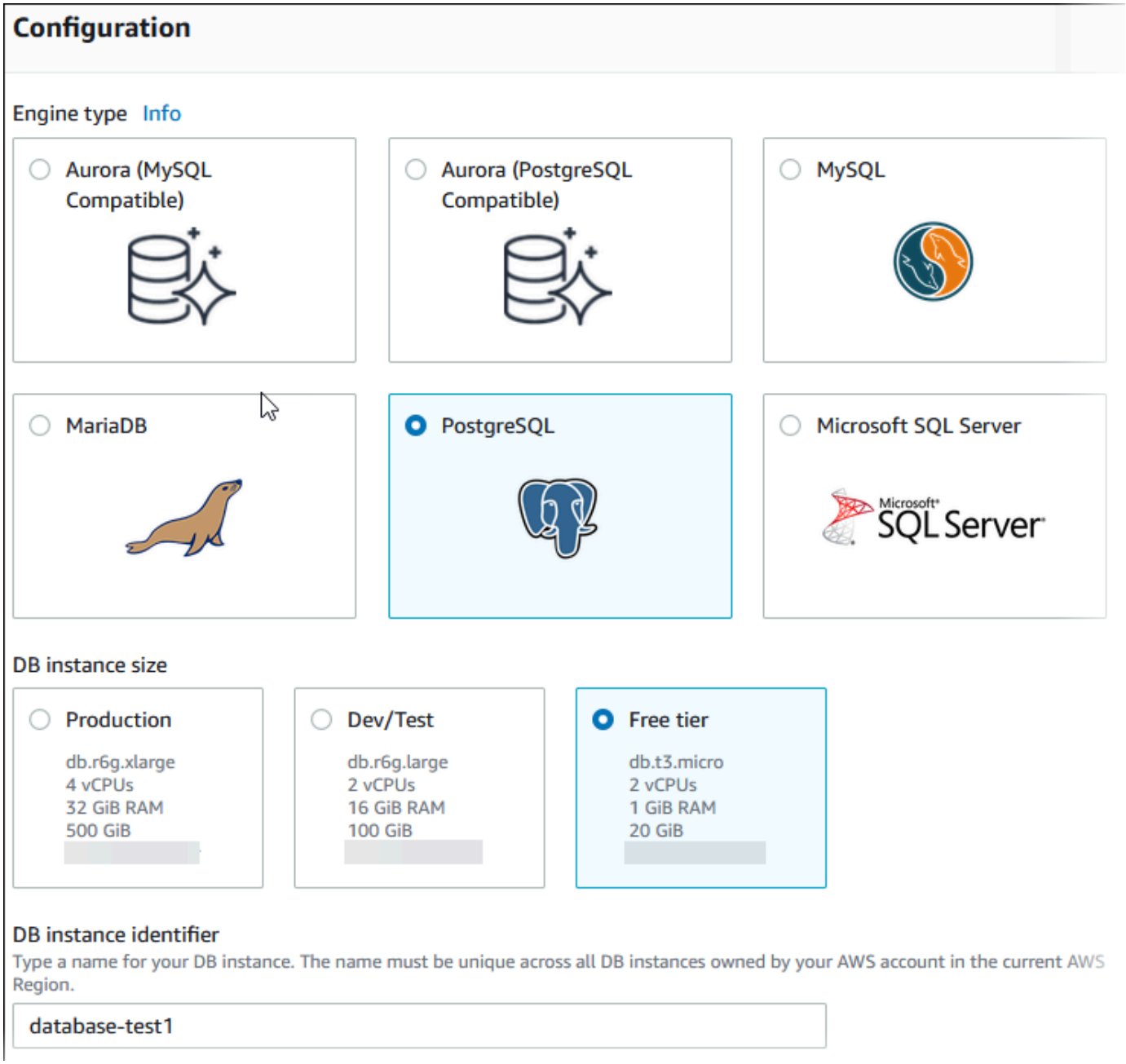
Per creare un'istanza database PostgreSQL con Easy Create (Creazione rapida)

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nell'angolo in alto a destra della console Amazon RDS, scegli la AWS regione in cui desideri creare l'istanza DB.
3. Nel riquadro di navigazione, scegliere Databases (Database).
4. Scegliere Create database (Crea database) e verificare che l'opzione Easy Create (Creazione rapida) sia selezionata.



5. In Configuration (Configurazione), seleziona PostgreSQL.
6. Per DB instance size (Dimensione istanza database), seleziona Free tier (Piano gratuito).
7. Per l'identificatore dell'istanza DB, inserisci **`database-test1`**.
8. In Nome utente master, inserisci un nome dell'utente master o lascia invariato il nome predefinito (**`postgres`**).

   La pagina Create database (Crea database) la pagina dovrebbe apparire simile alla seguente immagine.

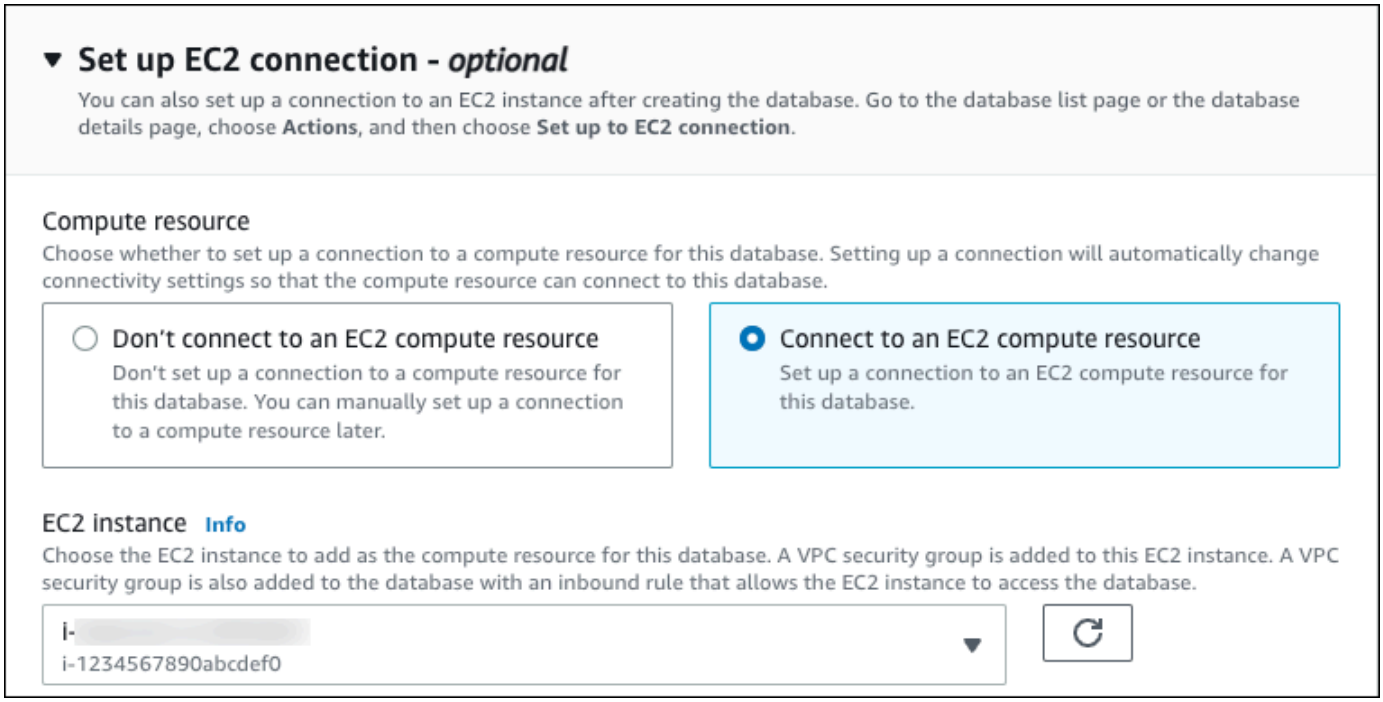


9. Per utilizzare una password master generata automaticamente per l'istanza database, seleziona Genera automaticamente una password.

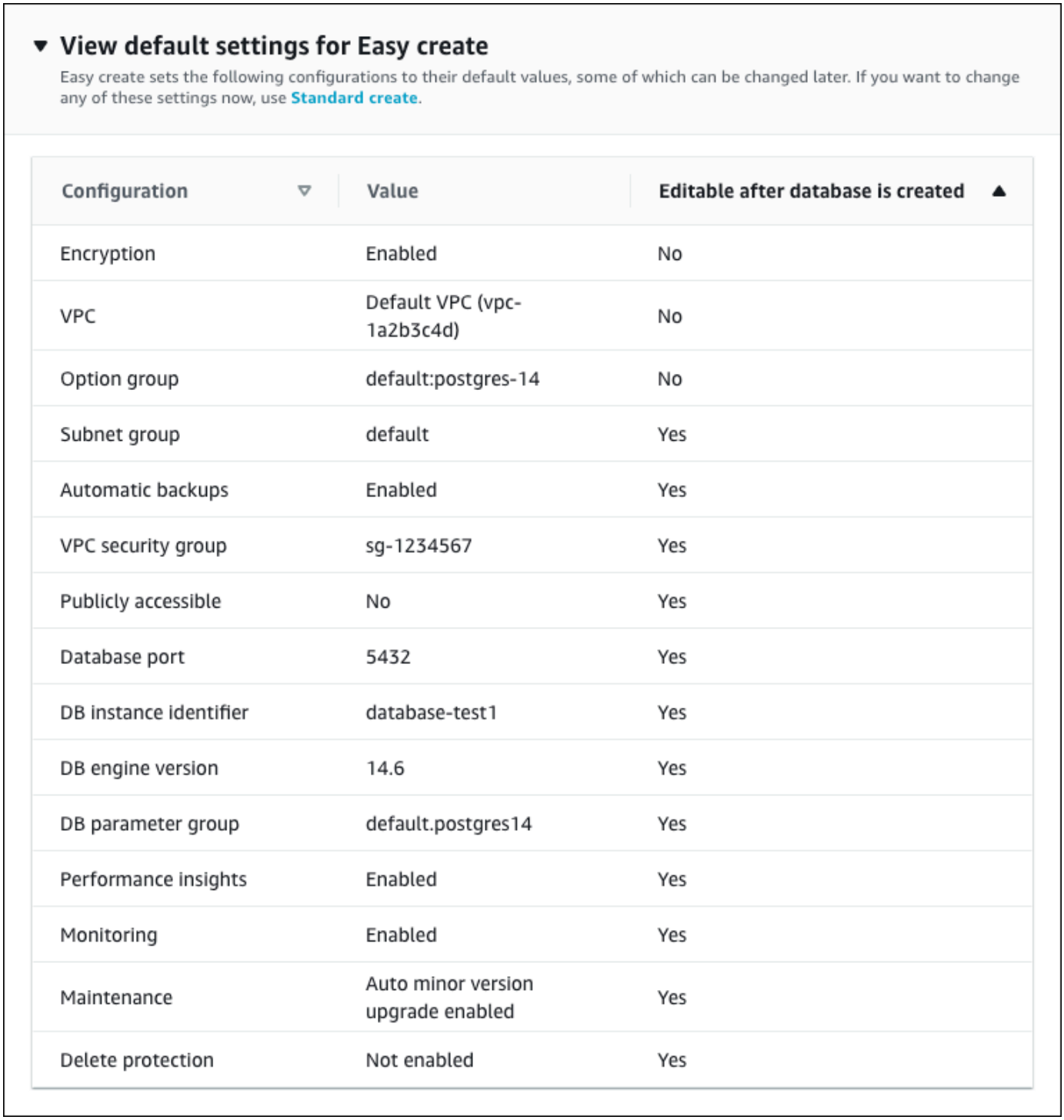
   Per inserire la password master, deseleziona la casella Genera automaticamente una password e inserisci la stessa password in Password master e Conferma password.

10. Per configurare una connessione con l'istanza EC2 creata in precedenza, apri Configura connessione EC2 - opzionale.

    Seleziona Connetti a una risorsa di calcolo EC2. Scegli l'istanza EC2 creata in precedenza.



11. Apri Visualizza le impostazioni predefinite per la creazione Semplice.



View default settings for Easy create

Easy create sets the following configurations to their default values, some of which can be changed later. If you want to change any of these settings now, use **Standard create**.

Configuration	Value	Editable after database is created
Encryption	Enabled	No
VPC	Default VPC (vpc-1a2b3c4d)	No
Option group	default:postgres-14	No
Subnet group	default	Yes
Automatic backups	Enabled	Yes
VPC security group	sg-1234567	Yes
Publicly accessible	No	Yes
Database port	5432	Yes
DB instance identifier	database-test1	Yes
DB engine version	14.6	Yes
DB parameter group	default.postgres14	Yes
Performance insights	Enabled	Yes
Monitoring	Enabled	Yes
Maintenance	Auto minor version upgrade enabled	Yes
Delete protection	Not enabled	Yes

Puoi esaminare le impostazioni predefinite utilizzate con Easy create (Creazione rapida). La colonna Modificabile dopo la creazione del database mostra le opzioni che puoi modificare dopo aver creato il database.

- Se un'impostazione contiene No in quella colonna e desideri cambiarla, puoi utilizzare la Creazione standard per creare l'istanza database.
- Se un'impostazione contiene Sì in quella colonna e desideri cambiarla, puoi utilizzare la Creazione standard per creare l'istanza database o modificare l'istanza database dopo averla creata per cambiare l'impostazione.

12. Scegliere Crea database.

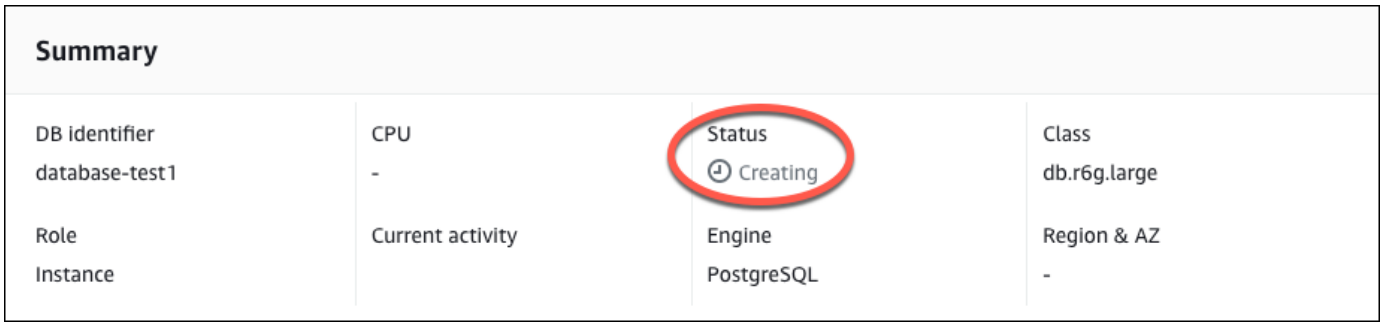
    Per vedere nome utente e password per l'istanza database, seleziona View credential details (Vedi dettagli delle credenziali).

    Per connetterti all'istanza database come utente principale, utilizza il nome utente e la password visualizzati.

    > **Important**
    >
    > Non potrai visualizzare di nuovo la password dell'utente principale. Se non la registri, potresti doverla modificare.
    > Se devi modificare la password dell'utente principale dopo che l'istanza database è disponibile, puoi modificare l'istanza database per eseguire tale operazione. Per ulteriori informazioni sulla modifica di un'istanza database , consulta Modifica di un'istanza database Amazon RDS.

13. Nell'elenco Database seleziona il nome della nuova istanza database PostgreSQL per visualizzarne i dettagli.

    L'istanza database ha lo stato Creazione in corso fino a quando non è pronta per essere utilizzata.



Quando lo stato cambia in Available (Disponibile), puoi connettersi all'istanza database. A seconda della classe di istanza database e della quantità di storage, prima che la nuova istanza sia disponibile possono trascorrere fino a 20 minuti.

(Facoltativo) Crea VPC, istanza EC2 e istanza PostgreSQL utilizzando AWS CloudFormation

Invece di usare la console per creare il tuo VPC, l'istanza EC2 e l'istanza PostgreSQL, puoi AWS CloudFormation usarla per fornire risorse trattando l'infrastruttura come codice. AWS Per aiutarti a organizzare AWS le tue risorse in unità più piccole e più gestibili, puoi utilizzare la funzionalità nested stack. AWS CloudFormation Per ulteriori informazioni, consulta Creare uno stack sulla AWS CloudFormation console e Lavorare con gli stack annidati.

> **Important**
>
> AWS CloudFormation è gratuito, ma le risorse che CloudFormation crea sono attive. Ti verranno addebitati i costi di utilizzo standard per queste risorse fino alla loro cessazione. L'addebito totale sarà minimo. Per informazioni su come ridurre al minimo gli addebiti, vai al AWS piano gratuito.

Per creare le tue risorse utilizzando la AWS CloudFormation console, completa i seguenti passaggi:

- Passaggio 1: scarica il CloudFormation modello
- Passaggio 2: configura le tue risorse utilizzando CloudFormation

Scarica il CloudFormation modello

Un CloudFormation modello è un file di testo JSON o YAML che contiene le informazioni di configurazione sulle risorse che desideri creare nello stack. Questo modello crea anche un VPC e un bastion host per te insieme all'istanza RDS.

Per scaricare il file modello, apri il seguente link, modello PostgreSQL CloudFormation .

Nella pagina Github, fai clic sul pulsante Scarica file raw per salvare il file YAML del modello.

Configura le tue risorse usando CloudFormation

> **Note**
>
> Prima di iniziare questo processo, assicurati di avere una coppia di chiavi per un'istanza EC2 nel tuo Account AWS. Per ulteriori informazioni, consulta Coppie di chiavi Amazon EC2 e istanze Linux.

Quando utilizzi il AWS CloudFormation modello, devi selezionare i parametri corretti per assicurarti che le risorse vengano create correttamente. Segui la procedura riportata di seguito:

1. Accedi AWS Management Console e apri la AWS CloudFormation console all'indirizzo https://console.aws.amazon.com/cloudformation.
2. Scegli Crea stack.
3. Nella sezione Specificare il modello, seleziona Carica un file modello dal computer, quindi scegli Avanti.
4. Nella pagina Specificare i dettagli dello stack, imposta i seguenti parametri:
   a. Imposta il nome dello stack su PostgreSQL TestStack.
   b. In Parametri, imposta le zone di disponibilità selezionando tre zone di disponibilità.
   c. Nella configurazione Linux Bastion Host, in Key Name, seleziona una coppia di chiavi per accedere alla tua istanza EC2.
   d. Nelle impostazioni di configurazione di Linux Bastion Host, imposta l'intervallo IP consentito sul tuo indirizzo IP. Per connetterti alle istanze EC2 nel tuo VPC utilizzando Secure Shell (SSH), determina il tuo indirizzo IP pubblico utilizzando il servizio all'indirizzo https://checkip.amazonaws.com. Un esempio di indirizzo IP è 192.0.2.1/32.

> **Warning**
>
> Se utilizzi `0.0.0.0/0` per l'accesso SSH, consenti a tutti gli indirizzi IP di accedere alle istanze EC2 pubbliche utilizzando SSH. Questo approccio è accettabile per un breve periodo di tempo in un ambiente di test, ma non è sicuro per gli ambienti di produzione. In produzione, autorizza solo un determinato indirizzo IP o un intervallo di indirizzi per accedere alle istanze EC2 utilizzando SSH.

e. Nella configurazione generale del database, imposta la classe dell'istanza del database su db.t3.micro.

f. Imposta il nome del database su. **`database-test1`**

g. Per Nome utente principale del database, inserisci un nome per l'utente principale.

h. Imposta la password utente principale di Manage DB con Secrets Manager su `false` per questo tutorial.

i. Per la password del database, imposta una password a tua scelta. Ricorda questa password per ulteriori passaggi del tutorial.

j. In Configurazione dell'archiviazione del database, imposta il tipo di archiviazione del database su gp2.

k. Nella configurazione Database Monitoring, imposta Enable RDS Performance Insights su false.

l. Lascia tutte le altre impostazioni come valori predefiniti. Fate clic su Avanti per continuare.

5. Nella pagina Configura le opzioni dello stack, lascia tutte le opzioni predefinite. Fai clic su Avanti per continuare.
6. Nella pagina Review stack, seleziona Invia dopo aver verificato le opzioni del database e dell'host Linux bastion.

Una volta completato il processo di creazione dello stack, visualizza gli stack con nomi BastionStacke RDSNS per annotare le informazioni necessarie per connetterti al database. Per ulteriori informazioni, vedere Visualizzazione dei dati e delle risorse AWS CloudFormation dello stack su. AWS Management Console

Fase 3: connessione a un'istanza database PostgreSQL

È possibile connettersi all'istanza database utilizzando pgadmin o psql. Questo esempio spiega come connettersi a un'istanza database PostgreSQL utilizzando il client della linea di comando psql.

Per connettersi a un'istanza database PostgreSQL tramite psql

1. Individuare l'endpoint (nome DNS) e il numero di porta per l'istanza database.

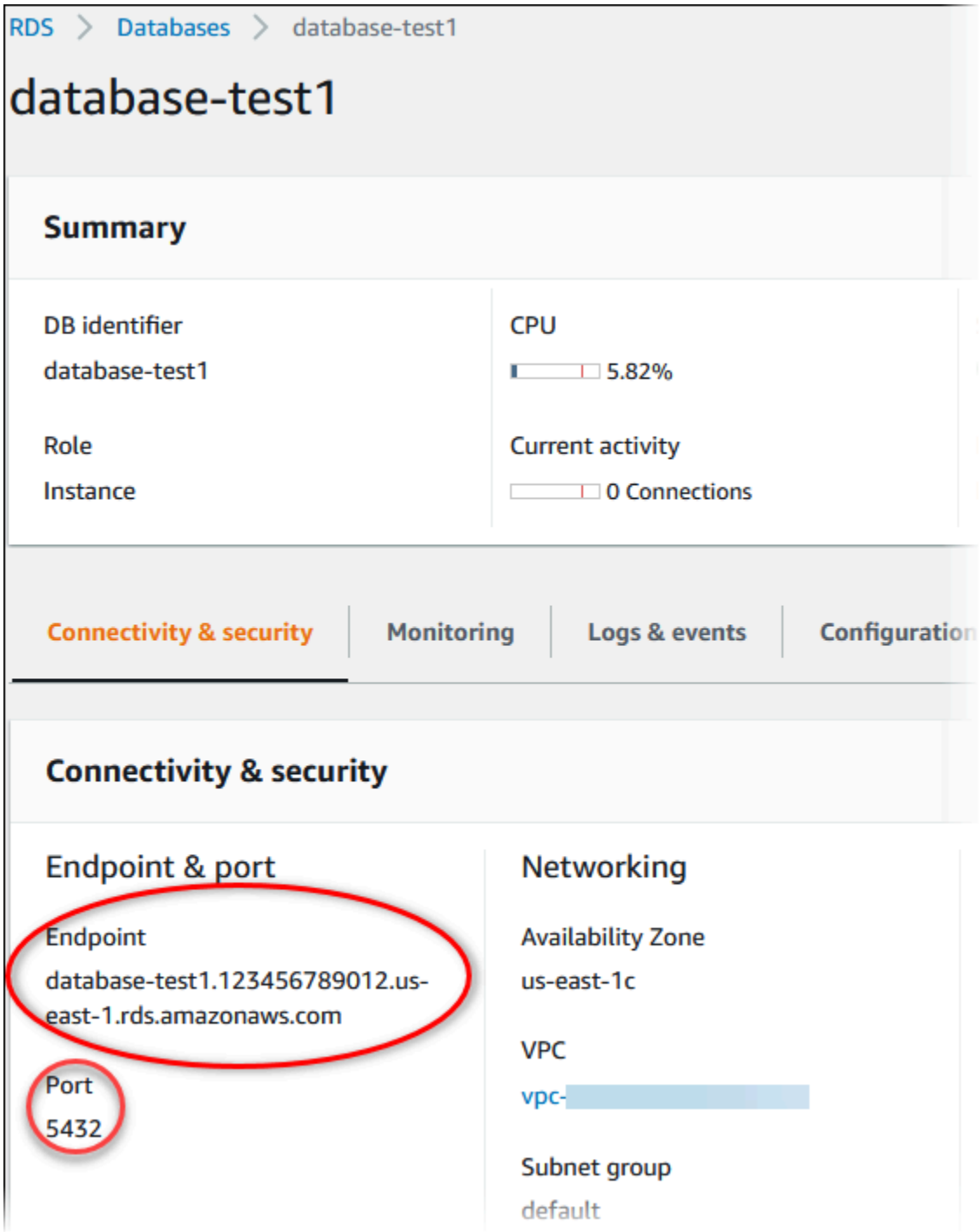
   a. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.

   b. Nell'angolo superiore destro della console Amazon RDS, scegli la Regione AWS dell'istanza database.

c. Nel riquadro di navigazione, scegli Databases (Database).

d. Scegliere il nome dell'istanza database PostgreSQL per visualizzarne i dettagli.

e. Nella scheda Connectivity & security (Connettività e sicurezza), copiare l'endpoint. Annotare anche il numero di porta. L'endpoint e il numero di porta sono necessari per la connessione all'istanza database.



2. Esegui la connessione all'istanza EC2 creata in precedenza seguendo la procedura riportata in Connessione all'istanza di Linux nella Guida per l'utente per istanze Linux di Amazon EC2.

   Ti consigliamo di connetterti all'istanza EC2 tramite SSH. Se l'utilità client SSH è installata su Windows, Linux o Mac, puoi connetterti all'istanza utilizzando il comando nel seguente formato:

   ```
   ssh -i location_of_pem_file ec2-user@ec2-instance-public-dns-name
   ```

   Ad esempio, supponi che `ec2-database-connect-key-pair.pem` sia archiviato in `/dir1` su Linux e che il DNS IPv4 pubblico per l'istanza EC2 sia `ec2-12-345-678-90.compute-1.amazonaws.com`. Il comando SSH sarà simile al seguente:

   ```
   ssh -i /dir1/ec2-database-connect-key-pair.pem ec2-
   user@ec2-12-345-678-90.compute-1.amazonaws.com
   ```

3. Ottieni le ultime correzioni di bug e gli aggiornamenti di sicurezza aggiornando il software sulla tua istanza EC2. A questo scopo, eseguire il comando seguente.

   > **Note**
   >
   > L'opzione `-y` installa gli aggiornamenti senza chiedere conferma. Per esaminare gli aggiornamenti prima di installarli, omettere questa opzione.

   ```
   sudo dnf update -y
   ```

4. Per installare il client della linea di comando psql da PostgreSQL su Amazon Linux 2023, esegui il comando seguente:

   ```
   sudo dnf install postgresql15
   ```

5. Stabilisci una connessione a un'istanza database PostgreSQL. Ad esempio, immetti il seguente comando in un prompt dei comandi in un computer client. Questa azione consente di connetterti all'istanza database PostgreSQL utilizzando il client psql.

   Sostituisci l'endpoint dell'istanza database (nome DNS) per *`endpoint`*, sostituisci il nome database `--dbname` a cui vuoi connetterti per *`postgres`* e sostituisci il nome utente master

utilizzato per *postgres*. Devi fornire la password master utilizzata quando viene richiesta una password.

```
psql --host=endpoint --port=5432 --dbname=postgres --username=postgres
```

Dopo aver immesso la password per l'utente, l'output dovrebbe essere analogo a quanto mostrato di seguito:

```
psql (14.3, server 14.6)
SSL connection (protocol: TLSv1.2, cipher: ECDHE-RSA-AES256-GCM-SHA384, bits: 256,
 compression: off)
Type "help" for help.

postgres=>
```

Per ulteriori informazioni sulla connessione a un'istanza database PostgreSQL, consulta Connessione a un'istanza database che esegua il motore di database di PostgreSQL. In caso di mancata connessione all'istanza database, consulta Risoluzione dei problemi relativi alle connessioni all'istanza RDS per PostgreSQL.

Per motivi di sicurezza, la best practice è utilizzare connessioni crittografate. Utilizza una connessione PostgreSQL non crittografata solo quando il client e il server sono nello stesso VPC e la rete è attendibile. Per ulteriori informazioni sull'uso di connessioni crittografate, consulta Connessione a un'istanza database PostgreSQL tramite SSL.

6. Eseguire comandi SQL.

   Ad esempio, il seguente comando SQL mostra la data e l'ora correnti:

   ```
   SELECT CURRENT_TIMESTAMP;
   ```

Fase 4: eliminazione dell'istanza EC2 e dell'istanza database

Dopo la connessione e l'esplorazione dell'istanza EC2 e dell'istanza database di esempio che hai creato, eliminale per evitare di ricevere l'addebito dei relativi costi.

Se in passato AWS CloudFormation creavi risorse, salta questo passaggio e vai al passaggio successivo.

Per eliminare l'istanza EC2

1. Accedi AWS Management Console e apri la console Amazon EC2 all'indirizzo https://console.aws.amazon.com/ec2/.
2. Nel riquadro di navigazione, seleziona Istanze.
3. Seleziona l'istanza EC2 e scegli Stato istanza, Termina istanza.
4. Quando viene richiesta la conferma, seleziona Terminate (Interrompi).

Per ulteriori informazioni sull'eliminazione di un'istanza EC2, consulta Interruzione di un'istanza nella Guida per l'utente per istanze Linux di Amazon EC2.

Per eliminare un'istanza database senza snapshot DB finale

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, scegliere Databases (Database).
3. Scegliere l'istanza database da eliminare.
4. In Actions (Azioni), selezionare Delete (Elimina).
5. Deseleziona Creare uno snapshot finale? e Conserva backup automatizzati.
6. Completa la conferma e scegli Elimina.

(Facoltativo) Elimina l'istanza EC2 e l'istanza DB create con CloudFormation

Se prima creavi AWS CloudFormation risorse, elimina lo CloudFormation stack dopo esserti connesso ed esplorato l'istanza EC2 e l'istanza DB di esempio, in modo che non ti vengano più addebitati costi.

Per eliminare le risorse CloudFormation

1. Apri la AWS CloudFormation console.
2. Nella pagina Stacks CloudFormationconsole, seleziona lo stack principale (lo stack senza il nome VPCStack o RDSNS). BastionStack
3. Scegli Elimina.
4. Seleziona Elimina stack quando viene richiesta la conferma.

Per ulteriori informazioni sull'eliminazione di uno stack in CloudFormation, consulta Eliminazione di uno stack sulla console nella Guida per l' AWS CloudFormationutente.AWS CloudFormation

(Facoltativo) Connessione dell'istanza database a una funzione Lambda

Puoi anche connettere la tua istanza database RDS per PostgreSQL a una risorsa di elaborazione serverless Lambda. Le funzioni Lambda consentono di eseguire il codice senza il provisioning o la gestione dell'infrastruttura. Una funzione Lambda consente inoltre di rispondere automaticamente alle richieste di esecuzione del codice su qualsiasi scala, da una dozzina di eventi al giorno a centinaia al secondo. Per ulteriori informazioni, consulta Connessione automatica di una funzione Lambda e di un'istanza database.

Tutorial: creazione di un server Web e un'istanza database Amazon RDS

Questo tutorial descrive come installare un server Web Apache con PHP e creare un database MariaDB, MySQL o PostgreSQL. Il server Web viene eseguito in un'istanza Amazon EC2 utilizzando Amazon Linux 2023 e puoi scegliere tra un'istanza database MySQL o PostgreSQL. Sia l'istanza Amazon EC2 che l'istanza database vengono eseguite in un virtual private cloud (VPC) basato sul servizio Amazon VPC.

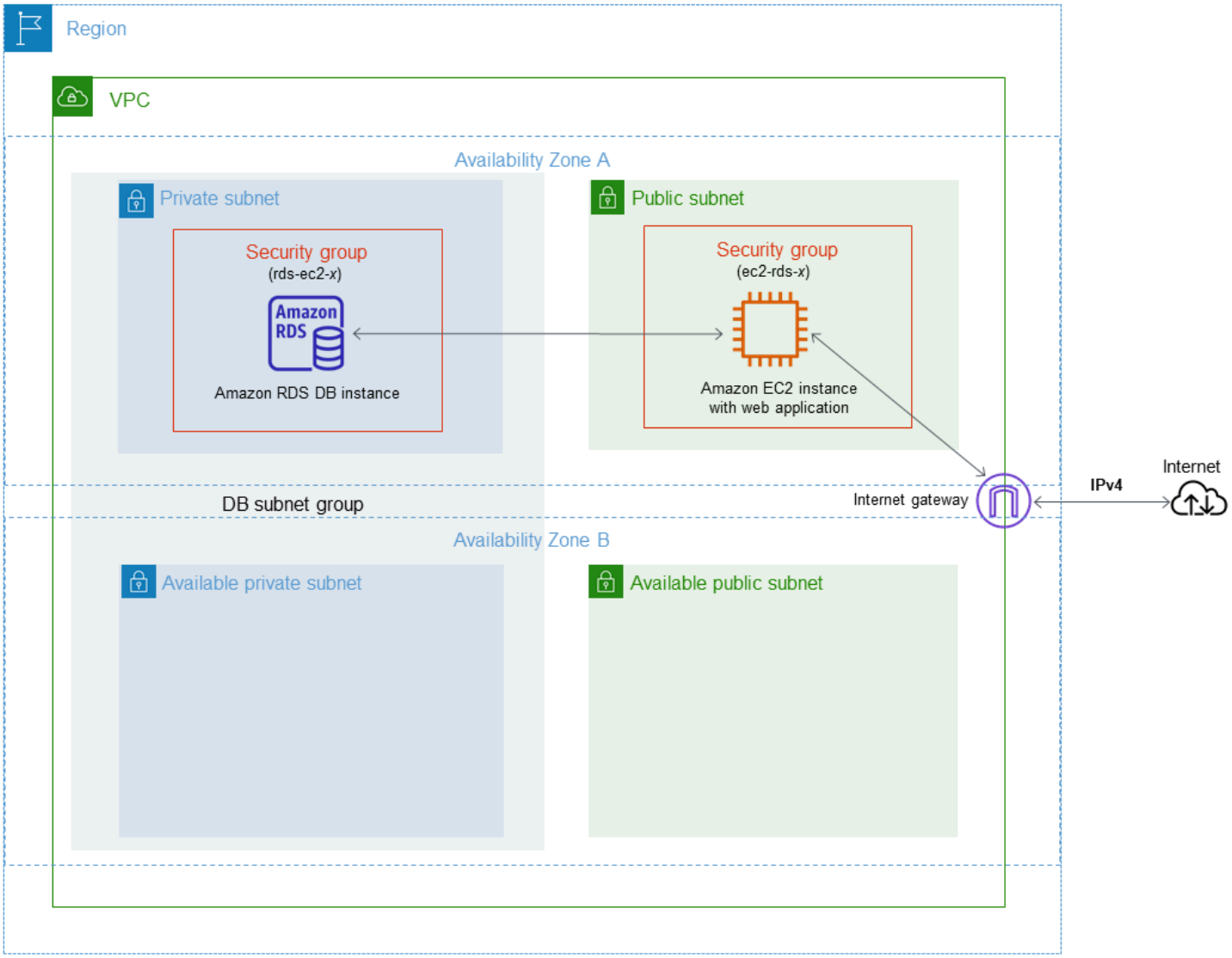
> **Important**
>
> Non vi sono costi aggiuntivi per la creazione di un account AWS. Tuttavia, completando l'esercitazione, potresti incorrere in costi per le risorse AWS utilizzate. È possibile eliminare queste risorse dopo aver completato l'esercitazione se non sono più necessarie.

> **Note**
>
> Questo tutorial funziona con Amazon Linux 2023 e potrebbe non funzionare per altre versioni di Linux.

Nel tutorial che segue, viene creata un'istanza EC2 che utilizza VPC, sottoreti e gruppo di sicurezza predefiniti per Account AWS. In questo tutorial viene illustrato come creare l'istanza database e configurare automaticamente la connettività con l'istanza EC2 creata. Nel tutorial viene quindi mostrato come installare il server Web sull'istanza EC2. Il server Web viene connesso all'istanza database nel VPC utilizzando l'endpoint dell'istanza del database.

1. Avvio di un'istanza EC2
2. Creazione di un'istanza database Amazon RDS
3. Installazione di un server Web nell'istanza EC2

Il seguente diagramma illustra la configurazione al completamento del tutorial.



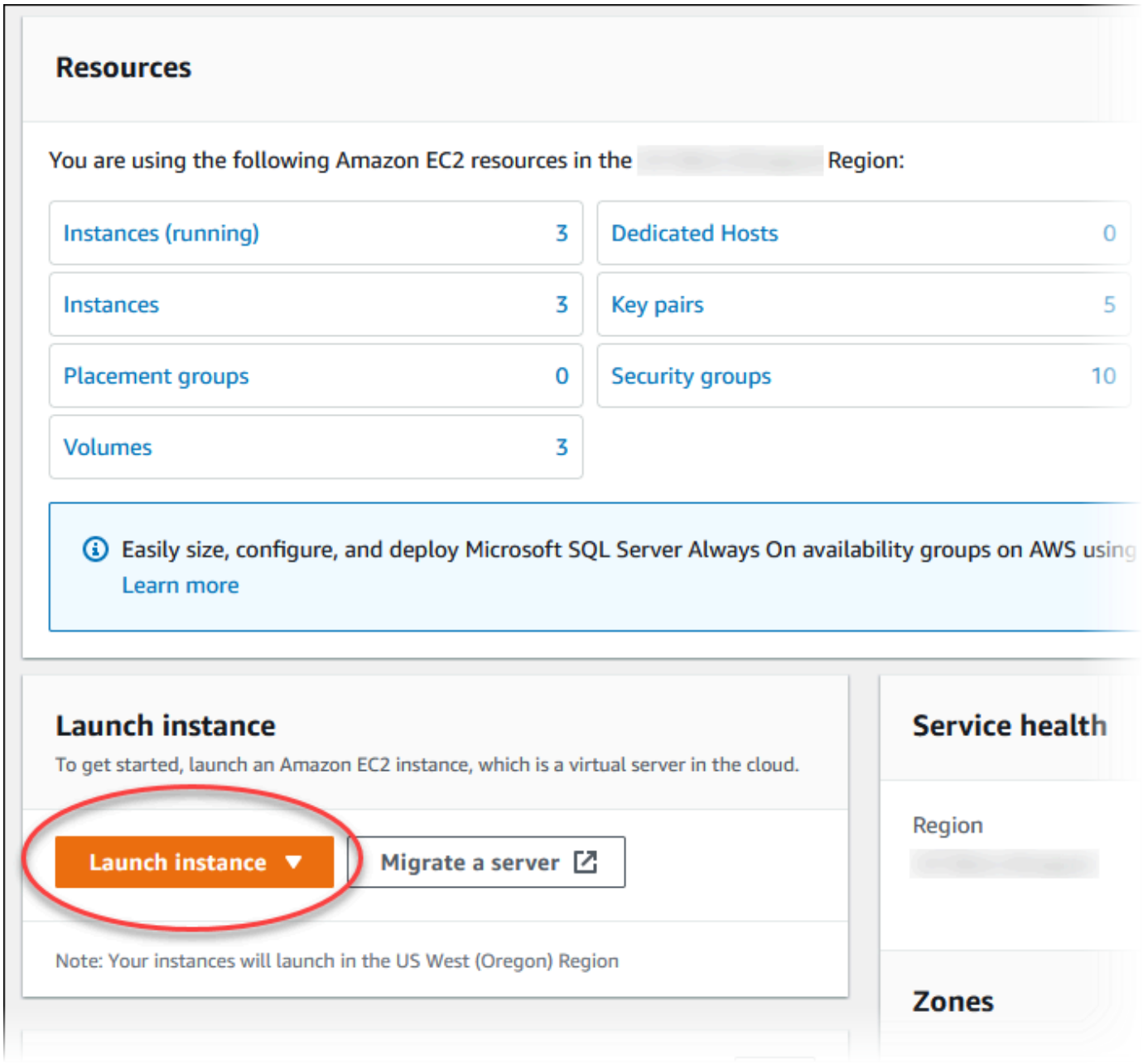
> **Note**
>
> Dopo aver completato il tutorial, è presente una sottorete pubblica e una privata in ogni zona di disponibilità del VPC. Questo tutorial utilizza il VPC predefinito per Account AWS e configura automaticamente la connettività tra l'istanza EC2 e l'istanza database. Se preferisci invece configurare un nuovo VPC per questo scenario, completa le attività in Tutorial: Creazione di un Amazon VPC da utilizzare con un'istanza database (solo IPv4).

Avvio di un'istanza EC2

Crea un'istanza Amazon EC2 nella sottorete pubblica del VPC.

Per avviare un'istanza EC2

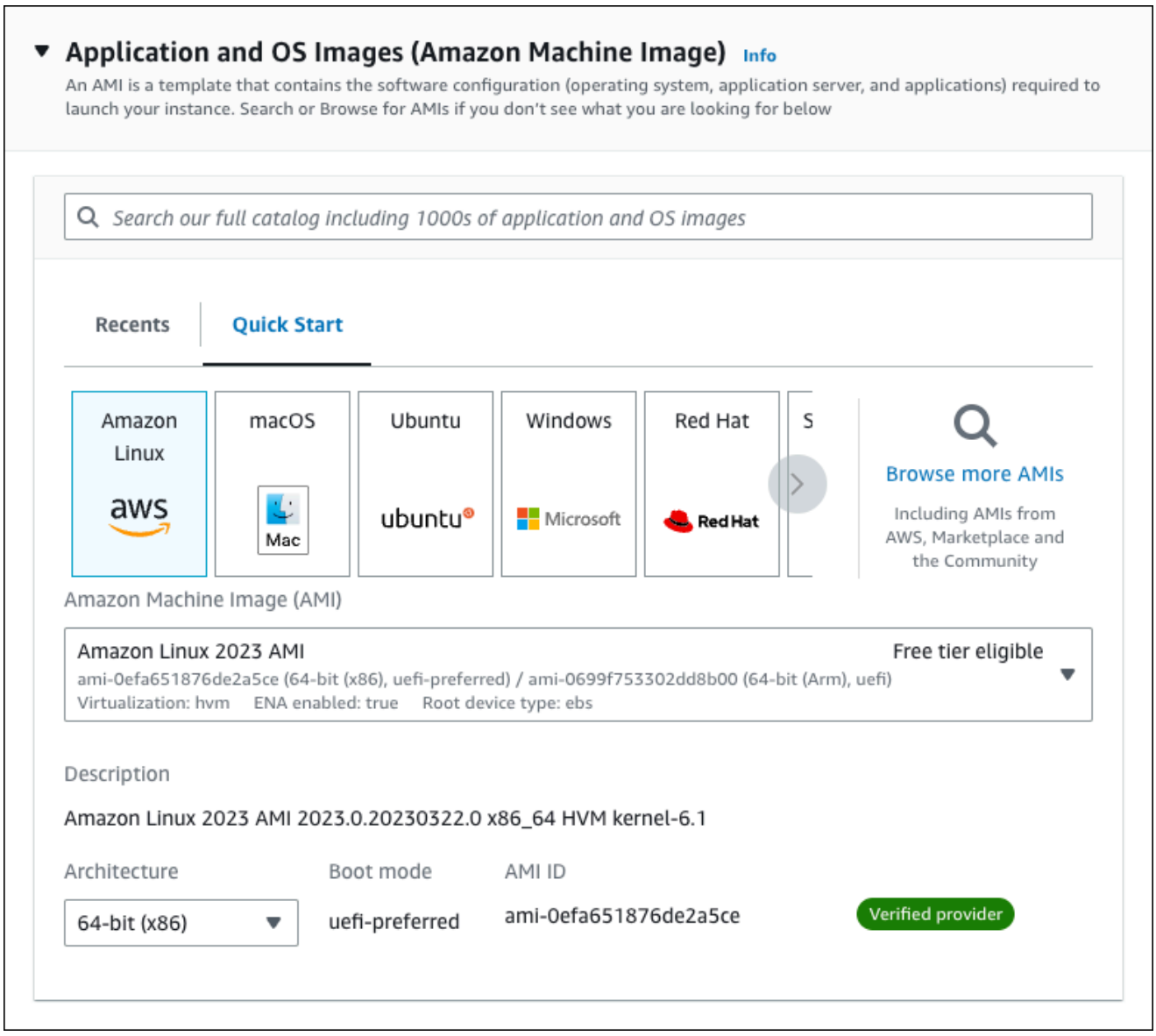
1. Accedi a AWS Management Console e apri la console Amazon EC2 all'indirizzo https://console.aws.amazon.com/ec2/.
2. Nell'angolo in alto a destra della AWS Management Console, scegli la Regione AWS in cui si desidera creare l'istanza EC2.
3. Selezionare Pannello di controllo EC2, quindi Avvia istanza, come visualizzato di seguito.



4. Scegli le seguenti impostazioni nella pagina Avvia un'istanza.

a. Nell'area Name and tags (Nome e tag), in Name (Nome) inserisci **`tutorial-ec2-instance-web-server`**.

b. In Immagini applicazione e sistema operativo (Amazon Machine Image), scegli Amazon Linux, quindi AMI Amazon Linux 2023. Manteni i valori predefiniti per le altre opzioni.



c. In Instance type (Tipo di istanza), scegli t2.micro.

d. In Key pair (login) (Coppia di chiavi (login), per Key pair name (Nome della coppia di chiavi), scegli una coppia di chiavi esistente. Per creare una nuova coppia di chiavi per l'istanza Amazon EC2, scegli Create new key pair (Crea nuova coppia di chiavi) e quindi utilizza la finestra Create key pair (Crea coppia di chiavi) per crearla.

Per ulteriori informazioni sulla creazione di una nuova coppia di chiavi, consulta Creazione di una coppia di chiavi nella Guida per l'utente di Amazon EC2 per istanze Linux.

e. In Network settings (Impostazioni di rete), imposta questi valori e conserva le impostazioni predefinite degli altri valori:

- In Allow SSH traffic from (Consenti traffico SSH da), scegli l'origine delle connessioni SSH all'istanza EC2.

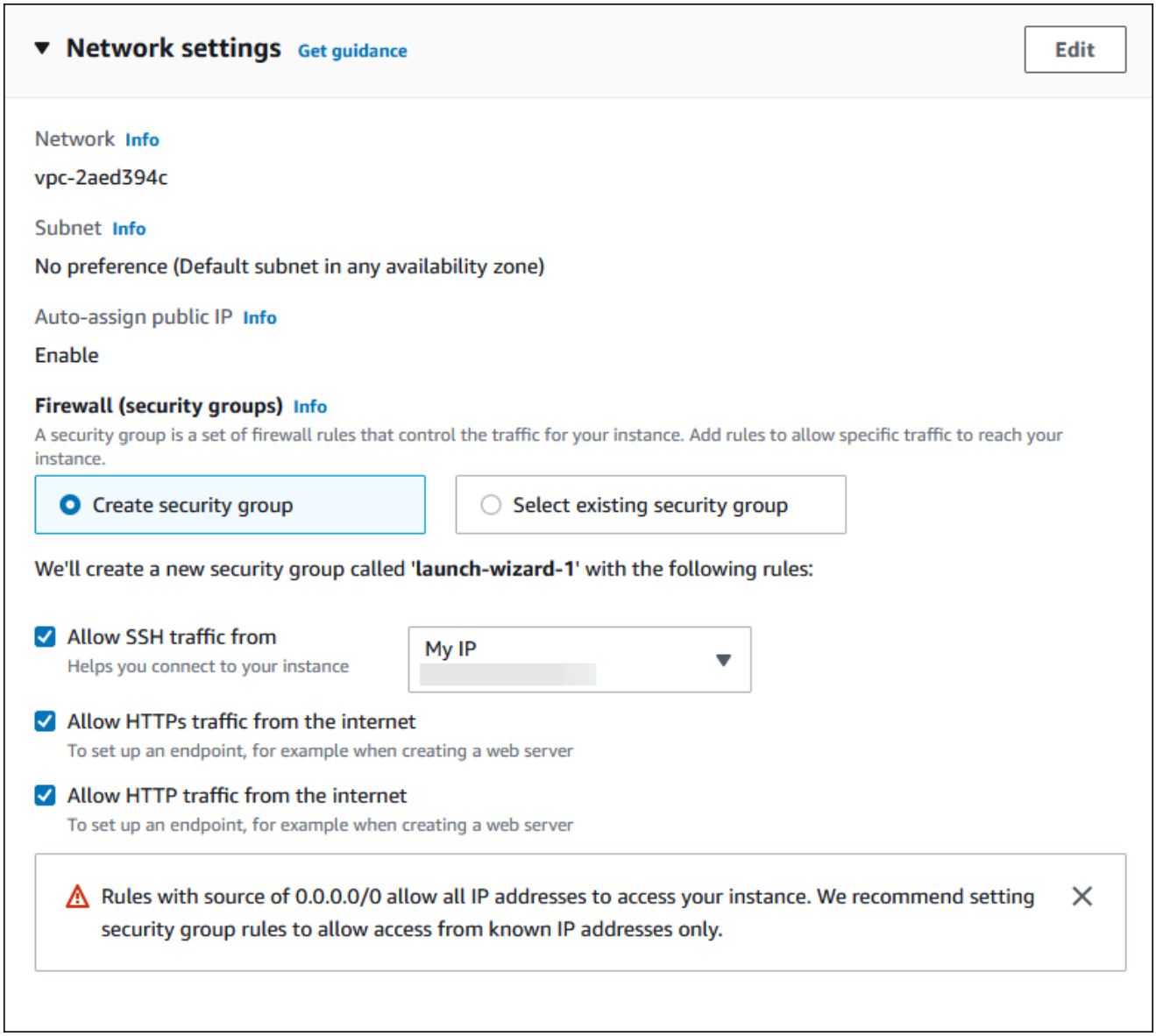
  È possibile scegliere My IP (Il mio IP) se l'indirizzo IP visualizzato è corretto per le connessioni SSH.

  In caso contrario, è possibile determinare l'indirizzo IP da utilizzare per connettersi alle istanze EC2 nel VPC utilizzando Secure Shell (SSH). Per determinare l'indirizzo IP pubblico, in una finestra o una scheda del browser diversa, è possibile utilizzare il servizio all'indirizzo https://checkip.amazonaws.com. Un esempio di indirizzo IP è `203.0.113.25/32`.

  In molti casi, è possibile eseguire la connessione tramite un fornitore di servizi Internet (ISP) o con la protezione di un firewall senza un indirizzo IP statico. In tal caso, accertati di determinare l'intervallo di indirizzi IP utilizzati dai computer client.

  > **Warning**
  >
  > Se utilizzi `0.0.0.0/0` per l'accesso SSH, consenti a tutti gli indirizzi IP di accedere alle istanze pubbliche utilizzando SSH. Questo approccio è accettabile per un breve periodo di tempo in un ambiente di test, ma non è sicuro per gli ambienti di produzione. In produzione, autorizza solo un determinato indirizzo IP o un intervallo di indirizzi per accedere alle istanze utilizzando SSH.

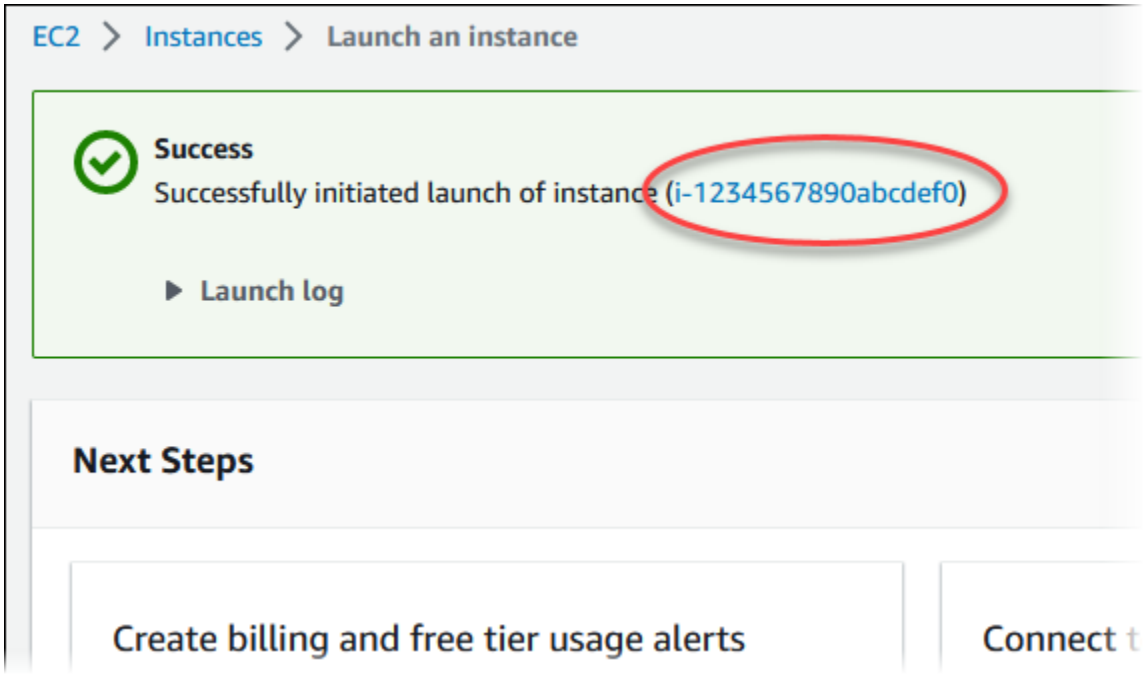
- Attiva Allow HTTPs traffic from the internet (Autorizzare il traffico HTTPS da Internet).
- Attiva Allow HTTP traffic from the internet (Autorizzare il traffico HTTP da Internet).



f. Lascia i valori predefiniti per le sezioni rimanenti.

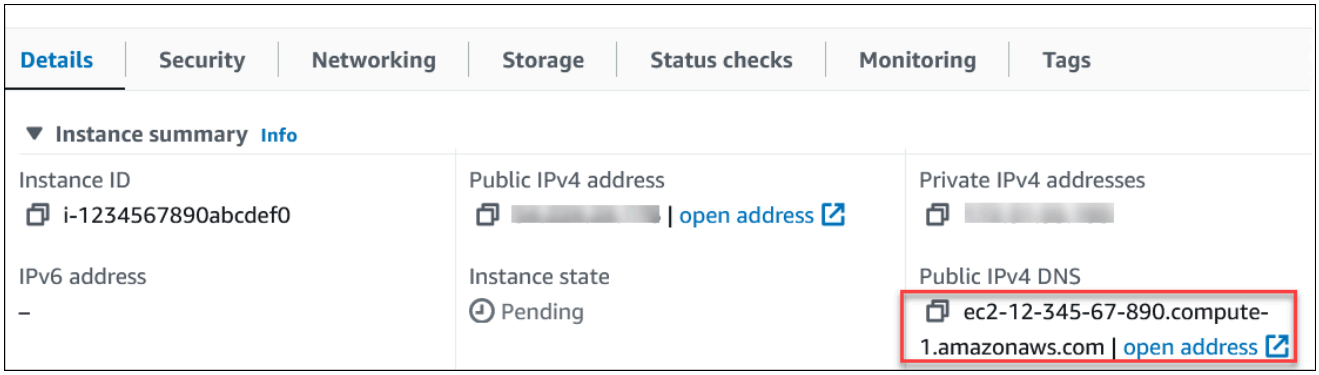
g. Analizza un riepilogo della configurazione dell'istanza nel pannello Summary (Riepilogo) e, quando è tutto pronto, scegli Launch instance (Avvia istanza).

5. Nella pagina Stato avvio prendi nota dell'identificatore per la nuova istanza EC2, ad esempio: `i-1234567890abcdef0`.

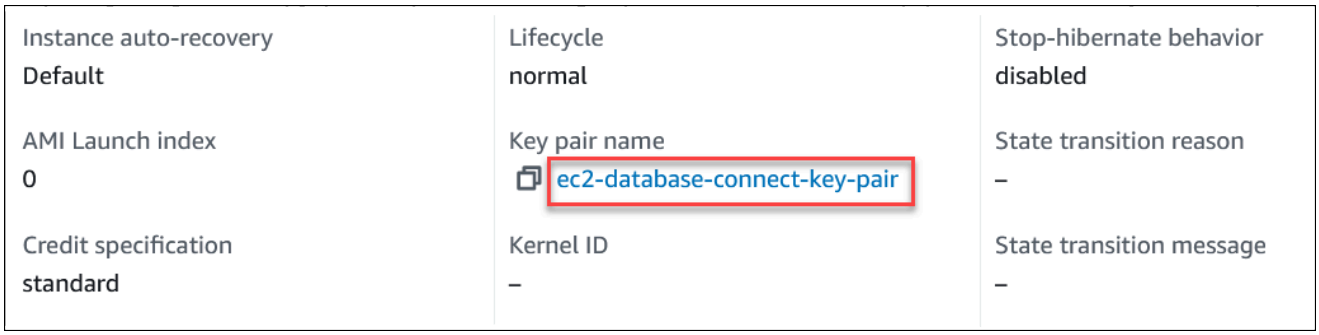


6. Scegli l'identificatore dell'istanza EC2 per aprire l'elenco delle istanze EC2, quindi seleziona l'istanza EC2.
7. Nella scheda Dettagli, annota i seguenti valori, necessari quando ti connetti tramite SSH:

   a. In Riepilogo istanza, annota il valore visualizzato in DNS IPv4 pubblico.



   b. In Dettagli istanza, annota il valore visualizzato in Nome coppia di chiavi.



8. Attendi che Instance state (Stato istanza) per l'istanza sia Running (In esecuzione) prima di continuare.
9. Completo Creazione di un'istanza database Amazon RDS.

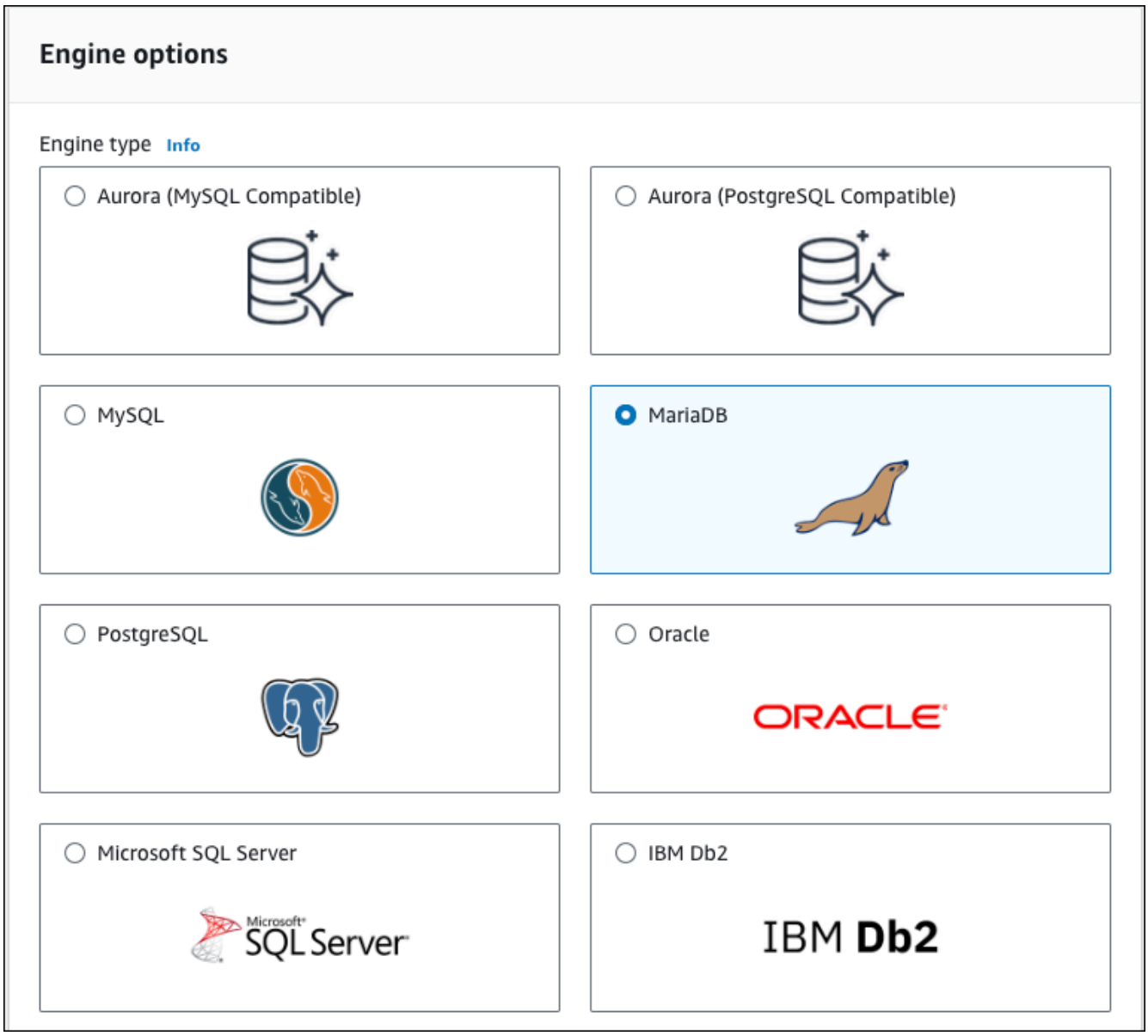
Creazione di un'istanza database Amazon RDS

Crea un'istanza database RDS per MariaDB, RDS per MySQL or RDS per PostgreSQL che mantiene i dati utilizzati da un'applicazione Web.

RDS for MariaDB

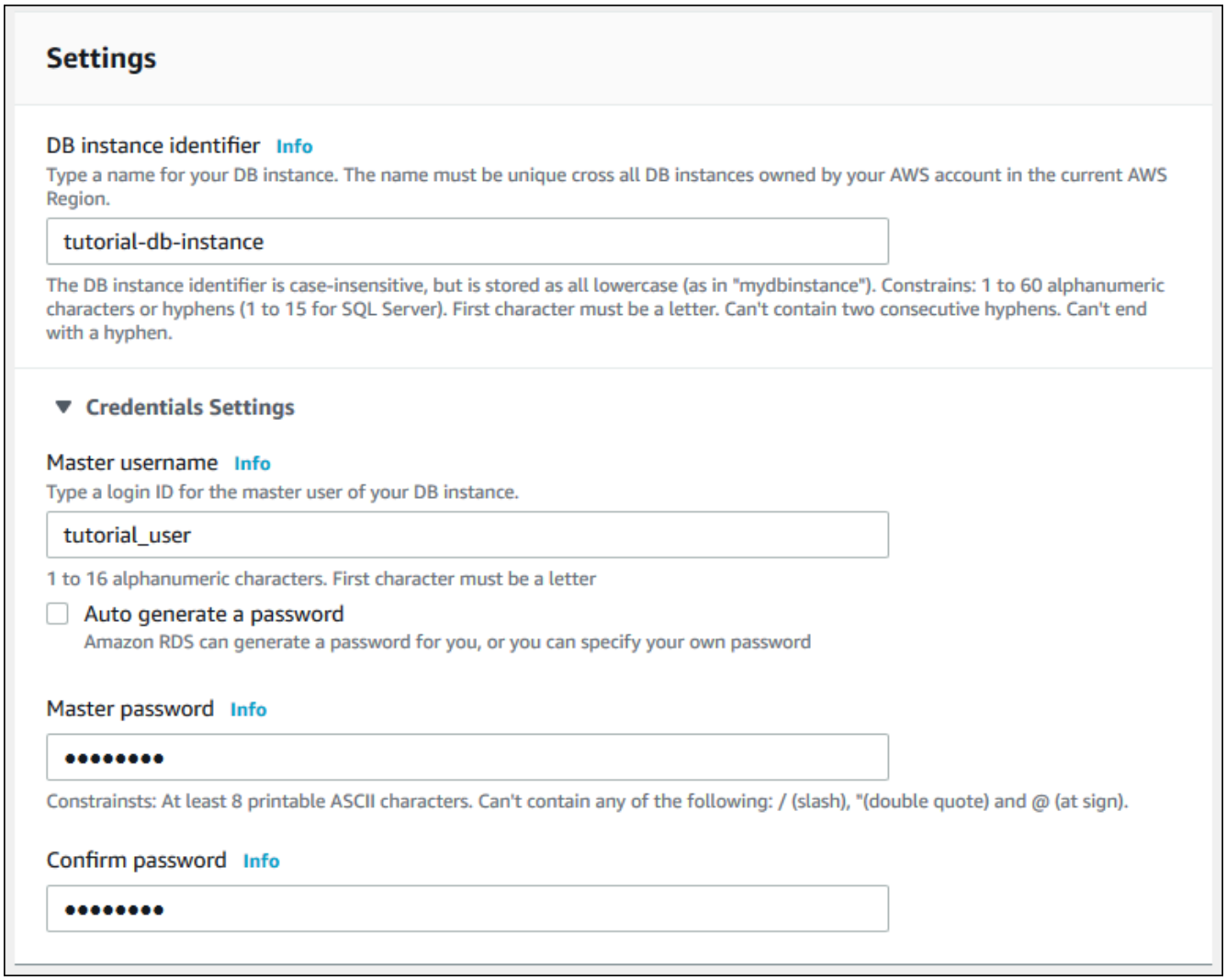
Creazione di un'istanza database MariaDB

1. Accedi alla AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nell'angolo in alto a destra della AWS Management Console, seleziona la Regione AWS. Deve essere identica a quella in cui hai creato l'istanza EC2.
3. Nel riquadro di navigazione, scegliere Databases (Database).
4. Scegliere Create database (Crea database).
5. Nella pagina Crea database scegli Creazione standard.
6. In Opzioni motore, seleziona MariaDB.



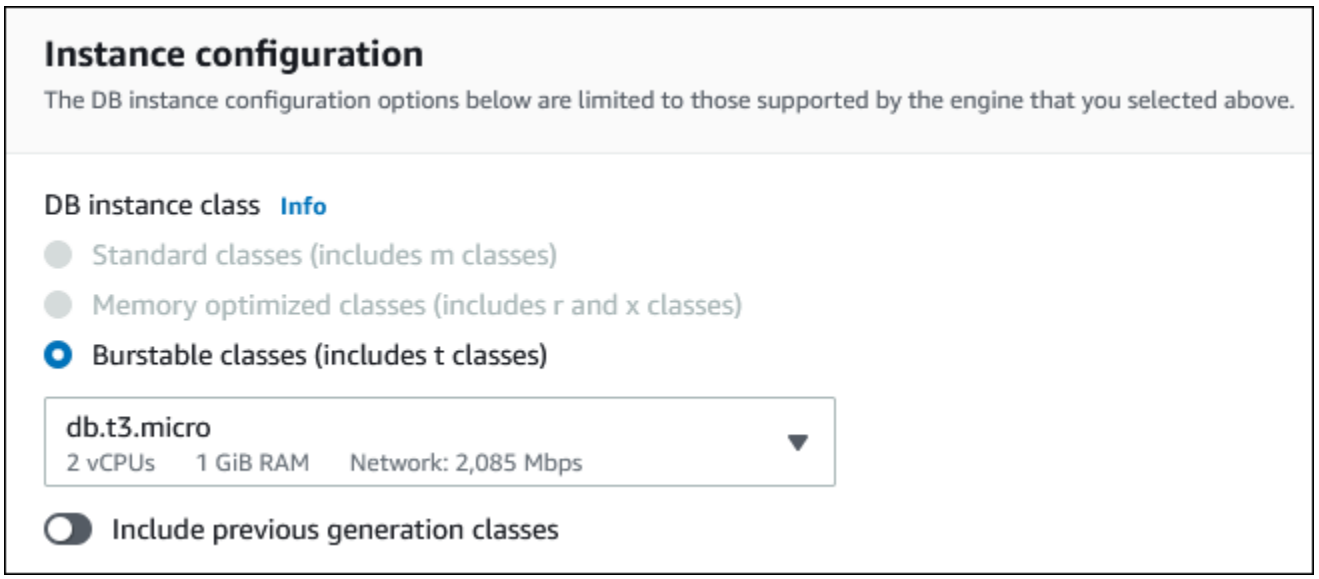
7. Per Modelli scegli Piano gratuito.

8. Nella sezione Availability and durability (Disponibilità e durata), mantieni i valori predefiniti.
9. Nella sezione Settings (Impostazioni) impostare questi valori:
   - DB Instance Identifier (Identificatore istanze database): **`tutorial-db-instance`**
   - Master username (Nome utente master): digita **`tutorial_user`**.
   - Auto generate a password (Genera automaticamente una password): lascia l'opzione disattivata.
   - Master password (Password master): scegli una password.
   - Confirm password (Conferma la password): –digitare di nuovo la password.



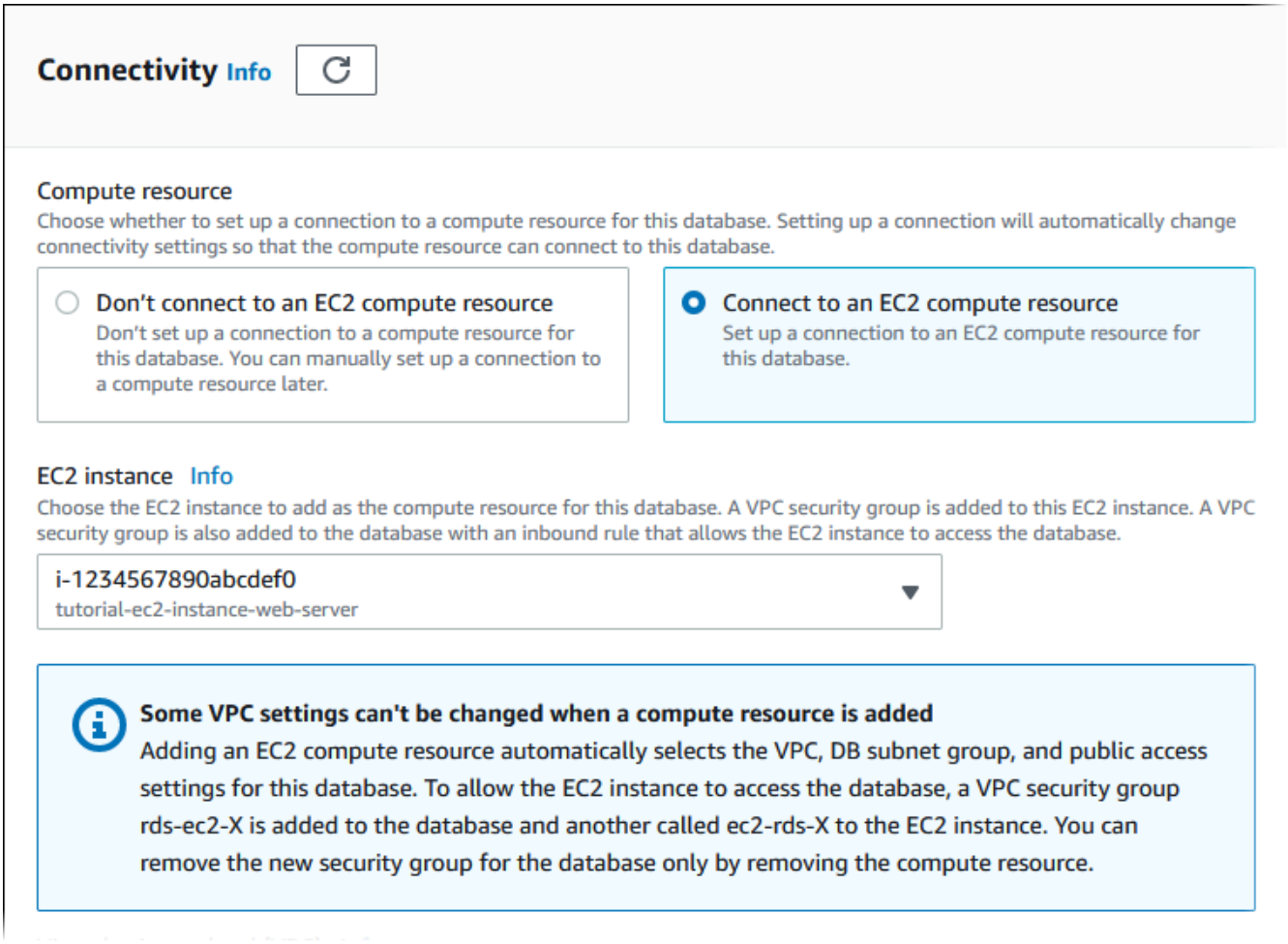
10. Nella sezione Instance configuration (Configurazione dell'istanza), imposta i seguenti valori:
    - Classi espandibili (include le classi t)

- db.t3.micro



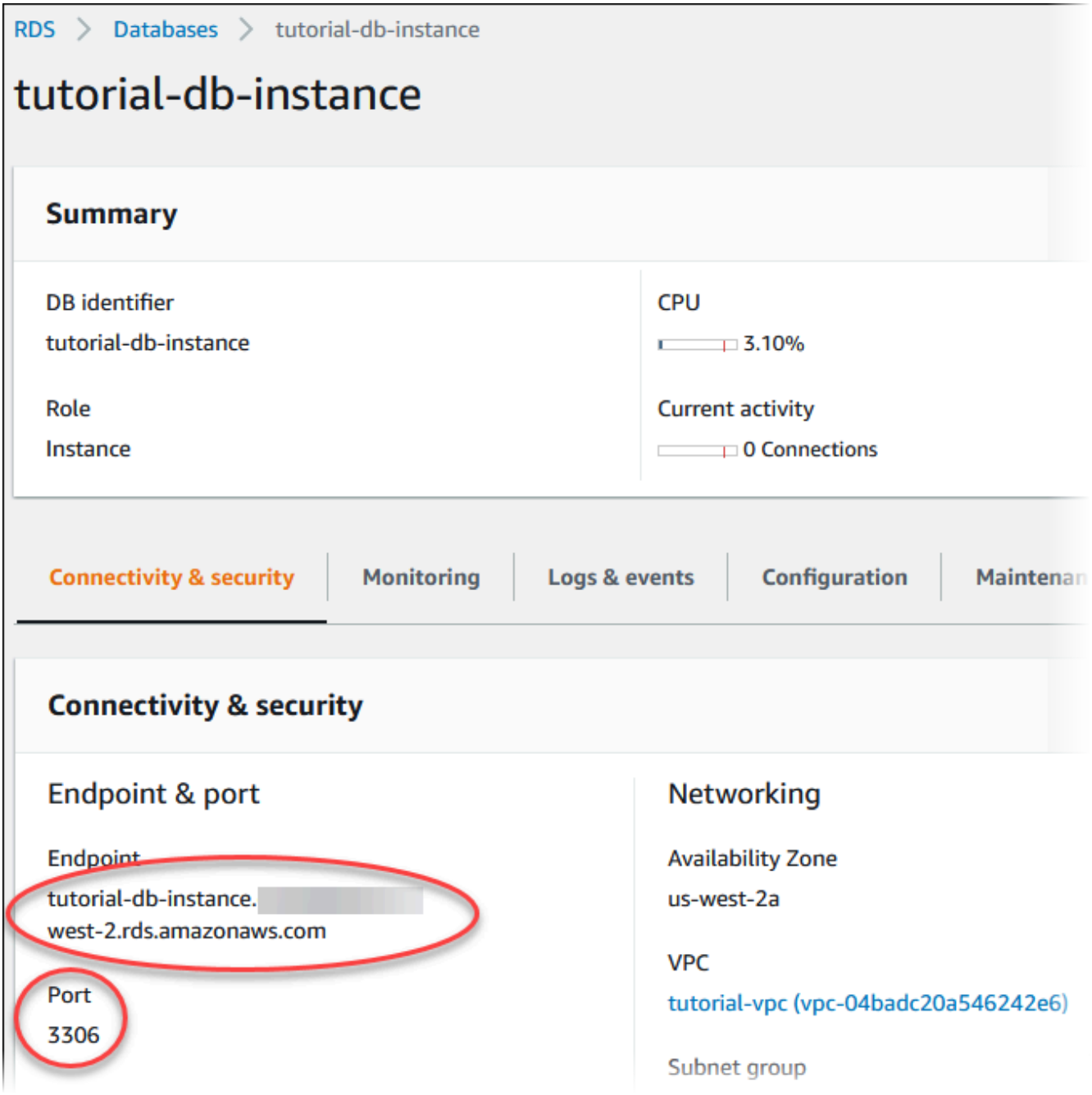
11. Nella sezione Storage (Archiviazione), mantieni le impostazioni di default.
12. Nella sezione Connectivity (Connettività), imposta i seguenti valori e lascia gli altri valori come predefiniti:

    - In Compute resource (Risorse di calcolo), seleziona Connect to an EC2 compute resource (Connetti a una risorsa di calcolo EC2).
    - Per l'istanza EC2, scegli l'istanza EC2 che hai creato in precedenza, ad esempio tutorial-ec2 -. instance-web-server



13. Nella sezione Autenticazione database verifica che sia selezionata l'opzione Autenticazione con password.
14. Aprire la sezione Additional configuration (Configurazione aggiuntiva) e specificare **`sample`** per Initial database name (Nome database iniziale). Lasciare le impostazioni predefinite per le altre opzioni.
15. Per creare l'istanza MariaDB, scegli Crea database.

    La nuova istanza database apparirà nell'elenco Databases con lo stato Creating (Creazione in corso).
16. Attendere che lo Status (Stato) della nuova istanza database appaia come Available (Disponibile). Quindi scegliere il nome dell'istanza database per visualizzarne i dettagli.
17. Nella sezione Connectivity & security (Connettività e sicurezza, visualizzare Endpoint e Port (Porta) dell'istanza database.



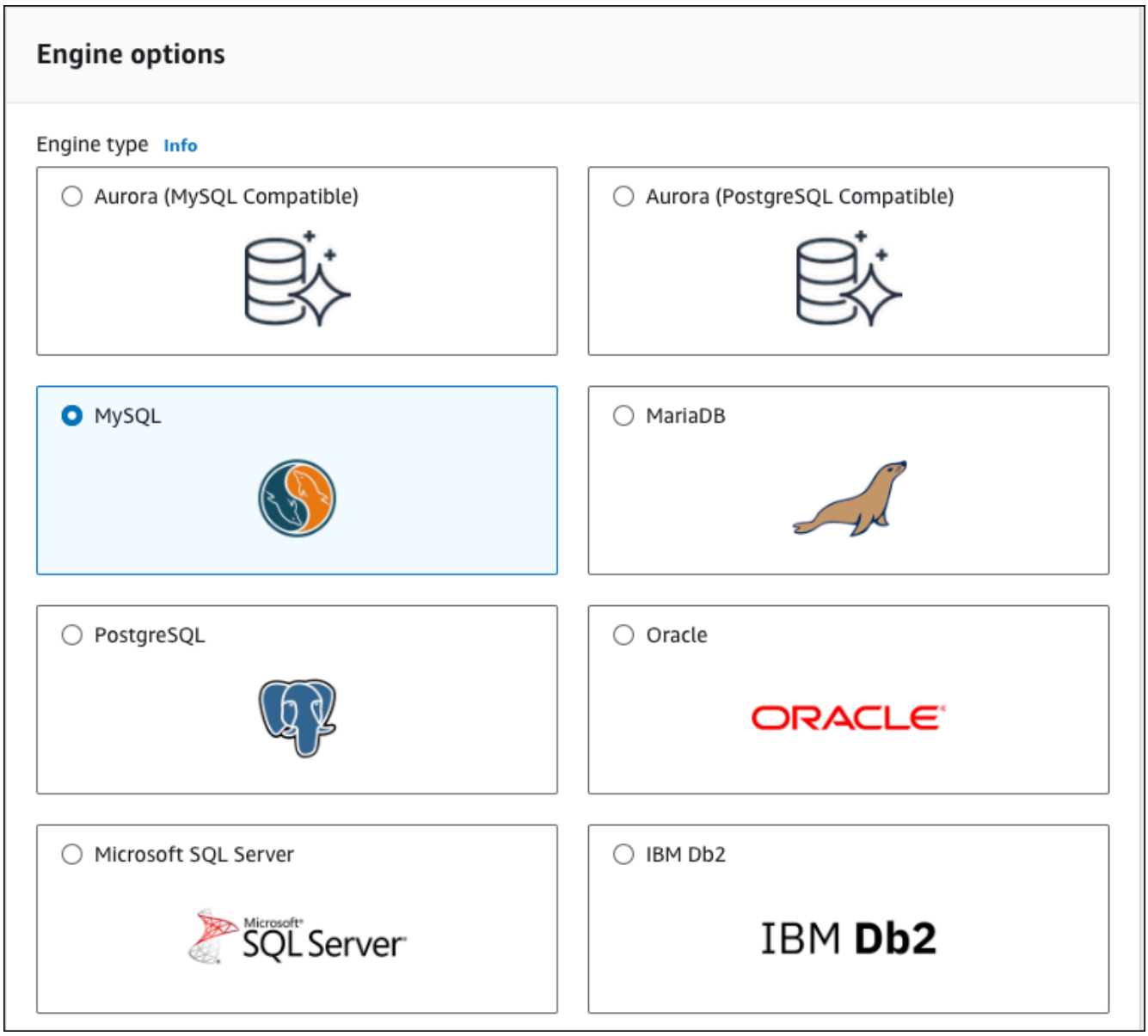
Prendere nota dell'endpoint e della porta dell'istanza database. Queste informazioni verranno utilizzate per effettuare la connessione del server Web all'istanza del database.

18. Completo Installazione di un server Web nell'istanza EC2.

RDS for MySQL

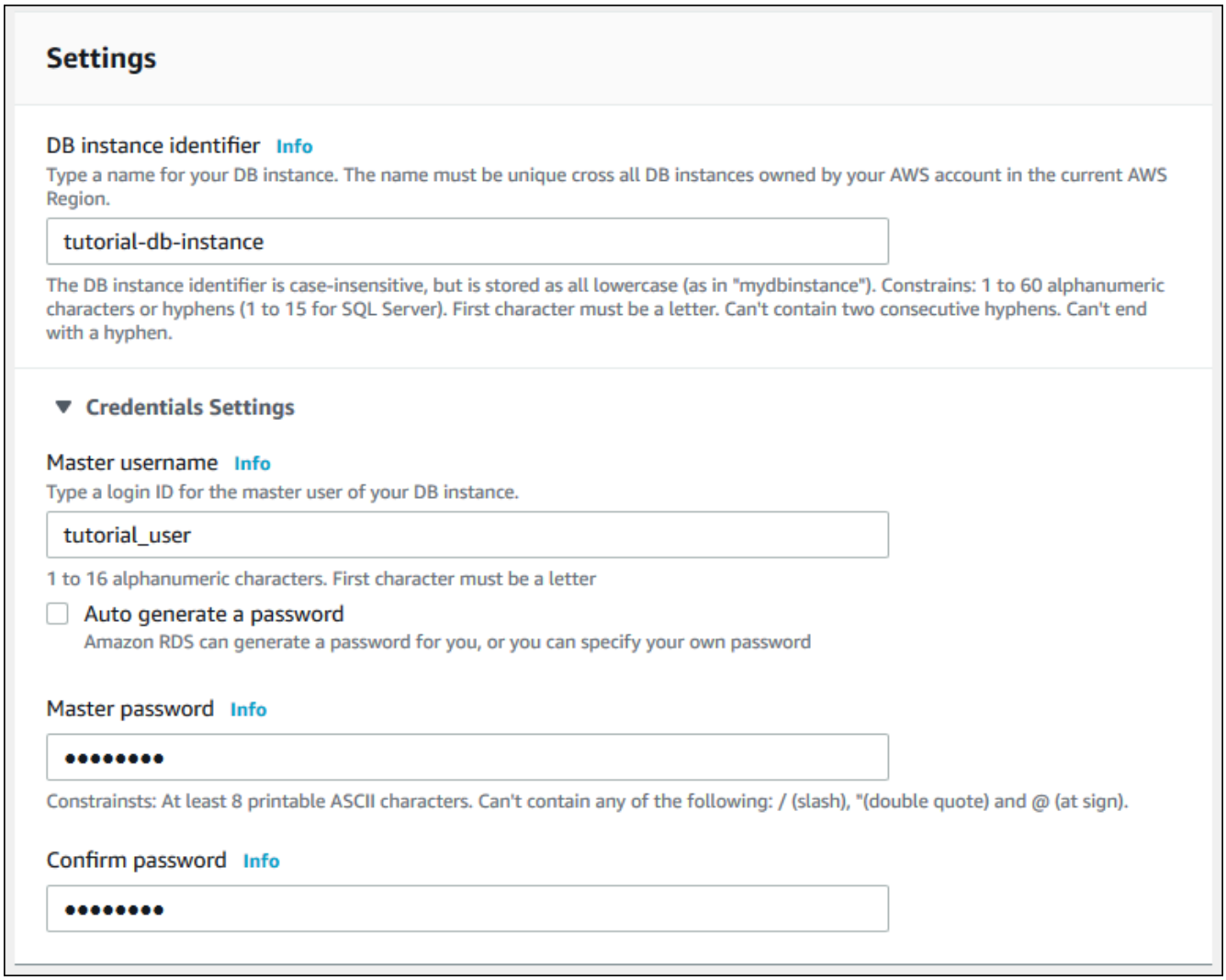
Per creare un'istanza database MySQL

1. Accedi alla AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nell'angolo in alto a destra della AWS Management Console, seleziona la Regione AWS. Deve essere identica a quella in cui hai creato l'istanza EC2.
3. Nel riquadro di navigazione, scegliere Databases (Database).
4. Scegliere Create database (Crea database).
5. Nella pagina Crea database scegli Creazione standard.
6. In Opzioni motore, seleziona MySQL.



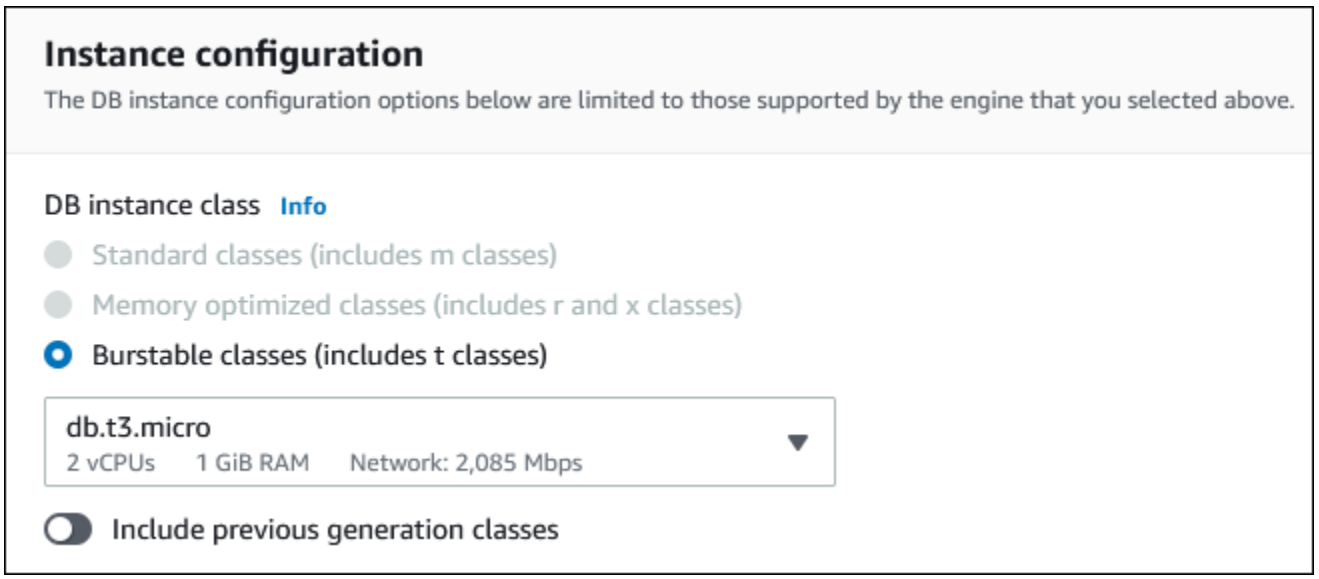
7. Per Modelli scegli Piano gratuito.

8. Nella sezione Availability and durability (Disponibilità e durata), mantieni i valori predefiniti.
9. Nella sezione Settings (Impostazioni) impostare questi valori:
   - DB Instance Identifier (Identificatore istanze database): **`tutorial-db-instance`**
   - Master username (Nome utente master): digita **`tutorial_user`**.
   - Auto generate a password (Genera automaticamente una password): lascia l'opzione disattivata.
   - Master password (Password master): scegli una password.
   - Confirm password (Conferma la password): –digitare di nuovo la password.



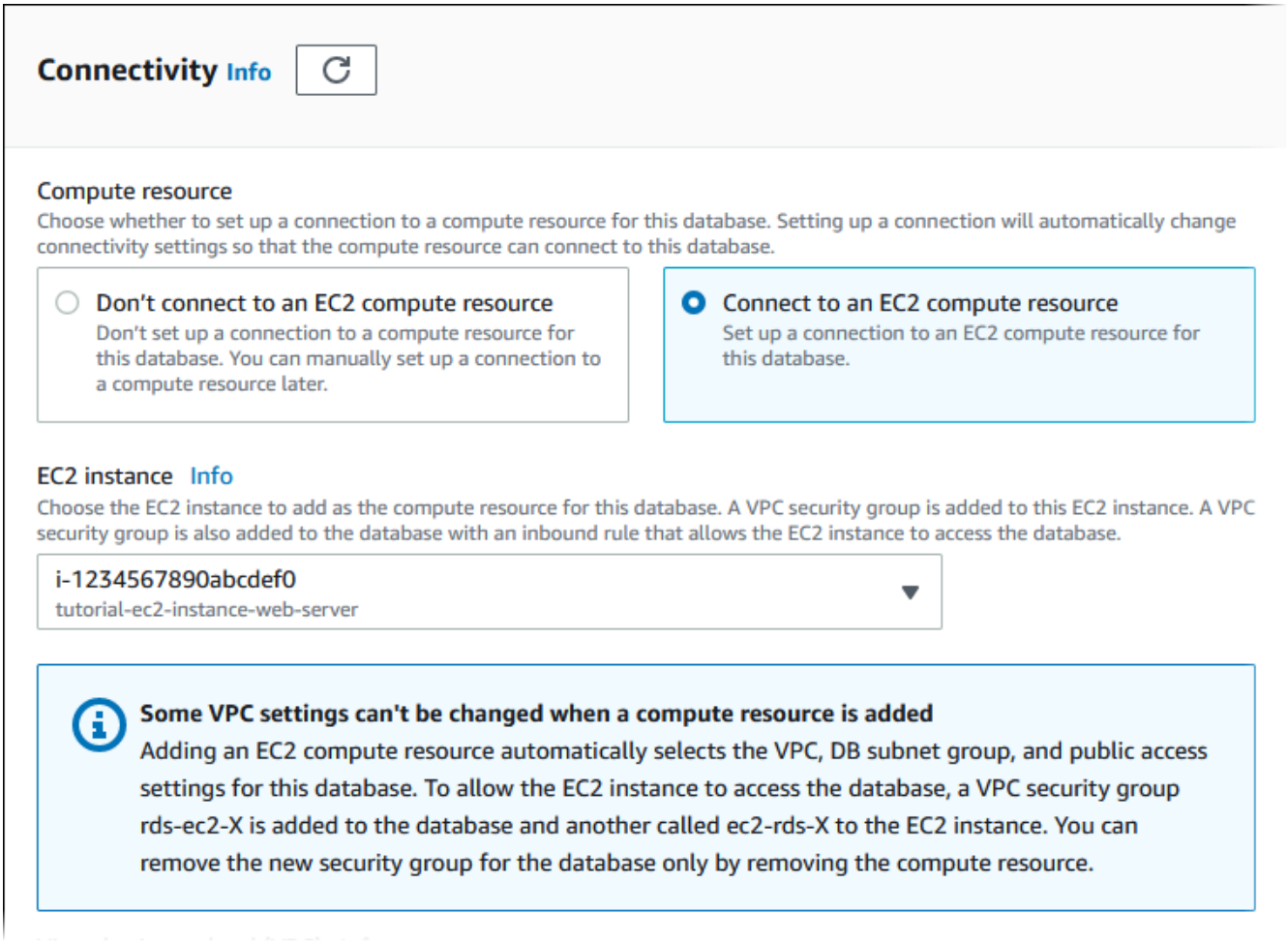
10. Nella sezione Instance configuration (Configurazione dell'istanza), imposta i seguenti valori:
    - Classi espandibili (include le classi t)

- db.t3.micro



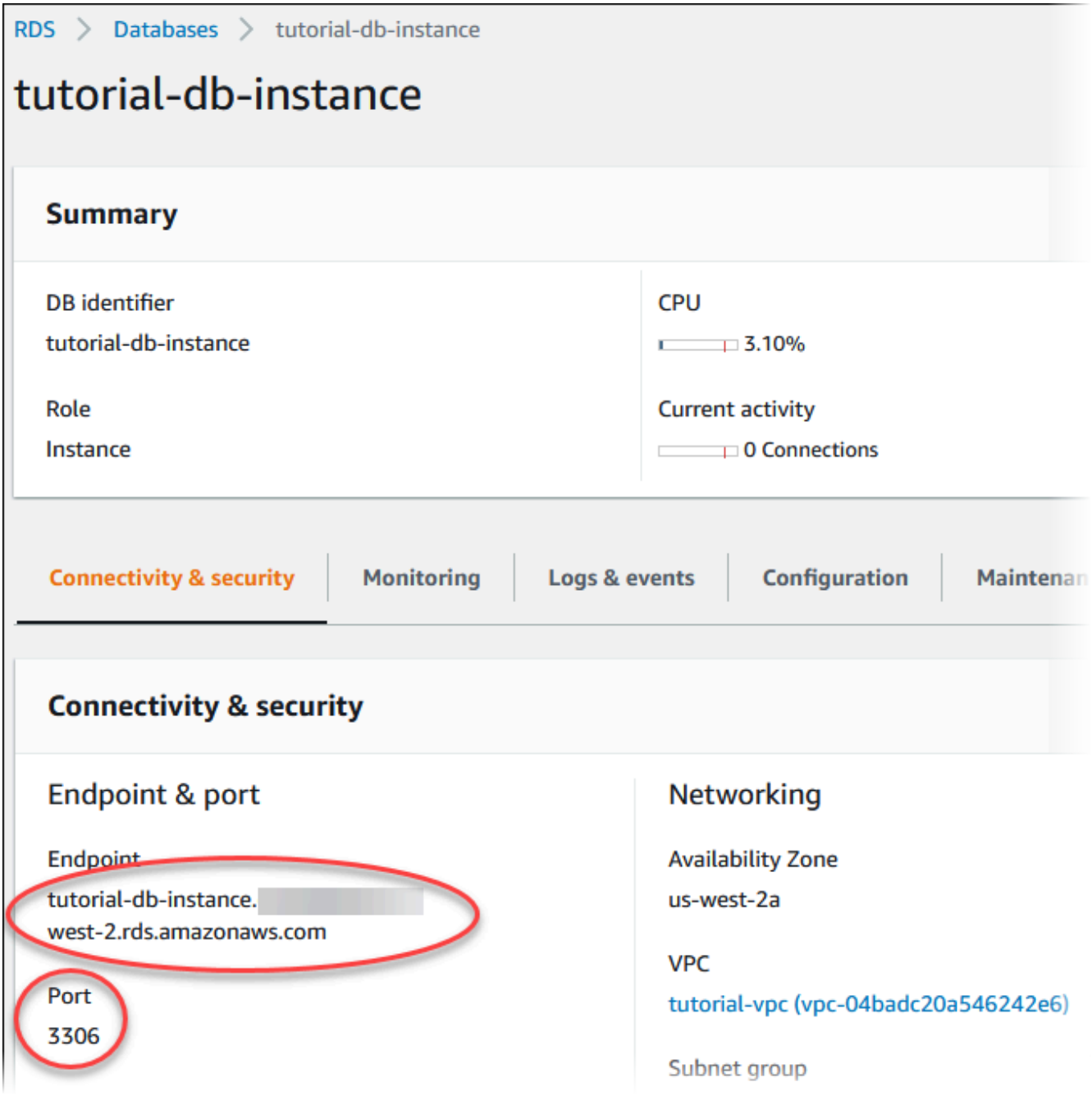
11. Nella sezione Storage (Archiviazione), mantieni le impostazioni di default.
12. Nella sezione Connectivity (Connettività), imposta i seguenti valori e lascia gli altri valori come predefiniti:

    - In Compute resource (Risorse di calcolo), seleziona Connect to an EC2 compute resource (Connetti a una risorsa di calcolo EC2).
    - Per l'istanza EC2, scegli l'istanza EC2 che hai creato in precedenza, ad esempio tutorial-ec2 -. instance-web-server



13. Nella sezione Autenticazione database verifica che sia selezionata l'opzione Autenticazione con password.
14. Aprire la sezione Additional configuration (Configurazione aggiuntiva) e specificare **`sample`** per Initial database name (Nome database iniziale). Lasciare le impostazioni predefinite per le altre opzioni.
15. Per creare l'istanza database MySQL, scegli Crea database.

    La nuova istanza database apparirà nell'elenco Databases con lo stato Creating (Creazione in corso).
16. Attendere che lo Status (Stato) della nuova istanza database appaia come Available (Disponibile). Quindi scegliere il nome dell'istanza database per visualizzarne i dettagli.
17. Nella sezione Connectivity & security (Connettività e sicurezza, visualizzare Endpoint e Port (Porta) dell'istanza database.



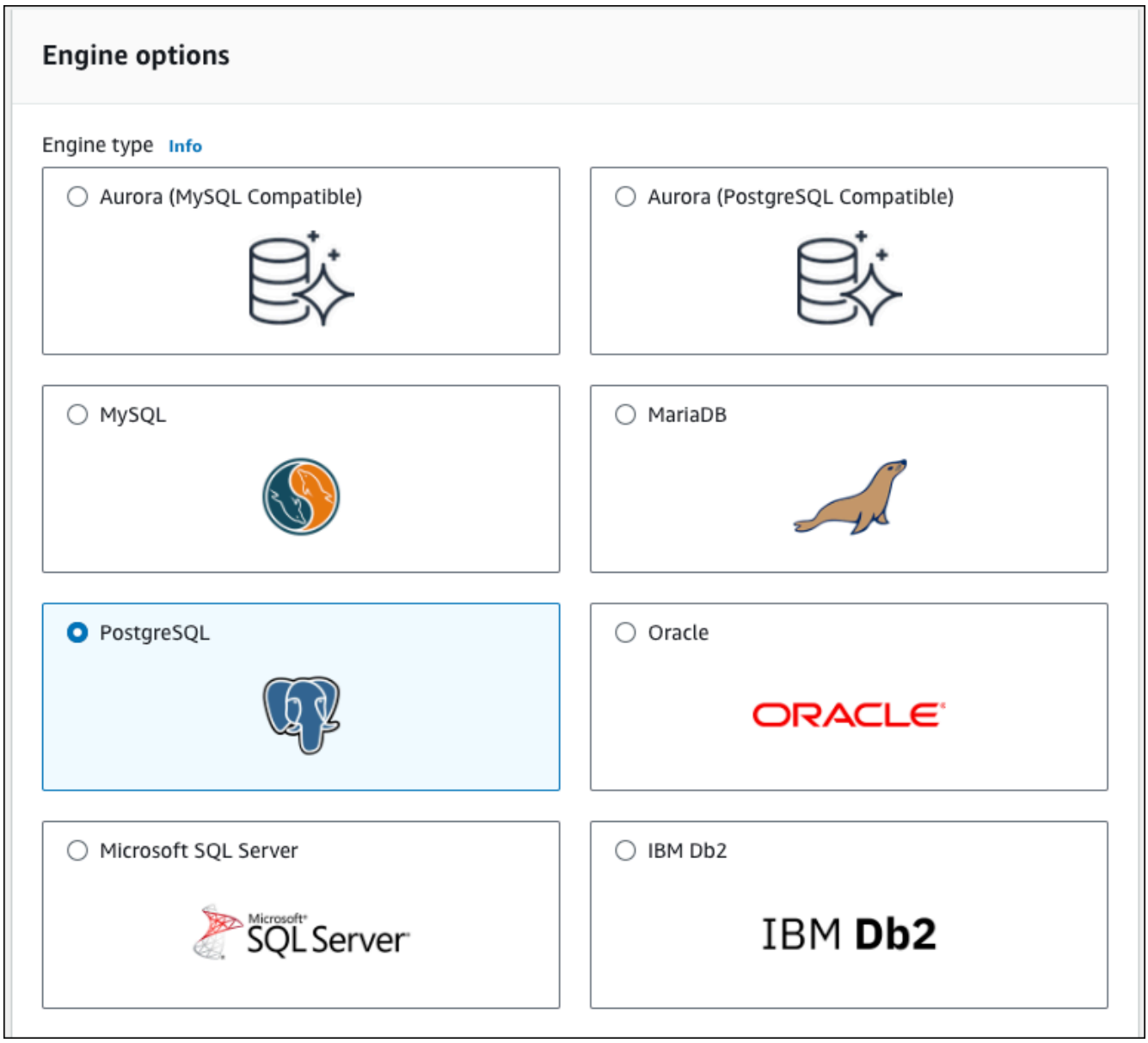
Prendere nota dell'endpoint e della porta dell'istanza database. Queste informazioni verranno utilizzate per effettuare la connessione del server Web all'istanza del database.

18. Completo Installazione di un server Web nell'istanza EC2.

RDS for PostgreSQL

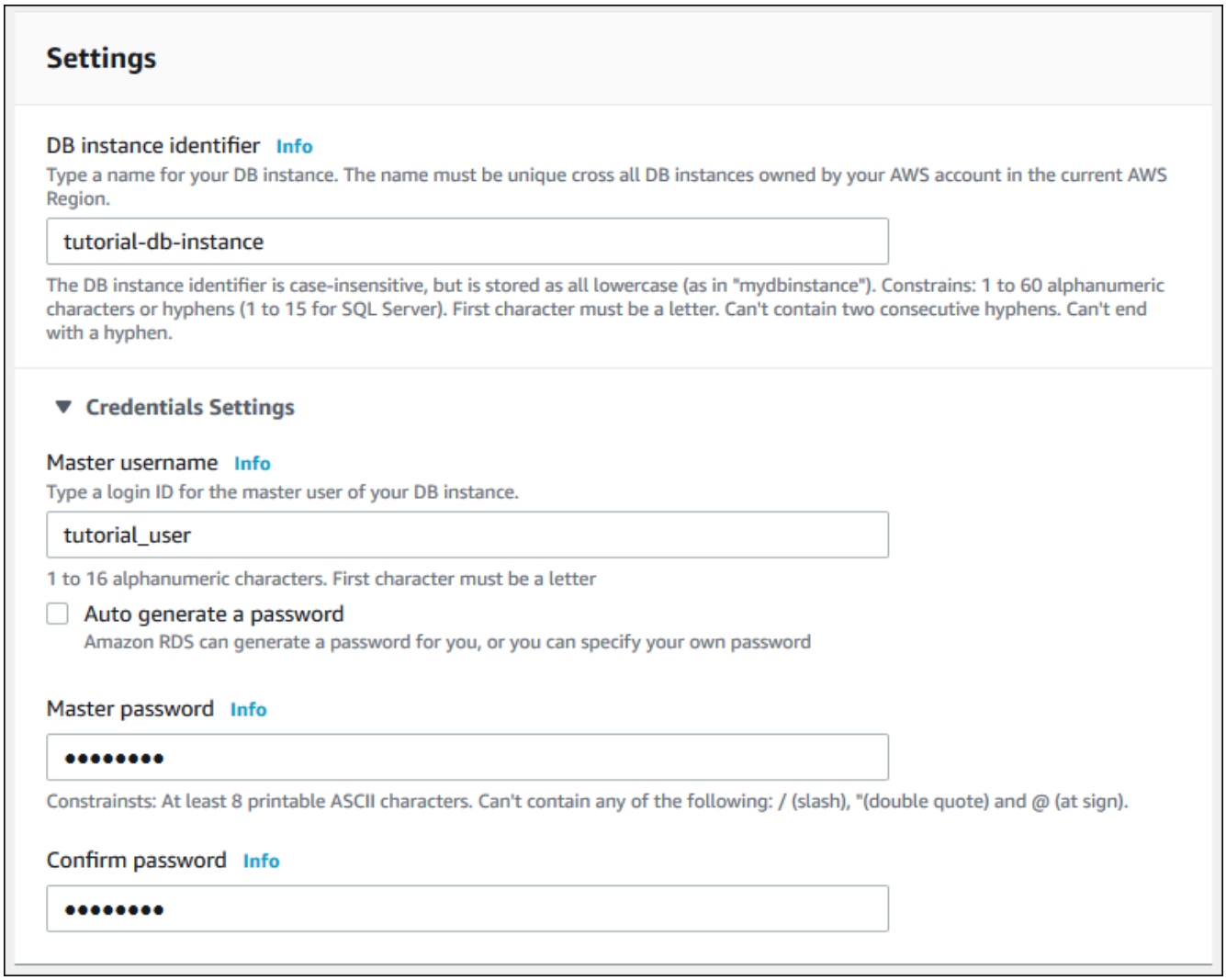
Creazione di un'istanza database PostgreSQL

1. Accedi alla AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nell'angolo in alto a destra della AWS Management Console, seleziona la Regione AWS. Deve essere identica a quella in cui hai creato l'istanza EC2.
3. Nel riquadro di navigazione, scegliere Databases (Database).
4. Scegliere Create database (Crea database).
5. Nella pagina Crea database scegli Creazione standard.
6. In Opzioni motore, seleziona PostgreSQL.



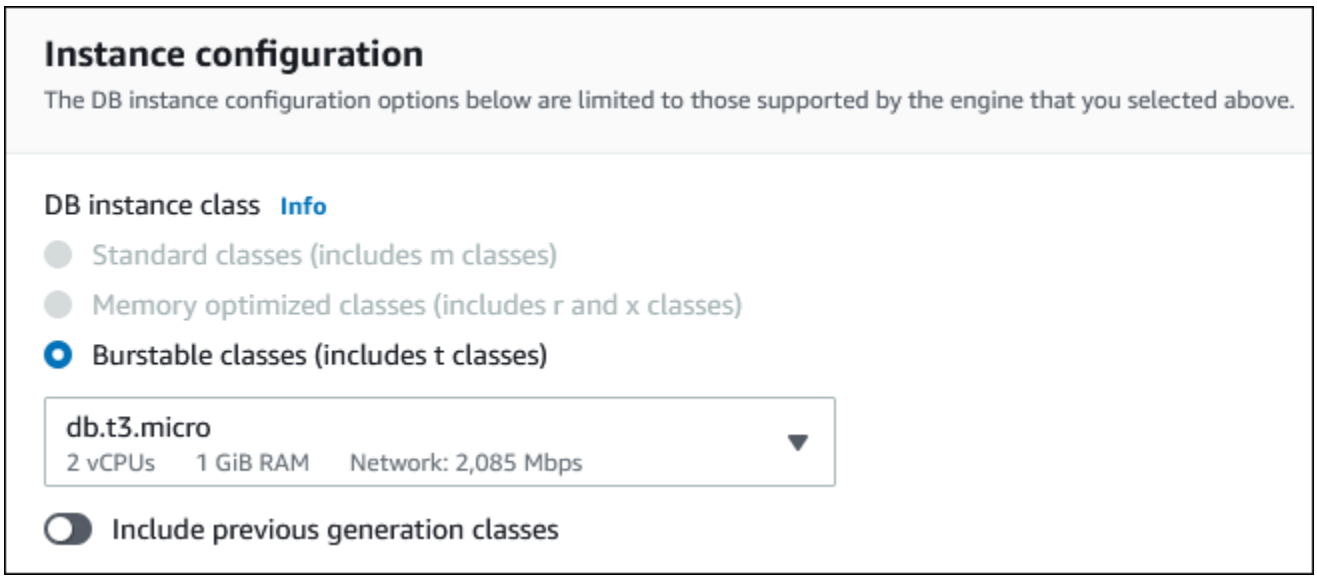
7. Per Modelli scegli Piano gratuito.

8. Nella sezione Availability and durability (Disponibilità e durata), mantieni i valori predefiniti.
9. Nella sezione Settings (Impostazioni) impostare questi valori:
   - DB Instance Identifier (Identificatore istanze database): **`tutorial-db-instance`**
   - Master username (Nome utente master): digita **`tutorial_user`**.
   - Auto generate a password (Genera automaticamente una password): lascia l'opzione disattivata.
   - Master password (Password master): scegli una password.
   - Confirm password (Conferma la password): –digitare di nuovo la password.



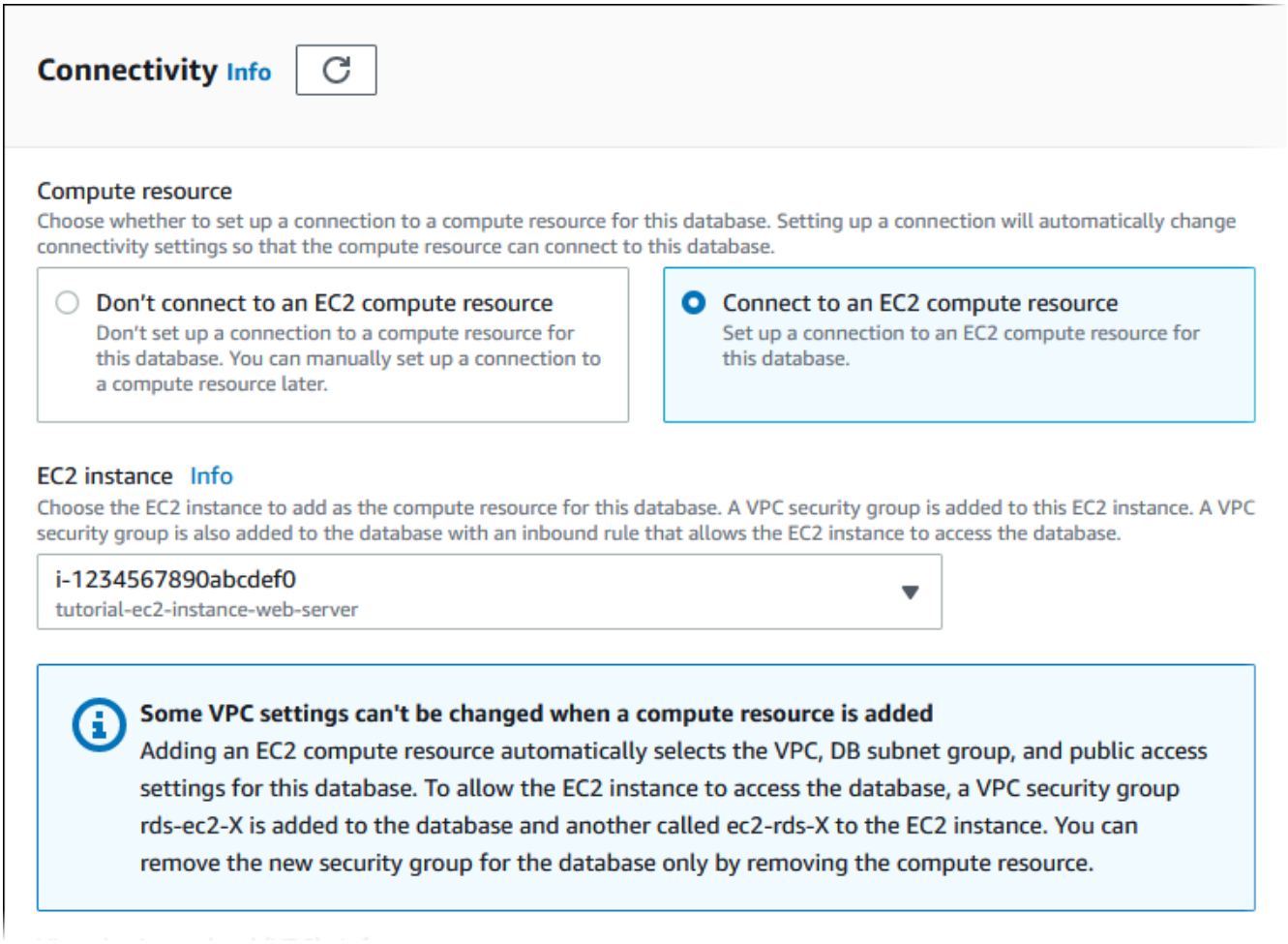
10. Nella sezione Instance configuration (Configurazione dell'istanza), imposta i seguenti valori:
    - Classi espandibili (include le classi t)

- db.t3.micro



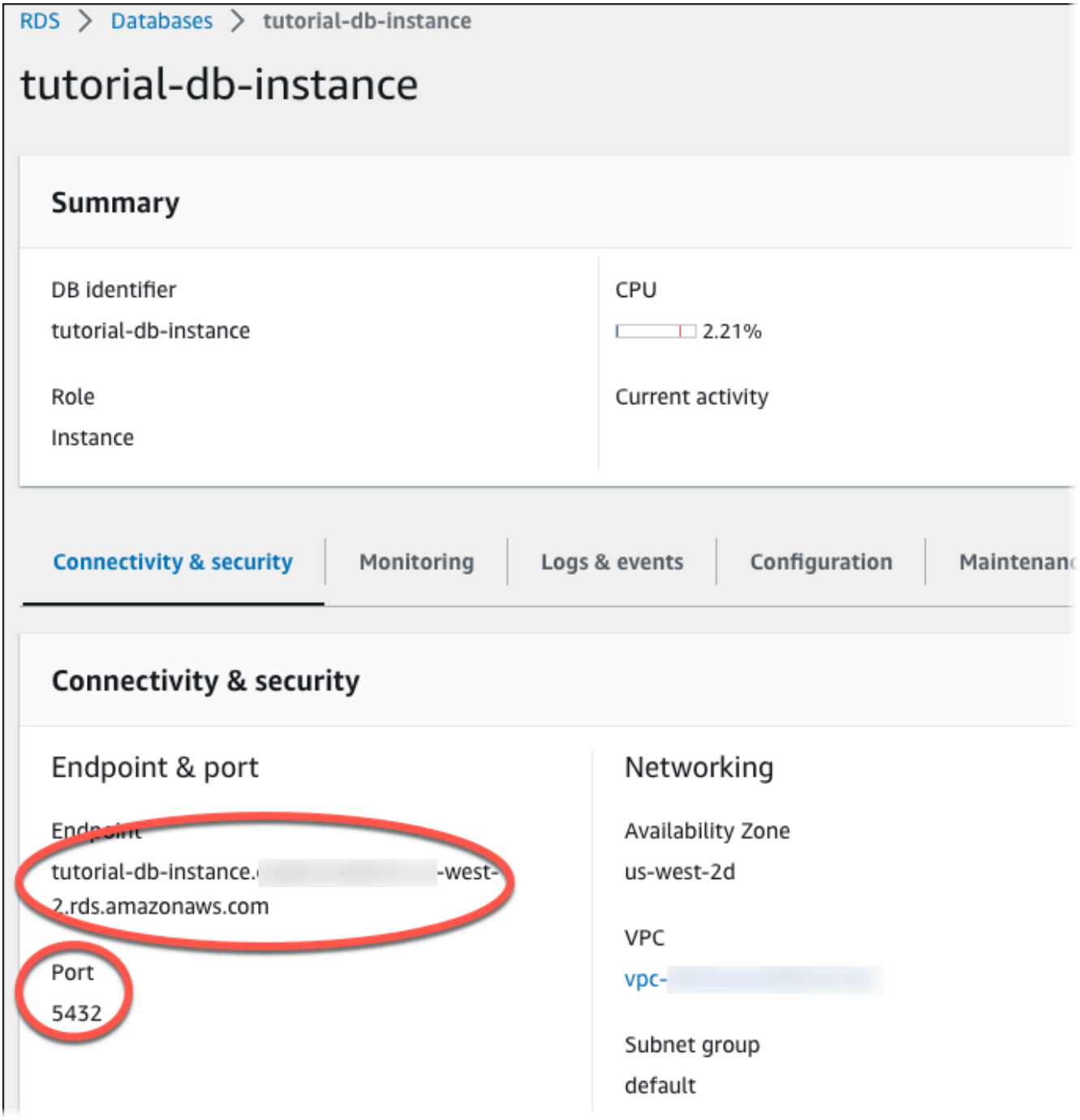
11. Nella sezione Storage (Archiviazione), mantieni le impostazioni di default.
12. Nella sezione Connectivity (Connettività), imposta i seguenti valori e lascia gli altri valori come predefiniti:

    - In Compute resource (Risorse di calcolo), seleziona Connect to an EC2 compute resource (Connetti a una risorsa di calcolo EC2).
    - Per l'istanza EC2, scegli l'istanza EC2 che hai creato in precedenza, ad esempio tutorial-ec2 -. instance-web-server



13. Nella sezione Autenticazione database verifica che sia selezionata l'opzione Autenticazione con password.
14. Aprire la sezione Additional configuration (Configurazione aggiuntiva) e specificare **`sample`** per Initial database name (Nome database iniziale). Lasciare le impostazioni predefinite per le altre opzioni.
15. Per creare l'istanza database PostgreSQL, scegli Crea database.

    La nuova istanza database apparirà nell'elenco Databases con lo stato Creating (Creazione in corso).
16. Attendere che lo Status (Stato) della nuova istanza database appaia come Available (Disponibile). Quindi scegliere il nome dell'istanza database per visualizzarne i dettagli.
17. Nella sezione Connectivity & security (Connettività e sicurezza, visualizzare Endpoint e Port (Porta) dell'istanza database.



Prendere nota dell'endpoint e della porta dell'istanza database. Queste informazioni verranno utilizzate per effettuare la connessione del server Web all'istanza del database.

18. Completo Installazione di un server Web nell'istanza EC2.

Installazione di un server Web nell'istanza EC2

Installa un server Web in un'istanza EC2 creata in Avvio di un'istanza EC2. Il server Web si connette all'istanza database Amazon RDS creata in Creazione di un'istanza database Amazon RDS.

Installazione di un server Web Apache con PHP e MariaDB

Esegui la connessione all'istanza EC2 e installa il server Web Apache.

Per effettuare la connessione all'istanza EC2 e installare il server Web Apache con PHP.

1. Esegui la connessione all'istanza EC2 creata in precedenza seguendo la procedura riportata in Connessione all'istanza di Linux nella Guida per l'utente per istanze Linux di Amazon EC2.

   Ti consigliamo di connetterti all'istanza EC2 tramite SSH. Se l'utilità client SSH è installata su Windows, Linux o Mac, puoi connetterti all'istanza utilizzando il comando nel seguente formato:

   ```
   ssh -i location_of_pem_file ec2-user@ec2-instance-public-dns-name
   ```

   Ad esempio, supponi che `ec2-database-connect-key-pair.pem` sia archiviato in `/dir1` su Linux e che il DNS IPv4 pubblico per l'istanza EC2 sia `ec2-12-345-678-90.compute-1.amazonaws.com`. Il comando SSH sarà simile al seguente:

   ```
   ssh -i /dir1/ec2-database-connect-key-pair.pem ec2-
   user@ec2-12-345-678-90.compute-1.amazonaws.com
   ```

2. Ottieni le ultime correzioni di bug e gli aggiornamenti di sicurezza aggiornando il software sulla tua istanza EC2. A questo scopo, eseguire il comando seguente.

   > **Note**
   >
   > L'opzione `-y` installa gli aggiornamenti senza chiedere conferma. Per esaminare gli aggiornamenti prima di installarli, omettere questa opzione.

   ```
   sudo dnf update -y
   ```

3. Al completamento degli aggiornamenti, installa il server Web Apache, PHP e il software MariaDB utilizzando i comandi seguenti. Con questo comando vengono installati contemporaneamente più pacchetti software e dipendenze correlate.

   MariaDB & MySQL

   ```
   sudo dnf install -y httpd php php-mysqli mariadb105
   ```

   PostgreSQL

   ```
   sudo dnf install -y httpd php php-pgsql postgresql15
   ```

   Se si verifica un errore, è possibile che l'istanza non sia stata lanciata con un'AMI Amazon Linux 2023. Puoi invece utilizzare l'AMI Amazon Linux 2. È possibile visualizzare la versione di Amazon Linux con il comando seguente.

   ```
   cat /etc/system-release
   ```

   Per ulteriori informazioni, consulta la pagina relativa all'aggiornamento del software dell'istanza.

4. Avviare il server Web con il comando visualizzato di seguito.

   ```
   sudo systemctl start httpd
   ```

   È possibile verificare che il server Web sia installato e avviato correttamente. A tale scopo, immettere il nome DNS (Domain Name System) pubblico dell'istanza EC2 nella barra degli indirizzi di un browser Web, ad esempio: `http://ec2-42-8-168-21.us-west-1.compute.amazonaws.com`. Se il server Web è in esecuzione, verrà visualizzata la pagina di test di Apache.

   Se la pagina di test Apache non viene visualizzata, controllare le regole in entrata per il gruppo di sicurezza VPC creato in Tutorial: Creazione di un Amazon VPC da utilizzare con un'istanza database (solo IPv4). Assicurati che le regole in entrata ne includano una che consenta l'accesso HTTP (porta 80) per l'indirizzo IP utilizzato per connettersi al server Web.

> **Note**
>
> La pagina di test di Apache viene visualizzata solo quando la directory principale dei documenti è vuota, `/var/www/html`. Dopo aver aggiunto contenuti alla directory root dei documenti, i contenuti vengono visualizzati all'indirizzo DNS pubblico dell'istanza EC2. Prima di questo punto, vengono visualizzati nella pagina di test di Apache.

5. Configurare il server Web affinché si avvii a ogni avvio del sistema tramite il comando `systemctl`.

```
sudo systemctl enable httpd
```

Per permettere a `ec2-user` di gestire file nella directory principale predefinita del server Web Apache, è necessario modificare la proprietà e le autorizzazioni della directory `/var/www`. Sono disponibili molti modi per completare questa attività. In questo tutorial, aggiungi l'utente `ec2-user` al gruppo `apache` per assegnare la proprietà del gruppo `apache` della directory `/var/www` e assegnare autorizzazioni di scrittura al gruppo.

Per impostare le autorizzazioni dei file sul server Web Apache

1. Aggiungere l'utente `ec2-user` al gruppo `apache`.

```
sudo usermod -a -G apache ec2-user
```

2. Per aggiornare le autorizzazioni e includere il nuovo gruppo `apache`, eseguire la disconnessione.

```
exit
```

3. Effettuare nuovamente l'accesso e verificare che il gruppo `apache` esista mediante il comando `groups`.

```
groups
```

   L'output avrà un aspetto simile al seguente:

```
ec2-user adm wheel apache systemd-journal
```

4. Cambiare la proprietà del gruppo della directory `/var/www` e dei suoi contenuti sul gruppo `apache`.

```
sudo chown -R ec2-user:apache /var/www
```

5. Cambiare le autorizzazioni della directory `/var/www` e delle sue sottodirectory per aggiungere le autorizzazioni di scrittura di gruppo e impostare l'ID di gruppo per le sottodirectory create in futuro.

```
sudo chmod 2775 /var/www
find /var/www -type d -exec sudo chmod 2775 {} \;
```

6. Cambiare le autorizzazioni in modo ricorsivo per i file nella directory `/var/www` e nelle sue sottodirectory per aggiungere le autorizzazioni di scrittura di gruppo.

```
find /var/www -type f -exec sudo chmod 0664 {} \;
```

Ora, `ec2-user` (e qualsiasi membro futuro del gruppo `apache`) può aggiungere, eliminare e modificare i file nella root del documento di Apache. Questo consente di aggiungere contenuti, ad esempio un sito Web statico o un'applicazione PHP.

> **Note**
>
> Un server Web che esegue il protocollo HTTP non offre alcuna sicurezza di trasporto per i dati inviati e ricevuti. Quando ti connetti a un server HTTP tramite un browser Web, la molte informazioni sono visibili a persone non autorizzate in qualsiasi punto del percorso di rete. Queste informazioni includono gli URL visitati, il contenuto delle pagine Web ricevute e i contenuti (incluse le password) di eventuali moduli HTML.
> La best practice per la protezione del tuo server Web prevede l'installazione del supporto per HTTPS (HTTP Secure). Questo protocollo protegge i dati con la crittografia SSL/TLS. Per ulteriori informazioni, consulta Esercitazione: Configurare SSL/TLS con l'AMI di Amazon Linux nella Guida per l'utente di Amazon EC2 .

Connessione del server web Apache all'istanza

Successivamente, aggiungere contenuti al server Web Apache che effettua la connessione all'istanza database Amazon RDS.

Per aggiungere contenuti al server Web Apache che effettua la connessione all'istanza database.

1. Mentre è ancora in corso la connessione all'istanza EC2, modificare la directory in `/var/www` e creare una nuova sottodirectory denominata `inc`.

```
cd /var/www
mkdir inc
cd inc
```

2. Creare un nuovo file nella directory `inc` denominato `dbinfo.inc` e poi modificarlo con nano (o un altro editor a scelta).

```
>dbinfo.inc
nano dbinfo.inc
```

3. Aggiungi i seguenti contenuti al file `dbinfo.inc`. Qui, *`db_instance_endpoint`* è l'endpoint dell'istanza DB, senza la porta, per l'istanza database.

> **Note**
>
> Si consiglia di inserire le informazioni relative al nome utente e alla password in una cartella che non fa parte della directory principale del documento per il server Web. In questo modo si riduce la possibilità che le informazioni di sicurezza vengano esposte. Assicurati di modificare `master password` in una password adatta per la tua applicazione.

```
<?php

define('DB_SERVER', 'db_instance_endpoint');
define('DB_USERNAME', 'tutorial_user');
define('DB_PASSWORD', 'master password');
define('DB_DATABASE', 'sample');
?>
```

4. Salvare e chiudere il file `dbinfo.inc`. Se stai usando nano, salva e chiudi il file usando Ctrl+S e Ctrl+X.

5. Cambiare la directory in `/var/www/html`.

```
cd /var/www/html
```

6. Creare un nuovo file nella directory `html` denominato `SamplePage.php` e poi modificarlo con nano (o un altro editor a scelta).

```
>SamplePage.php
nano SamplePage.php
```

7. Aggiungere i seguenti contenuti al file `SamplePage.php`:

   MariaDB & MySQL

```
<?php include "../inc/dbinfo.inc"; ?>
<html>
<body>
<h1>Sample page</h1>
<?php

  /* Connect to MySQL and select the database. */
  $connection = mysqli_connect(DB_SERVER, DB_USERNAME, DB_PASSWORD);

  if (mysqli_connect_errno()) echo "Failed to connect to MySQL: " .
 mysqli_connect_error();

  $database = mysqli_select_db($connection, DB_DATABASE);

  /* Ensure that the EMPLOYEES table exists. */
  VerifyEmployeesTable($connection, DB_DATABASE);

  /* If input fields are populated, add a row to the EMPLOYEES table. */
  $employee_name = htmlentities($_POST['NAME']);
  $employee_address = htmlentities($_POST['ADDRESS']);

  if (strlen($employee_name) || strlen($employee_address)) {
    AddEmployee($connection, $employee_name, $employee_address);
  }
?>

<!-- Input form -->
<form action="<?PHP echo $_SERVER['SCRIPT_NAME'] ?>" method="POST">
  <table border="0">
    <tr>
```

```
      <td>NAME</td>
      <td>ADDRESS</td>
    </tr>
    <tr>
      <td>
        <input type="text" name="NAME" maxlength="45" size="30" />
      </td>
      <td>
        <input type="text" name="ADDRESS" maxlength="90" size="60" />
      </td>
      <td>
        <input type="submit" value="Add Data" />
      </td>
    </tr>
  </table>
</form>

<!-- Display table data. -->
<table border="1" cellpadding="2" cellspacing="2">
  <tr>
    <td>ID</td>
    <td>NAME</td>
    <td>ADDRESS</td>
  </tr>

<?php

$result = mysqli_query($connection, "SELECT * FROM EMPLOYEES");

while($query_data = mysqli_fetch_row($result)) {
  echo "<tr>";
  echo "<td>",$query_data[0], "</td>",
       "<td>",$query_data[1], "</td>",
       "<td>",$query_data[2], "</td>";
  echo "</tr>";
}
?>

</table>

<!-- Clean up. -->
<?php

  mysqli_free_result($result);
```

```
  mysqli_close($connection);

?>

</body>
</html>


<?php

/* Add an employee to the table. */
function AddEmployee($connection, $name, $address) {
   $n = mysqli_real_escape_string($connection, $name);
   $a = mysqli_real_escape_string($connection, $address);

   $query = "INSERT INTO EMPLOYEES (NAME, ADDRESS) VALUES ('$n', '$a');";

   if(!mysqli_query($connection, $query)) echo("<p>Error adding employee data.</
p>");
}

/* Check whether the table exists and, if not, create it. */
function VerifyEmployeesTable($connection, $dbName) {
  if(!TableExists("EMPLOYEES", $connection, $dbName))
  {
     $query = "CREATE TABLE EMPLOYEES (
         ID int(11) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
         NAME VARCHAR(45),
         ADDRESS VARCHAR(90)
       )";

     if(!mysqli_query($connection, $query)) echo("<p>Error creating table.</
p>");
  }
}

/* Check for the existence of a table. */
function TableExists($tableName, $connection, $dbName) {
  $t = mysqli_real_escape_string($connection, $tableName);
  $d = mysqli_real_escape_string($connection, $dbName);

  $checktable = mysqli_query($connection,
      "SELECT TABLE_NAME FROM information_schema.TABLES WHERE TABLE_NAME = '$t'
 AND TABLE_SCHEMA = '$d'");
```

```
  if(mysqli_num_rows($checktable) > 0) return true;

  return false;
}
?>
```

PostgreSQL

```
<?php include "../inc/dbinfo.inc"; ?>

<html>
<body>
<h1>Sample page</h1>
<?php

/* Connect to PostgreSQL and select the database. */
$constring = "host=" . DB_SERVER . " dbname=" . DB_DATABASE . " user=" .
 DB_USERNAME . " password=" . DB_PASSWORD ;
$connection = pg_connect($constring);

if (!$connection){
 echo "Failed to connect to PostgreSQL";
 exit;
}

/* Ensure that the EMPLOYEES table exists. */
VerifyEmployeesTable($connection, DB_DATABASE);

/* If input fields are populated, add a row to the EMPLOYEES table. */
$employee_name = htmlentities($_POST['NAME']);
$employee_address = htmlentities($_POST['ADDRESS']);

if (strlen($employee_name) || strlen($employee_address)) {
  AddEmployee($connection, $employee_name, $employee_address);
}

?>

<!-- Input form -->
<form action="<?PHP echo $_SERVER['SCRIPT_NAME'] ?>" method="POST">
  <table border="0">
```

```
    <tr>
      <td>NAME</td>
      <td>ADDRESS</td>
    </tr>
    <tr>
      <td>
    <input type="text" name="NAME" maxlength="45" size="30" />
      </td>
      <td>
    <input type="text" name="ADDRESS" maxlength="90" size="60" />
      </td>
      <td>
    <input type="submit" value="Add Data" />
      </td>
    </tr>
  </table>
</form>
<!-- Display table data. -->
<table border="1" cellpadding="2" cellspacing="2">
  <tr>
    <td>ID</td>
    <td>NAME</td>
    <td>ADDRESS</td>
  </tr>

<?php

$result = pg_query($connection, "SELECT * FROM EMPLOYEES");

while($query_data = pg_fetch_row($result)) {
  echo "<tr>";
  echo "<td>",$query_data[0], "</td>",
       "<td>",$query_data[1], "</td>",
       "<td>",$query_data[2], "</td>";
  echo "</tr>";
}
?>
</table>

<!-- Clean up. -->
<?php

  pg_free_result($result);
  pg_close($connection);
```

```
?>
</body>
</html>


<?php

/* Add an employee to the table. */
function AddEmployee($connection, $name, $address) {
   $n = pg_escape_string($name);
   $a = pg_escape_string($address);
   echo "Forming Query";
   $query = "INSERT INTO EMPLOYEES (NAME, ADDRESS) VALUES ('$n', '$a');";

   if(!pg_query($connection, $query)) echo("<p>Error adding employee data.</
p>");
}

/* Check whether the table exists and, if not, create it. */
function VerifyEmployeesTable($connection, $dbName) {
  if(!TableExists("EMPLOYEES", $connection, $dbName))
  {
     $query = "CREATE TABLE EMPLOYEES (
         ID serial PRIMARY KEY,
         NAME VARCHAR(45),
         ADDRESS VARCHAR(90)
       )";

     if(!pg_query($connection, $query)) echo("<p>Error creating table.</p>");
  }
}
/* Check for the existence of a table. */
function TableExists($tableName, $connection, $dbName) {
  $t = strtolower(pg_escape_string($tableName)); //table name is case sensitive
  $d = pg_escape_string($dbName); //schema is 'public' instead of 'sample' db
 name so not using that

  $query = "SELECT TABLE_NAME FROM information_schema.TABLES WHERE TABLE_NAME =
 '$t';";
  $checktable = pg_query($connection, $query);

  if (pg_num_rows($checktable) >0) return true;
  return false;
```

```
}
?>
```

8. Salvare e chiudere il file `SamplePage.php`.
9. Verificare che il server Web effettui correttamente la connessione all'istanza aprendo un browser web e navigando fino a `http://EC2 instance endpoint/SamplePage.php`, ad esempio: `http://ec2-12-345-67-890.us-west-2.compute.amazonaws.com/SamplePage.php`.

È possibile utilizzare `SamplePage.php` per aggiungere dati all'istanza. I dati aggiunti verranno visualizzati nella pagina. Per verificare che i dati siano stati inseriti nella tabella, installa il client MySQL nell'istanza Amazon EC2. Esegui quindi la connessione all'istanza database ed esegui la query sulla tabella.

Per informazioni sull'installazione di un client SQL e la connessione a un'istanza database Oracle, consulta Connessione a un'istanza database che esegue il motore di database di MySQL.

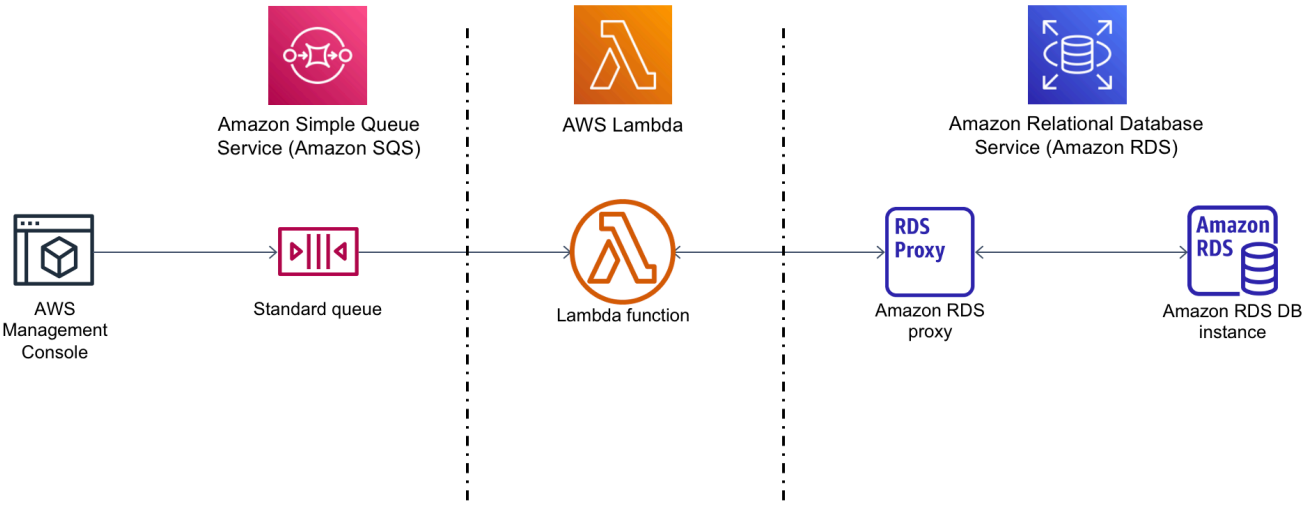
Per assicurarsi che l'istanza sia il più possibile sicura, verificare che le fonti esterne al VPC non possano connettersi all'istanza.

Dopo aver terminato il test del server Web e del database, è necessario eliminare l' il cluster DB e l'istanza Amazon EC2.

- Per eliminare un'istanza database, segui le istruzioni riportate in Eliminazione di un'istanza database. Non è necessario creare uno snapshot finale.
- Per terminare un'istanza Amazon EC2, segui le istruzioni riportate in Termina istanza nella Guida per l'utente di Amazon EC2.

Tutorial: utilizzo di una funzione Lambda per accedere a un database Amazon RDS

In questo tutorial, viene utilizzata una funzione Lambda per scrivere dati su un database Amazon Relational Database Service (Amazon RDS) tramite RDS Proxy. La funzione Lambda legge i record da una coda Amazon Simple Queue Service (Amazon SQS) e scrive un nuovo elemento in una tabella del database ogni volta che viene aggiunto un messaggio. In questo esempio, viene utilizzata la AWS Management Console per aggiungere manualmente i messaggi alla coda. Il diagramma seguente mostra le AWS risorse che usi per completare il tutorial.



Con Amazon RDS, è possibile eseguire un database relazionale gestito nel cloud utilizzando prodotti database comuni come Microsoft SQL Server, MariaDB, MySQL, Oracle Database e PostgreSQL. Utilizzando Lambda per accedere al tuo database, puoi leggere e scrivere dati in risposta a eventi, come ad esempio un nuovo cliente che si registra sul tuo sito Web. La funzione, l'istanza database e il proxy vengono dimensionati automaticamente per rispondere ai periodi di forte domanda.

Per completare questo tutorial, completa le seguenti attività:

1. Avvia un'istanza di database RDS for MySQL e un proxy nel tuo Account AWS VPC predefinito.
2. Crea e testa una funzione Lambda che crea una nuova tabella nel tuo database e vi scrive i dati.
3. Crea una coda Amazon SQS e configurala per richiamare la funzione Lambda ogni volta che viene aggiunto un nuovo messaggio.

4. Verifica la configurazione completa aggiungendo messaggi alla coda utilizzando AWS Management Console e monitorando i risultati utilizzando Logs. CloudWatch

Completando questi passaggi, imparerai:

- Come usare Amazon RDS per creare un'istanza database e un proxy e connettere una funzione Lambda al proxy.
- Come usare Lambda per eseguire operazioni di creazione e lettura su un database Amazon RDS.
- Come utilizzare Amazon SQS per richiamare una funzione Lambda.

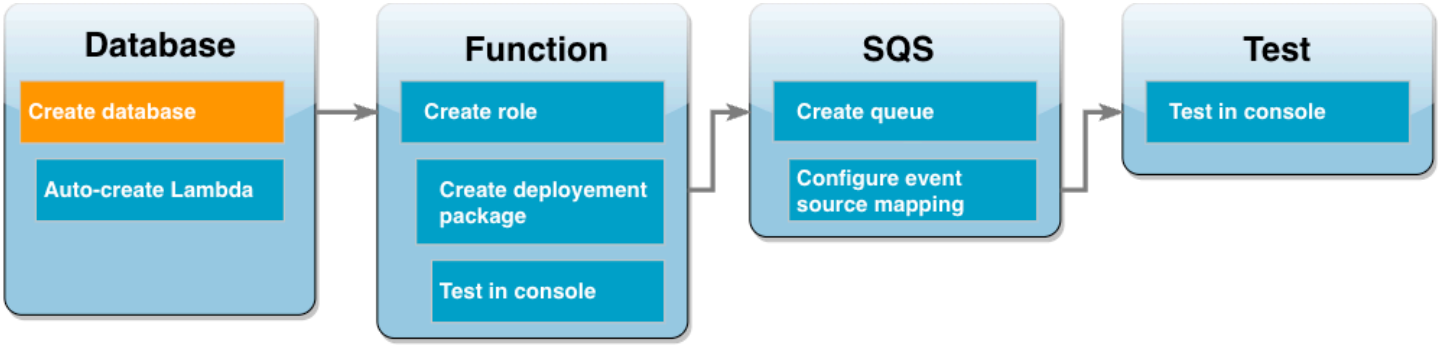
Puoi completare questo tutorial usando AWS Management Console o il AWS Command Line Interface ()AWS CLI.

Prerequisiti

Prima di iniziare, completa le fasi descritte in questa sezione:

- Registrati per un Account AWS
- Crea un utente con accesso amministrativo

Creazione di un'istanza database Amazon RDS



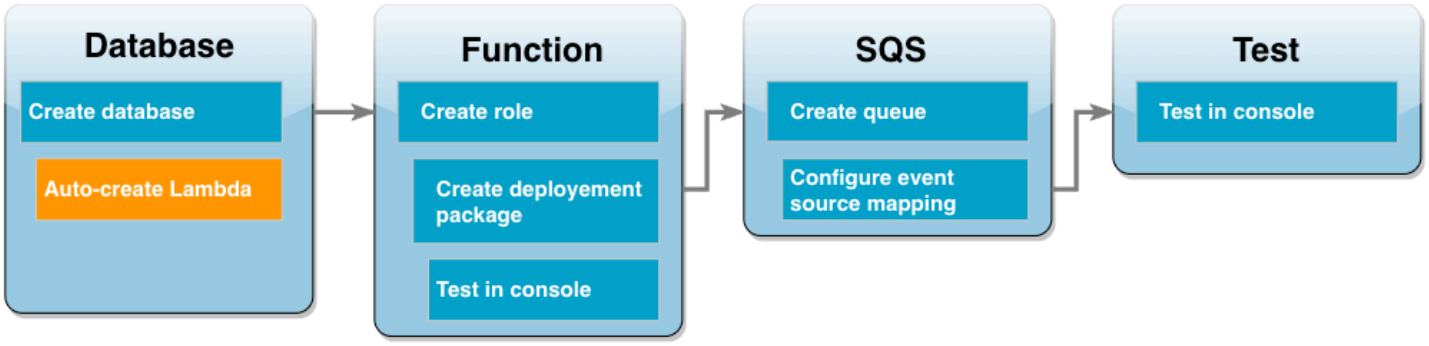
Un'istanza database Amazon RDS è un ambiente di database isolato in esecuzione nel Cloud AWS. Un'istanza può contenere uno o più database creati dall'utente. Se non diversamente specificato, Amazon RDS crea nuove istanze di database nel VPC predefinito incluso nel tuo. Account AWS Per ulteriori informazioni su Amazon VPC, consulta la Guida per l'utente di Amazon Virtual Private Cloud.

In questo tutorial, crei una nuova istanza nel tuo Account AWS VPC predefinito e crei un database denominato `ExampleDB` in quell'istanza. È possibile creare l'istanza DB e il database utilizzando il AWS Management Console o il AWS CLI.

Per creare un'istanza database

1. Apri la console Amazon RDS e scegli Crea database.
2. Lascia selezionata l'opzione Creazione standard, quindi in Opzioni del motore, scegli MySQL.
3. Nella sezione Modelli, seleziona Piano gratuito.
4. In Impostazioni, per Identificatore istanza database, immetti **`MySQLForLambda`**.
5. Per impostare il nome e la password, procedi come segue:
   a. In Impostazioni delle credenziali, lascia il campo Nome utente master impostato su `admin`.
   b. Per Password master, inserisci una password e confermala per accedere al database.
6. Specifica il nome del database effettuando le seguenti operazioni:
   - Lascia selezionate tutte le opzioni predefinite rimanenti e scorri verso il basso fino alla sezione Configurazione aggiuntiva.
   - Espandi questa sezione e immetti **`ExampleDB`** in Nome database iniziale.
7. Lascia selezionate tutte le opzioni predefinite rimanenti e scegli Crea database.

Creazione di una funzione Lambda e un proxy



È possibile utilizzare la console RDS per creare una funzione Lambda e un proxy nello stesso VPC del database.

> **Note**
>
> È possibile creare queste risorse associate solo al termine della creazione del database e quando si trova nello stato Disponibile.

Per creare una funzione e un proxy associati

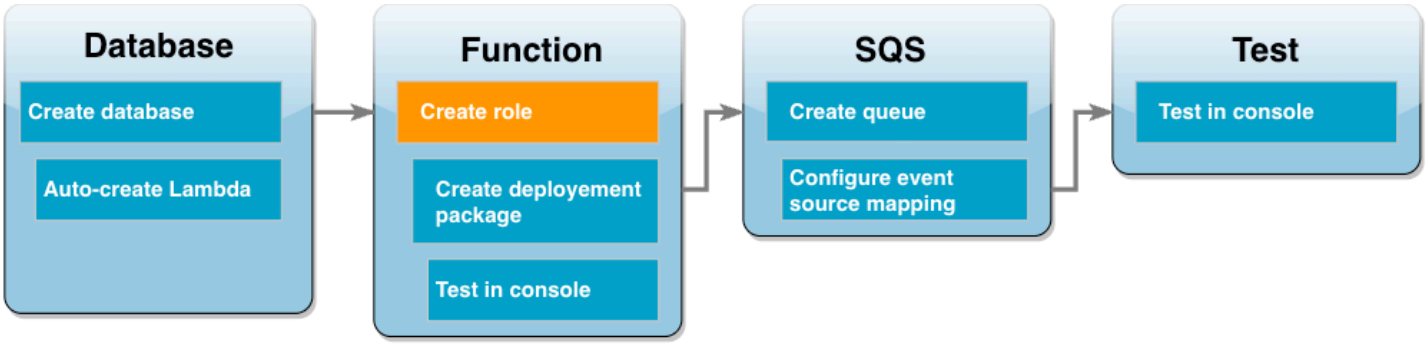
1. Dalla pagina Database, controlla se il database si trova nello stato Disponibile. In questo caso, passa alla fase successiva. Altrimenti, attendi che il database sia disponibile.
2. Seleziona il database e scegli Configurazione della connessione Lambda da Azioni.
3. Nella pagina Configurazione della connessione Lambda, scegli Crea una nuova funzione.

   Imposta il nuovo nome della funzione Lambda su **`LambdaFunctionWithRDS`**.
4. Nella sezione RDS Proxy, seleziona l'opzione Connetti tramite RDS Proxy. Scegli Crea nuovo proxy.

   - Per Credenziali del database, scegli nome utente e password del database.
   - Per Nome utente, specifica `admin`.
   - Per Password, immetti la password creata per l'istanza database.
5. Seleziona Configurare per completare la creazione del proxy e della funzione Lambda.

La procedura guidata completa la configurazione e fornisce un collegamento alla console Lambda per esaminare la nuova funzione. Prendi nota dell'endpoint proxy prima di passare alla console Lambda.

Creazione di un ruolo di esecuzione della funzione



Prima di creare la funzione Lambda, è necessario creare un ruolo di esecuzione per assegnare alla funzione le autorizzazioni necessarie. Per questo tutorial, Lambda richiede l'autorizzazione per gestire la connessione di rete al VPC contenente l'istanza database e per eseguire il polling dei messaggi da una coda Amazon SQS.

Per assegnare alla funzione Lambda le autorizzazioni necessarie, questo tutorial utilizza policy gestite da IAM. Si tratta di policy che concedono le autorizzazioni per molti casi d'uso e sono disponibili nel tuo Account AWS. Per ulteriori informazioni sull'uso delle policy gestite, consulta Best practice delle policy.

Creazione del ruolo di esecuzione di Lambda

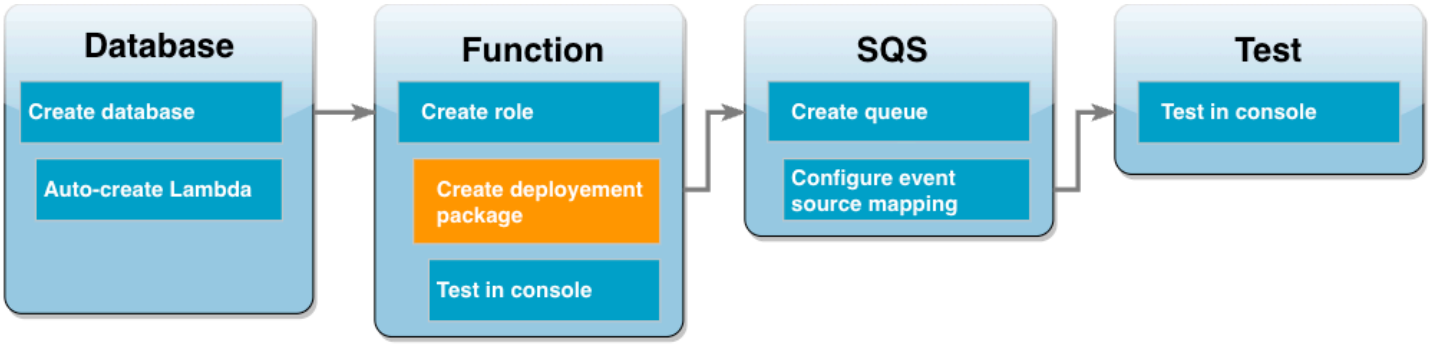
1. Apri la pagina Ruoli della console IAM, quindi scegli Crea ruolo.
2. Per Tipo di entità attendibile, scegli il servizio AWS e per Caso d'uso, scegli Lambda.
3. Seleziona Successivo.
4. Aggiungi le policy gestite da IAM effettuando le seguenti operazioni:
    a. Utilizzando la casella di ricerca delle policy, cerca **`AWSLambdaSQSQueueExecutionRole`**.
    b. Nell'elenco dei risultati, seleziona la casella di controllo accanto al ruolo, quindi scegli Cancella filtri.
    c. Utilizzando la casella di ricerca delle policy, cerca **`AWSLambdaVPCAccessExecutionRole`**.
    d. Nell'elenco dei risultati, seleziona la casella di controllo accanto al ruolo, quindi scegli Successivo.
5. In Nome ruolo, immetti **`lambda-vpc-sqs-role`** e quindi seleziona Crea ruolo.

Più avanti nel tutorial sarà necessario il nome della risorsa Amazon (ARN) del ruolo di esecuzione appena creato.

Individuazione dell'ARN del ruolo di esecuzione

1. Apri la pagina Ruoli della console IAM e scegli il ruolo (`lambda-vpc-sqs-role`).
2. Copia l'ARN visualizzato nella sezione Riepilogo.

Creazione di un pacchetto di implementazione Lambda



L'esempio seguente di codice Python utilizza il pacchetto PyMySQL per aprire una connessione al database. La prima volta che viene richiamata la funzione, crea anche una nuova tabella chiamata `Customer`. La tabella utilizza lo schema seguente, dove `CustID` è la chiave primaria:

```
Customer(CustID, Name)
```

La funzione utilizza anche PyMy SQL per aggiungere record a questa tabella. La funzione aggiunge i record utilizzando gli ID e i nomi dei clienti specificati nei messaggi che aggiungerai alla coda Amazon SQS.

Il codice crea la connessione al database al di fuori della funzione di gestione. La creazione della connessione nel codice di inizializzazione consente di riutilizzarla da chiamate della funzione successive e migliora le prestazioni. In un'applicazione di produzione, è inoltre possibile utilizzare la simultaneità fornita per inizializzare un numero richiesto di connessioni al database. Queste connessioni sono disponibili non appena viene richiamata la funzione.

```
import sys
import logging
import pymysql
import json
import os

# rds settings
user_name = os.environ['USER_NAME']
password = os.environ['PASSWORD']
rds_proxy_host = os.environ['RDS_PROXY_HOST']
db_name = os.environ['DB_NAME']

logger = logging.getLogger()
logger.setLevel(logging.INFO)
```

```
# create the database connection outside of the handler to allow connections to be
# re-used by subsequent function invocations.
try:
        conn = pymysql.connect(host=rds_proxy_host, user=user_name, passwd=password,
 db=db_name, connect_timeout=5)
except pymysql.MySQLError as e:
    logger.error("ERROR: Unexpected error: Could not connect to MySQL instance.")
    logger.error(e)
    sys.exit(1)

logger.info("SUCCESS: Connection to RDS for MySQL instance succeeded")

def lambda_handler(event, context):
    """
    This function creates a new RDS database table and writes records to it
    """
    message = event['Records'][0]['body']
    data = json.loads(message)
    CustID = data['CustID']
    Name = data['Name']

    item_count = 0
    sql_string = f"insert into Customer (CustID, Name) values(%s, %s)"

    with conn.cursor() as cur:
        cur.execute("create table if not exists Customer ( CustID  int NOT NULL, Name
 varchar(255) NOT NULL, PRIMARY KEY (CustID))")
        cur.execute(sql_string, (CustID, Name))
        conn.commit()
        cur.execute("select * from Customer")
        logger.info("The following items have been added to the database:")
        for row in cur:
            item_count += 1
            logger.info(row)
    conn.commit()

    return "Added %d items to RDS for MySQL table" %(item_count)
```

> **Note**
>
> In questo esempio, le credenziali di accesso al database vengono archiviate come variabili di ambiente. Nelle applicazioni di produzione, si consiglia di utilizzare AWS Secrets Manager per maggiore sicurezza. Tieni presente che, se la funzione Lambda si trova in un VPC, per eseguire la connessione a Secrets Manager devi creare un endpoint VPC. Per ulteriori informazioni, consulta la pagina How to connect to Secrets Manager service within a Virtual Private Cloud.

Per includere la dipendenza PyMy SQL nel codice della funzione, create un pacchetto di distribuzione.zip. I seguenti comandi funzionano per Linux, macOS o Unix:

Creazione di un pacchetto di implementazione .zip

1. Salva il codice di esempio come un file denominato `lambda_function.py`.
2. Nella stessa directory in cui hai creato il `lambda_function.py` file, crea una nuova directory denominata `package` e installa la libreria PyMy SQL.

   ```
   mkdir package
   pip install --target package pymysql
   ```

3. Create un file zip contenente il codice dell'applicazione e la libreria PyMy SQL. Su Linux o macOS, esegui i comandi della CLI riportati. Su Windows, usa il tuo strumento di compressione preferito per creare il file `lambda_function.zip`. Il file del codice sorgente `lambda_function.py` e le cartelle contenenti le dipendenze devono essere installati nella directory principale del file .zip.

   ```
   cd package
   zip -r ../lambda_function.zip .
   cd ..
   zip lambda_function.zip lambda_function.py
   ```

   Puoi creare il tuo pacchetto di implementazione anche utilizzando un ambiente virtuale Python. Consulta Distribuisci funzioni Lambda per Python con gli archivi di file .zip.

Aggiornamento della funzione Lambda

Utilizzando il nuovo pacchetto .zip creato, viene aggiornata una funzione Lambda tramite la console Lambda. Per consentire alla funzione di accedere al database, è inoltre necessario configurare le variabili di ambiente con le credenziali di accesso.

Per aggiornare la funzione Lambda

1. Apri la pagina Funzioni della console Lambda e scegli la tua funzione `LambdaFunctionWithRDS`.
2. Nella scheda Impostazioni di runtime, seleziona Modifica per modificare il Runtime della funzione in Python 3.10.
3. Cambia il Gestore in `lambda_function.lambda_handler`.
4. Nella scheda Codice, scegli Carica da, quindi File .zip.
5. Seleziona il file `lambda_function.zip` che hai creato nella fase precedente e scegli Salva.

A questo punto, configura la funzione con il ruolo di esecuzione creato in precedenza. Ciò concede alla funzione le autorizzazioni necessarie per accedere all'istanza del database ed eseguire il polling di una coda Amazon SQS.

Per configurare il ruolo di esecuzione della funzione

1. Nella pagina Funzioni della console Lambda, seleziona la scheda Configurazione, quindi scegli Autorizzazioni.
2. In Ruolo di esecuzione, scegli Modifica.
3. In Ruolo esistente, scegli il ruolo di esecuzione (`lambda-vpc-sqs-role`).
4. Selezionare Salva.

Per configurare le variabili di ambiente della funzione

1. Nella pagina Funzioni della console Lambda, seleziona la scheda Configurazione, quindi scegli Variabili di ambiente.
2. Scegli Modifica.
3. Per aggiungere le credenziali di accesso al database, procedi come segue:

a. Scegli Aggiungi variabili di ambiente, quindi in Chiave inserisci **`USER_NAME`** e in Valore inserisci **`admin`**.

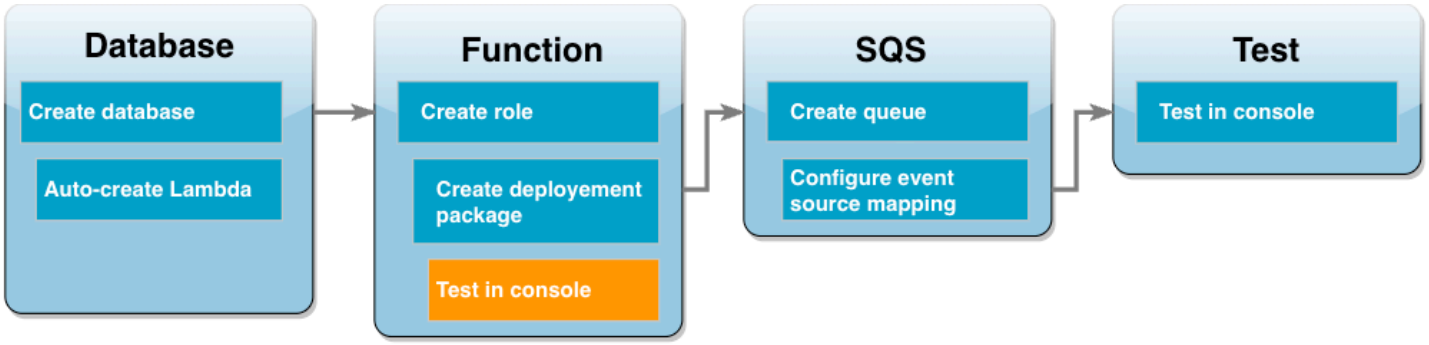
b. Scegli Aggiungi variabili di ambiente, quindi in Chiave inserisci **`DB_NAME`** e in Valore inserisci **`ExampleDB`**.

c. Scegli Aggiungi variabili di ambiente, quindi in Chiave inserisci **`PASSWORD`** e in Valore inserisci la password che hai scelto quando hai creato il database.

d. Scegli Aggiungi variabili di ambiente, quindi per Chiave inserisci **`RDS_PROXY_HOST`** e per Valore inserisci l'endpoint di RDS Proxy di cui hai preso nota in precedenza.

e. Selezionare Salva.

Test della funzione Lambda nella console



A questo punto è possibile utilizzare la console Lambda per testare la funzione. Viene creato un evento di test che imita i dati che verranno ricevuti dalla tua funzione quando questa viene richiamata utilizzando Amazon SQS nella fase finale del tutorial. L'evento di test contiene un oggetto JSON che specifica un ID cliente e un nome del cliente da aggiungere alla tabella `Customer` creata dalla funzione.

Verifica della funzione Lambda

1. Apri la pagina Funzioni della console Lambda e scegli la tua funzione.
2. Scegli la sezione Test.
3. Scegli Crea nuovo evento e immetti **`myTestEvent`** per il nome dell'evento.
4. Copia il seguente codice in JSON dell'evento e scegli Salva.

```
{
  "Records": [
```

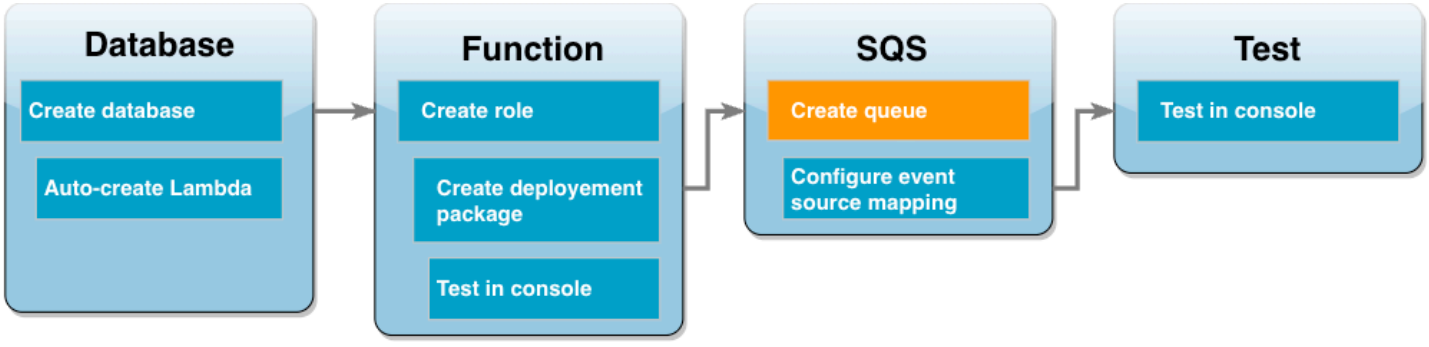
```
    {
      "messageId": "059f36b4-87a3-44ab-83d2-661975830a7d",
      "receiptHandle": "AQEBwJnKyrHigUMZj6rYigCgxlaS3SLy0a...",
      "body": "{\n     \"CustID\": 1021,\n     \"Name\": \"Martha Rivera\"\n}",
      "attributes": {
        "ApproximateReceiveCount": "1",
        "SentTimestamp": "1545082649183",
        "SenderId": "AIDAIENQZJOLO23YVJ4VO",
        "ApproximateFirstReceiveTimestamp": "1545082649185"
      },
      "messageAttributes": {},
      "md5OfBody": "e4e68fb7bd0e697a0ae8f1bb342846b3",
      "eventSource": "aws:sqs",
      "eventSourceARN": "arn:aws:sqs:us-west-2:123456789012:my-queue",
      "awsRegion": "us-west-2"
    }
  ]
}
```

5. Scegli Test (Esegui test).

Nella scheda Risultati di esecuzione, si dovrebbero ottenere risultati simili ai seguenti visualizzati nei log della funzione:

```
[INFO] 2023-02-14T19:31:35.149Z bdd06682-00c7-4d6f-9abb-89f4bbb4a27f The following
 items have been added to the database:
[INFO] 2023-02-14T19:31:35.149Z bdd06682-00c7-4d6f-9abb-89f4bbb4a27f (1021, 'Martha
 Rivera')
```

Creazione di una coda Amazon SQS

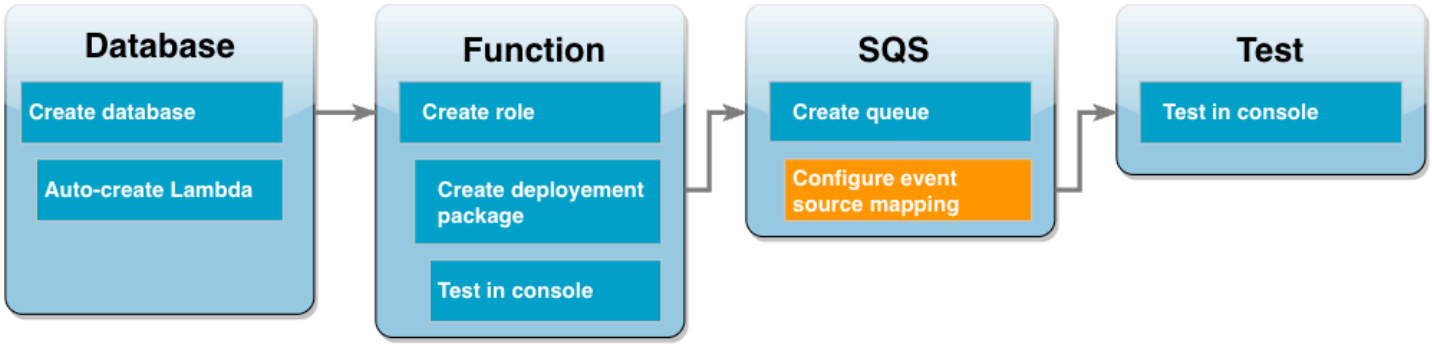


Hai testato con successo l'integrazione della tua funzione Lambda e dell'istanza del database Amazon RDS. Adesso creerai la coda Amazon SQS che verrà utilizzata per richiamare la funzione Lambda nella fase finale del tutorial.

Creazione della coda Amazon SQS (console)

1. Apri la pagina Code della console Amazon SQS e seleziona Crea coda.
2. Lascia il campo Tipo impostato su Standard e inserisci **`LambdaRDSQueue`** per il nome della coda.
3. Lascia selezionate tutte le opzioni predefinite e scegli Crea coda.

Creazione di uno strumento di mappatura dell'origine degli eventi per richiamare la funzione Lambda



Uno strumento di mappatura dell'origine degli eventi è una risorsa Lambda che legge gli elementi da un flusso o da una coda e chiama una funzione Lambda. Quando si configura uno strumento di mappatura dell'origine degli eventi, puoi specificare una dimensione batch in modo che i record del flusso o della coda vengano raggruppati in un unico payload. In questo esempio, la dimensione del batch viene impostata su 1 in modo che la funzione Lambda venga richiamata ogni volta che viene inviato un messaggio alla coda. È possibile configurare la mappatura delle sorgenti degli eventi utilizzando la console AWS CLI o Lambda.

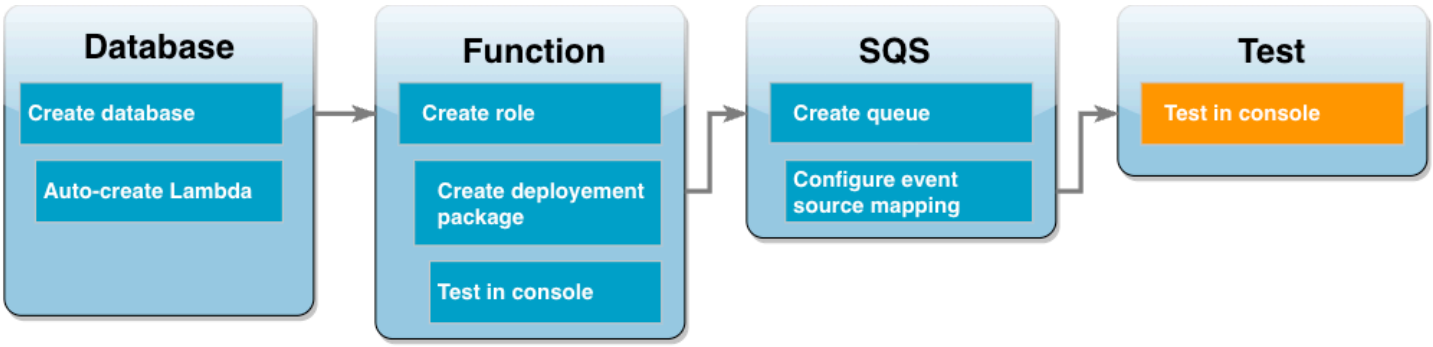
Creazione di uno strumento di mappatura dell'origine degli eventi (console)

1. Apri la pagina Funzioni della console Lambda e scegli la tua funzione (`LambdaFunctionWithRDS`).
2. Nella sezione Panoramica della funzione, scegli Aggiungi trigger.
3. Per l'origine, seleziona Amazon SQS, quindi seleziona il nome della coda (`LambdaRDSQueue`).

4. Per Dimensioni del batch, immetti **1**.
5. Lascia tutte le altre opzioni impostate sui valori predefiniti e scegli Aggiungi.

A questo punto, è possibile testare la configurazione completa aggiungendo un messaggio alla coda Amazon SQS.

Test e monitoraggio della configurazione



Per testare la configurazione completa, aggiungi i messaggi alla coda Amazon SQS utilizzando la console. Utilizzate quindi CloudWatch Logs per confermare che la funzione Lambda sta scrivendo record nel database come previsto.

Test e monitoraggio della configurazione

1. Apri la pagina Code della console Amazon SQS e seleziona la coda (`LambdaRDSQueue`).
2. Scegli Invio e ricezione di messaggi e incolla il seguente JSON nel Corpo del messaggio nella sezione Invia messaggio.

```
{
    "CustID": 1054,
    "Name": "Richard Roe"
}
```

3. Scegliere Invia messaggio.

   L'invio del messaggio alla coda farà sì che Lambda richiami la funzione tramite lo strumento di mappatura dell'origine degli eventi. Per confermare che Lambda abbia richiamato la funzione come previsto, usa CloudWatch Logs per verificare che la funzione abbia scritto il nome e l'ID del cliente nella tabella del database.

4. Apri la pagina Log groups della CloudWatch console e seleziona il gruppo di log per la tua funzione (). `/aws/lambda/LambdaFunctionWithRDS`
5. Nella sezione Flussi di log, scegli il flusso di log più recente.

   La tabella deve contenere due record relativi ai clienti, uno per ogni chiamata della funzione. Nel flusso di log, si dovrebbero visualizzare messaggi simili ai seguenti:

```
[INFO] 2023-02-14T19:06:43.873Z 45368126-3eee-47f7-88ca-3086ae6d3a77 The following
 items have been added to the database:
[INFO] 2023-02-14T19:06:43.873Z 45368126-3eee-47f7-88ca-3086ae6d3a77 (1021, 'Martha
 Rivera')
[INFO] 2023-02-14T19:06:43.873Z 45368126-3eee-47f7-88ca-3086ae6d3a77 (1054,
 'Richard Roe')
```

Pulizia delle risorse

Ora è possibile eliminare le risorse create per questo tutorial, a meno che non si voglia conservarle. Eliminando AWS le risorse che non utilizzi più, eviti addebiti inutili sul tuo AWS account.

Per eliminare la funzione Lambda

1. Aprire la pagina Functions (Funzioni) della console Lambda.
2. Selezionare la funzione creata.
3. Scegliere Actions (Operazioni), Delete (Elimina).
4. Scegliere Delete (Elimina).

Per eliminare il ruolo di esecuzione

1. Aprire la pagina Ruoli della console IAM.
2. Selezionare il ruolo di esecuzione creato.
3. Scegliere Delete role (Elimina ruolo).
4. Scegliere Yes, delete (Sì, elimina).

Per eliminare l'istanza database MySQL

1. Aprire la pagina Database della console Amazon RDS.

2. Selezionare il database creato.
3. Scegli Operazioni > Elimina.
4. Deselezionare la casella per Create final snapshot (Crea snapshot finale).
5. Immettere **`delete me`** nella casella di testo.
6. Scegliere Delete (Elimina).

Per eliminare la coda Amazon SQS

1. Accedi AWS Management Console e apri la console Amazon SQS all'indirizzo https://console.aws.amazon.com/sqs/.
2. Selezionare la coda creata.
3. Scegliere Delete (Elimina).
4. Immettere **`delete`** nella casella di testo.
5. Scegli Delete (Elimina).

Tutorial di Amazon RDS e codice di esempio

La AWS documentazione include diversi tutorial che ti guidano attraverso i casi d'uso più comuni di Amazon RDS Aurora. Molti di questi tutorial mostrano come usare Amazon RDS con altri servizi. AWS Inoltre, puoi accedere al codice di esempio in. GitHub

> **Note**
>
> Puoi trovare altri tutorial nel Blog di AWS Database. Per ulteriori informazioni sulla formazione, consulta AWS Training and Certification.

Argomenti

Tutorial in questa guida

I seguenti tutorial mostrano come eseguire le attività comuni con Amazon RDS:

- Tutorial: Creazione di un Amazon VPC da utilizzare con un'istanza database (solo IPv4)

  Scopri come includere un'istanza database in un cloud privato virtuale (VPC) basato sul servizio Amazon VPC. In questo caso, il VPC condivide i dati con un server Web in esecuzione su un'istanza Amazon EC2 nello stesso VPC.
- Tutorial: Creazione di un VPC per l'utilizzo con un'istanza database (modalità dual-stack)

  Scopri come includere un'istanza database in un cloud privato virtuale (VPC) basato sul servizio Amazon VPC. In questo caso, il VPC condivide i dati con un'istanza Amazon EC2 nello stesso VPC. In questo tutorial crei il VPC per questo scenario che funziona con un database in esecuzione in modalità dual-stack.

- Tutorial: creazione di un server Web e un'istanza database Amazon RDS

  Questo tutorial consente di installare un server Web Apache con PHP e creare un database MySQL. Il server web viene eseguito in un'istanza Amazon EC2 utilizzando Amazon Linux e il database MySQL è un'istanza database MySQL. Sia l'istanza Amazon EC2 che l'istanza sono eseguiti in un Amazon VPC.
- Tutorial: ripristino di un'istanza database Amazon RDS da uno snapshot DB

  Scopri come ripristinare un'istanza database da uno snapshot DB.
- Tutorial: utilizzo di una funzione Lambda per accedere a un database Amazon RDS

  Ulteriori informazioni su come creare una funzione Lambda dalla console RDS per accedere a un database tramite un proxy, creare una tabella, aggiungere alcuni record e recuperare i record dalla tabella. Imparerai anche come richiamare la funzione Lambda e verificare i risultati della query.
- Tutorial: Utilizzo dei tag per specificare le istanze database da interrompere

  Scopri come utilizzare i tag per specificare le istanze database da interrompere.
- Tutorial: registra le modifiche allo stato delle istanze DB utilizzando Amazon EventBridge

  Scopri come registrare una modifica dello stato di un'istanza DB utilizzando Amazon EventBridge e AWS Lambda.
- Tutorial: creazione di un allarme Amazon CloudWatch per il ritardo di replica del cluster di database Multi-AZ

  Scopri come creare un CloudWatch allarme che invii un messaggio Amazon SNS quando il ritardo di replica per un cluster DB Multi-AZ ha superato una soglia. Un allarme monitora il parametro `ReplicaLag` per il periodo di tempo specificato. L'operazione corrisponde all'invio di una notifica a un argomento Amazon SNS o a una policy Amazon EC2 Auto Scaling.

Tutorial in altre guide AWS

- Tutorial: Rotazione di un segreto per un AWS database nella guida per l'utente AWS Secrets Manager

  Scopri come creare un segreto per un AWS database e configurare il segreto in modo che ruoti secondo una pianificazione. Attivi una rotazione manualmente e confermi che la nuova versione del segreto continua a fornire l'accesso.

- Tutorial ed esempi nella Guida per gli sviluppatori di AWS Elastic Beanstalk

  Scopri come distribuire applicazioni che utilizzano database Amazon RDS con. AWS Elastic Beanstalk

- Utilizzo dei dati da un database Amazon RDS per creare un'origine dati Amazon ML nella Amazon Machine Learning Developer Guide

  Scopri come creare un oggetto dell'origine dati Amazon Machine Learning (Amazon ML) dai dati memorizzati in un'istanza database MySQL.

- Abilitazione manuale dell'accesso a un'istanza Amazon RDS in un VPC nella Amazon QuickSight User Guide

  Scopri come abilitare QuickSight l'accesso di Amazon a un'istanza database Amazon RDS in un VPC.

AWS portale di contenuti per workshop e laboratori per Amazon RDS Amazon Aurora

La seguente raccolta di workshop e altri contenuti pratici ti aiuta a comprendere le caratteristiche e le funzionalità di Amazon RDS PostgreSQL:

- Creazione di un'istanza database

  Informazioni su come creare l'istanza database.

- Monitoraggio delle prestazioni con strumenti RDS

  Scopri come utilizzare AWS gli strumenti SQL (Cloudwatch, Enhanced Monitoring, Slow Query Logs, Performance Insights, PostgreSQL Catalog Views) per comprendere i problemi di prestazioni e identificare modi per migliorare le prestazioni del tuo database.

AWS portale di contenuti per workshop e laboratori per Amazon RDS Amazon Aurora

La seguente raccolta di workshop e altri contenuti pratici ti aiuta a comprendere le caratteristiche e le funzionalità di Amazon RDS MySQL:

- Creazione di un'istanza database

  Informazioni su come creare l'istanza database.

- Uso di Approfondimenti sulle prestazioni

  Informazioni su come monitorare e ottimizzare la tua istanza DB tramite Approfondimenti sulle prestazioni.

Tutorial ed esempi di codice in GitHub

- Creazione del tracciatore di elementi Amazon Relational Database Service

  Scopri come creare un'applicazione che tiene traccia e segnala gli elementi di lavoro. Questa applicazione utilizza Amazon RDS, Amazon Simple Email Service, Elastic Beanstalk e SDK per Java 2.x.

Utilizzo di questo servizio con un AWS SDK

AWS I kit di sviluppo software (SDK) sono disponibili per molti linguaggi di programmazione più diffusi. Ogni SDK fornisce un'API, esempi di codice, e documentazione che facilitano agli sviluppatori la creazione di applicazioni nel loro linguaggio preferito.

Documentazione sugli SDK	Esempi di codice
AWS SDK for C++	AWS SDK for C++ esempi di codice
AWS CLI	AWS CLI esempi di codice
AWS SDK for Go	AWS SDK for Go esempi di codice
AWS SDK for Java	AWS SDK for Java esempi di codice
AWS SDK for JavaScript	AWS SDK for JavaScript esempi di codice
SDK AWS for Kotlin	SDK AWS for Kotlin esempi di codice
AWS SDK for .NET	AWS SDK for .NET esempi di codice

Documentazione sugli SDK	Esempi di codice
AWS SDK for PHP	AWS SDK for PHP esempi di codice
AWS Tools for PowerShell	Strumenti per esempi di PowerShell codice
AWS SDK for Python (Boto3)	AWS SDK for Python (Boto3) esempi di codice
AWS SDK for Ruby	AWS SDK for Ruby esempi di codice
AWS SDK for Rust	AWS SDK for Rust esempi di codice
SDK AWS per SAP ABAP	SDK AWS per SAP ABAP esempi di codice
SDK AWS per Swift	SDK AWS per Swift esempi di codice

Per esempi specifici del servizio, consulta Esempi di codice per Amazon RDS con SDK AWS.

Esempio di disponibilità

Non riesci a trovare quello che ti serve? Richiedi un esempio di codice utilizzando il link Provide feedback (Fornisci un feedback) nella parte inferiore di questa pagina.

Best practice per Amazon RDS

Scopri le best practice per l'utilizzo di Amazon RDS. Man mano che vengono identificate nuove best practice, aggiorneremo questa sezione.

Argomenti

> **Note**
>
> Per suggerimenti comuni per Amazon RDS, consulta Visualizzazione e risposta ai consigli di RDS.

Linee guida operative di base per Amazon RDS

Di seguito sono illustrate le linee guida operative di base e le best practice che tutti dovrebbero seguire durante l'utilizzo di Amazon RDS. Tieni presente che il contratto sul livello di servizio di Amazon RDS richiede che tu segua queste linee guida:

- Usa i parametri per monitorare l'utilizzo della memoria, della CPU, del ritardo di replica e dello storage. Puoi configurare Amazon in modo che ti CloudWatch avvisi quando i modelli di utilizzo cambiano o quando la tua distribuzione si avvicina ai limiti di capacità. Ciò consente di mantenere le prestazioni e la disponibilità del sistema.
- Incrementa la capacità dell'istanza database quando stai per raggiungere i limiti della capacità di storage. Avrai bisogno di memoria e storage aggiuntivi per soddisfare aumenti imprevisti della domanda delle tue applicazioni.
- Abilita i backup automatici e configurane l'esecuzione nel momento della giornata in cui il carico di operazioni di scrittura I/O al secondo è inferiore. Questo quando un backup è meno dannoso per l'utilizzo del database.
- Se il carico di lavoro del database richiede più operazioni di I/O rispetto a quelle che hai assegnato, il ripristino dopo un failover o un errore del database sarà lento. Per aumentare la capacità I/O di un'istanza di database, effettua una o più delle seguenti operazioni:
  - Effettua la migrazione a una classe di istanza database con elevata capacità di I/O.
  - Converti lo storage magnetico in storage General Purpose o Provisioned IOPS, in base all'incremento di cui hai bisogno. Per informazioni sui tipi di storage disponibili, consulta Tipi di storage Amazon RDS.

    Se effettui la conversione allo storage Provisioned IOPS, assicurati di utilizzare anche una classe di istanza di database ottimizzata per le opzioni Provisioned IOPS. Per ulteriori informazioni sull'opzione Provisioned IOPS, consulta Storage SSD Provisioned IOPS.
  - Se utilizzi già lo storage Provisioned IOPS, assegna capacità di throughput ulteriore.
- Se l'applicazione client memorizza nella cache i dati DNS (Domain Name Service) delle istanze DB, imposta un valore time-to-live (TTL) inferiore a 30 secondi. L'indirizzo IP sottostante di un'istanza DB può cambiare dopo un failover. La memorizzazione nella cache dei dati DNS per un periodo prolungato può causare errori di connessione. L'applicazione potrebbe tentare di connettersi a un indirizzo IP non più in uso.
- Prova il failover per l'istanza database per capire quanto tempo impiega il processo per il caso d'uso particolare. Inoltre, garantisci che l'applicazione che accede all'istanza database è in grado di connettersi automaticamente alla nuova istanza database dopo il failover.

Suggerimenti relativi alla RAM per un'istanza di database

Una best practice per le prestazioni di Amazon RDS consiste nell'allocare RAM sufficiente in modo che il working set risieda quasi interamente nella memoria. Il working set è formato dai dati e dagli indici che vengono utilizzati spesso nell'istanza. Più utilizzi l'istanza DB, più il working set crescerà.

Per sapere se il tuo set di lavoro è quasi tutto in memoria, controlla la metrica ReadiOps (usando CloudWatch Amazon) mentre l'istanza DB è sotto carico. Il valore ReadIOPS dovrebbe essere di piccola entità e stabile. In alcuni casi, il passaggio della classe di istanza database a una classe con più RAM provoca un drastico calo di ReadIOPS. In questi casi, il set di lavoro non è completamente in memoria. Continua con l'incremento fino a quando il valore di ReadIOPS non diminuisce più drasticamente dopo un'operazione di dimensionamento o è di entità molto piccola. Per ulteriori informazioni sul monitoraggio dei parametri di un'istanza di database, consulta Visualizzazione dei parametri nella console Amazon RDS.

AWS driver di database

Consigliamo la AWS suite di driver per la connettività delle applicazioni. I driver sono stati progettati per fornire supporto per tempi di switchover e failover più rapidi e per l'autenticazione con AWS Secrets Manager, AWS Identity and Access Management (IAM) e Federated Identity. I AWS driver si basano sul monitoraggio dello stato dell'istanza DB e sulla conoscenza della topologia dell'istanza per determinare il nuovo writer. Questo approccio riduce i tempi di switchover e failover a secondi a una cifra, rispetto alle decine di secondi dei driver open source.

Con l'introduzione di nuove funzionalità di servizio, l'obiettivo della AWS suite di driver è disporre di un supporto integrato per queste funzionalità di servizio.

Per ulteriori informazioni, consulta Connessione alle istanze DB con i driver AWS.

Utilizzo del monitoraggio avanzato per identificare problemi del sistema operativo

Quando il monitoraggio avanzato è abilitato, Amazon RDS fornisce parametri in tempo reale per il sistema operativo (OS) su cui viene eseguita l'istanza database. È possibile visualizzare le metriche per l'istanza DB utilizzando la console. Puoi anche utilizzare l'output JSON di Enhanced Monitoring di Amazon CloudWatch Logs in un sistema di monitoraggio a tua scelta. Per ulteriori informazioni su Enhanced Monitoring, consult Monitoraggio dei parametri del sistema operativo con il monitoraggio avanzato.

Utilizzo di parametri per identificare problemi a livello di prestazioni

Per identificare problemi a livello di prestazioni causati da risorse insufficienti e altri colli di bottiglia comuni, puoi monitorare i parametri disponibili per l'istanza di database Amazon RDS.

Visualizzazione dei parametri relativi alle prestazioni

Dovresti monitorare regolarmente i parametri relativi alle prestazione per osservare i valori medi, massimi e minimi per vari intervalli di tempo. Ciò ti consente di identificare quando le prestazioni subiscono un calo. Puoi anche impostare CloudWatch allarmi Amazon per determinate soglie metriche in modo da essere avvisato se vengono raggiunte.

Per risolvere i problemi relativi alle prestazioni, è importante comprendere le prestazioni di base del sistema. Quando configuri un'istanza database e la esegui con un carico di lavoro tipico, acquisisci i valori medi, massimi e minimi di tutte le metriche delle prestazioni. Puoi farlo a diversi intervalli (ad esempio, un'ora, 24 ore, una settimana, due settimane) e ti permette di avere un quadro dei valori normali. Ciò aiuta anche a effettuare confronti delle attività durante le ore di punta e non di punta. Puoi quindi utilizzare queste informazioni per identificare quando le prestazioni scendono al di sotto dei livelli standard.

Se utilizzi cluster database multi-AZ, puoi monitorare la differenza di tempo tra l'ultima transazione sull'istanza database di scrittura e l'ultima transazione applicata su un'istanza database di lettura. Questa differenza è chiamata ritardo di replica. Per ulteriori informazioni, consulta Ritardo di replica e cluster di database Multi-AZ.

Puoi visualizzare la combinazione di Performance Insights e CloudWatch metriche nella dashboard di Performance Insights e monitorare la tua istanza DB. Per utilizzare questa visualizzazione di monitoraggio, Performance Insights deve essere attivato per l'istanza database specifica. Per ulteriori informazioni su questa visualizzazione di monitoraggio, consulta Visualizzazione delle metriche combinate nella console Amazon RDS.

È possibile creare un report di analisi delle prestazioni per un periodo di tempo specifico e visualizzare le informazioni dettagliate identificate e i suggerimenti per risolvere i problemi. Per ulteriori informazioni, consulta Creazione di un report di analisi delle prestazioni.

Per visualizzare i parametri relativi alle prestazioni

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.

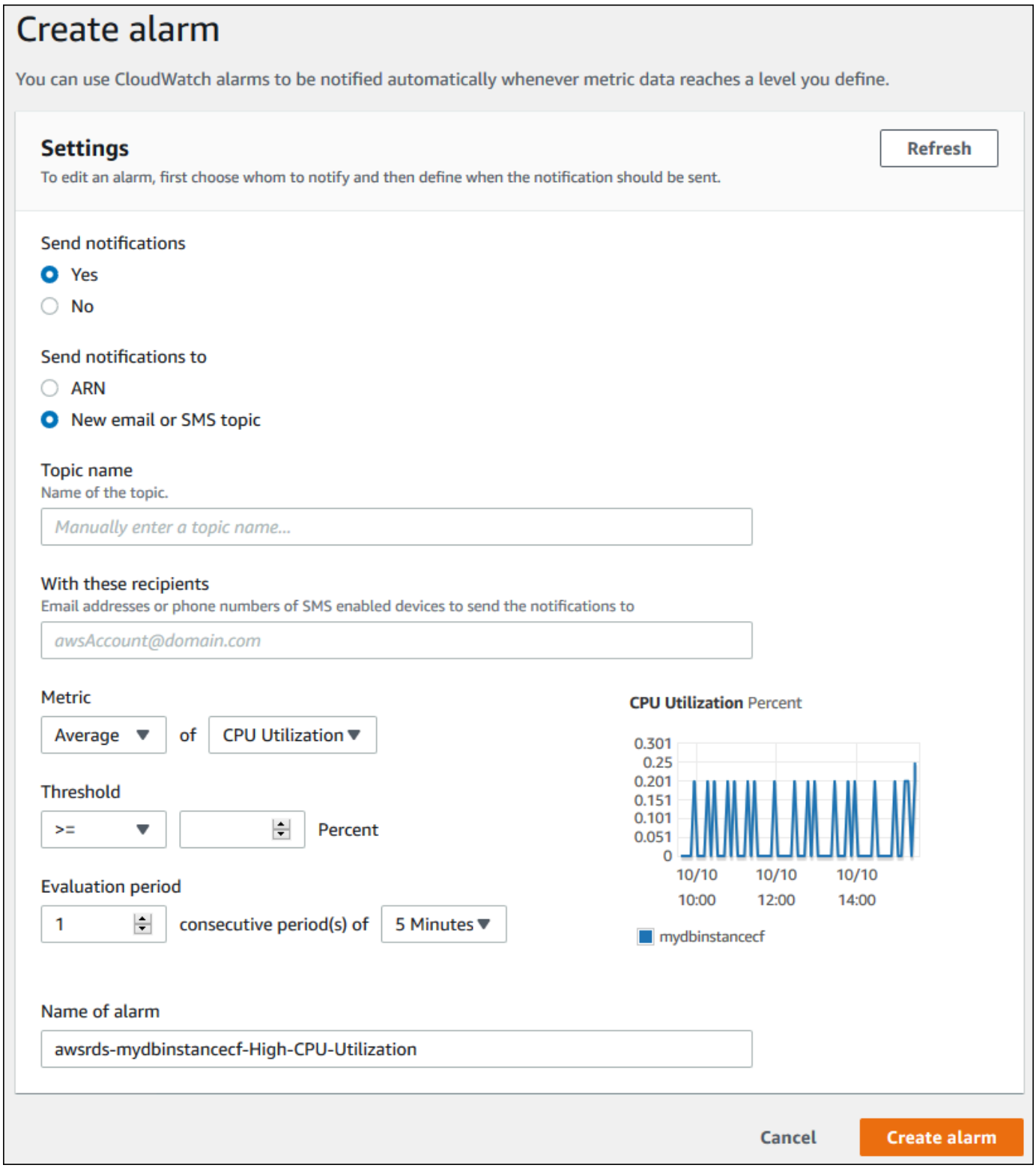
2. Nel riquadro di navigazione, scegliere Databases (Database), quindi scegliere un'istanza database.
3. Selezionare Monitoring (Monitoraggio).

   Il pannello di controllo fornisce le metriche sulle prestazioni. L'impostazione predefinita delle metriche consente di visualizzare le informazioni relative alle ultime tre ore.
4. Utilizzare i pulsanti numerati in alto a destra per sfogliare le metriche aggiuntive o modificare le impostazioni per visualizzare altre metriche.
5. Scegliere un parametro relativo alle prestazioni per regolare l'intervallo di tempo per la visualizzazione dei dati per i giorni diversi da quello corrente. È possibile modificare i valori dei campi Statistic (Statistica), Time Range (Intervallo di tempo) e Period (Periodo) in base alle informazioni che si desidera visualizzare. Ad esempio, potresti voler visualizzare i valori di picco di un parametro per ogni giorno nelle ultime due settimane. In tal caso, imposta Statistic (Statistiche) su Maximum (Massimo), Time Range (Intervallo di tempo) su Last 2 Weeks (Ultime 2 settimane) e Period (Periodo) su Day (Giorno).

È possibile anche visualizzare i parametri relativi alle prestazioni mediante CLI o API. Per ulteriori informazioni, consulta Visualizzazione dei parametri nella console Amazon RDS.

Per impostare una sveglia CloudWatch

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, scegliere Databases (Database), quindi scegliere un'istanza database.
3. Scegliere Logs & events (Log ed eventi).
4. Nella sezione CloudWatch Allarmi, scegli Crea allarme.



5. In Send notifications (Invia notifiche) scegliere Yes (Sì), mentre in Send notifications to (Invia notifiche a) seleziona New email or SMS topic (Nuovo argomento e-mail o SMS).

6. In Topic name (Nome argomento) digitare un nome per la notifica e in With these recipients (Con questi destinatari) digitare un elenco separato da virgole con gli indirizzi e-mail e i numeri di telefono.
7. In Metric (Parametro) scegliere la statistica e il parametro dell'allarme da impostare.
8. In Threshold (Soglia) specificare se il parametro deve essere maggiore, minore o uguale alla soglia e specificare il valore di soglia.
9. In Evaluation period (Periodo di valutazione), scegli il periodo di valutazione per l'allarme. In consecutive period(s) of (periodo/i consecutivo/i di) scegli il periodo durante il quale si deve raggiungere la soglia per attivare l'allarme.
10. Per Nome dell'allarme, inserire un nome per l'allarme.
11. Scegli Crea Alarm (Crea allarme).

L'allarme appare nella sezione CloudWatch Allarmi.

Valutazione dei parametri relativi alle prestazioni

Un'istanza di database include diverse categorie di parametri e il modo in cui stabilire valori accettabili dipende dal parametro.

CPU

- Utilizzo della CPU; percentuale di capacità di elaborazione del computer utilizzata.

Memoria

- Memoria liberabile: quanta RAM è disponibile sull'istanza DB, in byte. Nella scheda Monitoring (Monitoraggio), la linea rossa indica un livello del 75% per i parametri CPU, Memory (Memoria) e Storage. Se il consumo di memoria dell'istanza supera regolarmente questa linea, significa che è necessario controllare il carico di lavoro o aggiornare l'istanza.
- Utilizzo dello swap: la quantità di spazio di swap utilizzata dall'istanza DB, in byte.

Spazio su disco

- Spazio di storage libero: quantità di spazio su disco non attualmente utilizzato dall'istanza database, in megabyte.

Operazioni di input/output

- IOPS di lettura, IOPS di scrittura: il numero medio di operazioni di scrittura o lettura su disco al secondo.
- Latenza di lettura, Latenza di scrittura: il tempo medio per un'operazione di lettura o scrittura, in millisecondi.
- Throughput di lettura, Throughput di scrittura: il numero medio di megabyte letti dal e scritti sul disco al secondo.
- Profondità coda: il numero di operazioni I/O in attesa di essere scritte sul o lette dal disco.

Traffico di rete

- Throughput di ricezione di rete, Throughput di trasmissione di rete – La velocità del traffico di rete verso e dall'istanza database in megabyte al secondo.

Connessioni database

- Connessioni DB: il numero di sessioni client connesse all'istanza database.

Per descrizioni individuali più dettagliate dei parametri disponibili relativi alle prestazioni, consultare Monitoraggio dei parametri di Amazon RDS con Amazon CloudWatch.

In generale, i valori accettabili per i parametri relativi alle prestazioni dipendono dalla baseline e dall'attività dell'applicazione. Indagare le variazioni della baseline coerenti o che rappresentano dei trend. I seguenti sono alcuni suggerimenti su tipi di parametri specifici:

- High CPU or RAM consumption (Consumo elevato di CPU o RAM): i valori elevati per il consumo di CPU o RAM potrebbero essere appropriati. Ad esempio, se sono in linea con gli obiettivi dell'applicazione (come velocità di trasmissione effettiva o simultaneità) e sono previsti.
- Consumo dello spazio su disco: esamina il consumo dello spazio su disco se lo spazio usato supera costantemente l'85% dello spazio su disco totale. Verifica se è possibile eliminare dati dall'istanza o archiviare dati su un sistema diverso per liberare spazio.
- Traffico di rete – Per il traffico di rete, rivolgiti al tuo amministratore di sistema per identificare il throughput previsto per la rete del dominio e la connessione Internet. Indaga il traffico di rete se il throughput è costantemente al di sotto del valore previsto.

- Connessioni al database: valuta se limitare le connessioni al database se noti un numero elevato di connessioni utente e contemporaneamente un peggioramento delle prestazioni e del tempo di risposta delle istanze. Il numero ideale di connessioni utente per l'istanza di database dipende dalla classe di istanza e dalla complessità delle operazioni eseguite. Per determinare il numero di connessioni di database, associa l'istanza database a un gruppo di parametri. In questo gruppo, imposta il parametro User Connections (Connessioni utente) su un valore diverso da 0 (illimitate). Puoi utilizzare un gruppo di parametri esistente o crearne uno nuovo. Per ulteriori informazioni, consulta Utilizzo di gruppi di parametri.
- Parametri di IOPS: poiché i valori previsti per i parametri di IOPS dipendono dalle specifiche del disco e dalla configurazione del server, usa i valori di riferimento per identificare i comportamenti tipici. Verifica se i valori sono costantemente diversi dalla baseline. Per prestazioni IOPS ottimali, assicurati che il working set tipico possa essere caricato nella memoria per ridurre al minimo le operazioni di lettura e scrittura.

In caso di problemi relativi alle metriche delle prestazioni, puoi ottimizzare le query più utilizzate e più costose per migliorare le prestazioni. Ottimizzale per vedere se così facendo riduci la pressione sulle risorse di sistema. Per ulteriori informazioni, consulta Ottimizzazione di query.

Se le query sono state ottimizzate e il problema persiste, valuta la possibilità di eseguire l'upgrade delle Classi di istanze database Amazon RDS. Puoi passare a una classe con una quantità maggiore della risorsa correlata al problema (CPU, RAM, spazio su disco, larghezza di banda della rete, capacità I/O).

Ottimizzazione di query

Uno dei modi migliori per migliorare le prestazioni di un'istanza database consiste nell'ottimizzare le query più comuni e a uso più intensivo di risorse. Puoi ottimizzarle per renderle meno costose da eseguire. Per informazioni sul miglioramento delle query, utilizzare le risorse seguenti:

- MySQL – Consulta Optimizing SELECT statements (Ottimizzazione delle istruzioni SELECT) nella documentazione MySQL. Per ulteriori risorse di ottimizzazione delle query, consulta MySQL performance tuning and optimization resources (Risorse di ottimizzazione delle prestazioni di MySQL).
- Oracle – Consulta Database SQL Tuning Guide (Guida all'ottimizzazione di database SQL) nella documentazione Oracle.
- SQL Server – Consulta Analyzing a query (Analisi di una query) nella documentazione di Microsoft. È inoltre possibile utilizzare le viste DMV (Data Management Views) relative all'esecuzione,

all'indice e all'I/O descritte in System Dynamic Management Views (Viste a gestione dinamica di sistema) nella documentazione di Microsoft per risolvere i problemi relativi alle query di SQL Server.

Un aspetto comune dell'ottimizzazione delle query è la creazione di indici efficaci. Per potenziali miglioramenti dell'indice per l'istanza database, consulta Database Engine Tuning Advisor (Ottimizzazione guidata al motore di database) nella documentazione di Microsoft. Per informazioni sull'utilizzo di Tuning Advisor su RDS per SQL Server, consulta Analisi del carico di lavoro del database su un'istanza database Amazon RDS for SQL Server con Tuning Advisor motore di database.

- PostgreSQL – Per informazioni su come analizzare un piano di query, consulta Using EXPLAIN (Utilizzo di EXPLAIN) nella documentazione PostgreSQL. Puoi utilizzare queste informazioni per modificare una query o tabelle sottostanti in modo da migliorare le prestazioni della query.

  Per informazioni su come specificare le clausole join nella query per ottenere prestazioni ottimali, consulta Controlling the planner with explicit JOIN clauses (Controllo del pianificatore con clausole JOIN esplicite).
- MariaDB – Consulta Query optimizations (Ottimizzazioni delle query) nella documentazione MariaDB.

Best practice per l'utilizzo di MySQL

Sia le dimensioni delle tabelle che il numero di tabelle in un database MySQL possono influire sulle prestazioni.

Dimensione della tabella

In genere, i vincoli del sistema operativo sulle dimensioni dei file determinano la dimensione massima effettiva della tabella per i database MySQL. Quindi, i limiti di solito non sono determinati dai vincoli interni MySQL.

In un'istanza di database MySQL, evita che le tabelle nel database diventino troppo grandi. Sebbene il limite di storage generale sia 64 TiB, i limiti di storage forniti limitano la dimensione massima di un file di tabella MySQL a 16 TiB. Suddividi le tabelle più grandi in file di dimensioni ben al di sotto del limite di 16 TiB. Ciò può anche migliorare le prestazioni e i tempi di recupero. Per ulteriori informazioni, consulta Limiti delle dimensioni dei file MySQL in Amazon RDS.

Le tabelle di dimensioni molto grandi (maggiori di 100 GB) possono influire negativamente sulle prestazioni di lettura e scrittura (incluse le istruzioni DML e in particolare le istruzioni DDL). Gli indici sulle tabelle di grandi dimensioni possono migliorare significativamente le prestazioni selezionate, ma possono anche ridurre le prestazioni delle istruzioni DML. Le istruzioni DDL, ad esempio `ALTER TABLE`, possono essere significativamente più lente per le tabelle di grandi dimensioni perché tali operazioni potrebbero ricostruire completamente una tabella in alcuni casi. Queste istruzioni DDL potrebbero bloccare le tabelle per la durata dell'operazione.

La quantità di memoria richiesta da MySQL per le letture e le scritture dipende dalle tabelle coinvolte nelle operazioni. È una best practice avere almeno abbastanza RAM per contenere gli indici delle tabelle utilizzate attivamente . Per trovare le dieci tabelle e gli indici più grandi in un database, utilizzare la seguente query:

```
select table_schema, TABLE_NAME, dat, idx from
(SELECT table_schema, TABLE_NAME,
       ( data_length ) / 1024 / 1024 as dat,
       ( index_length ) / 1024 / 1024 as idx
FROM information_schema.TABLES
order by 3 desc ) a
order by 3 desc
limit 10;
```

Numero di tabelle

Il file system sottostante può avere un limite al numero di file che rappresentano le tabelle. Tuttavia, MySQL non ha limiti per il numero di tabelle. Ciononostante, il numero totale di tabelle nel motore di archiviazione MySQL InnoDB può contribuire al deterioramento delle prestazioni, indipendentemente dalle dimensioni di tali tabelle. Per limitare l'impatto del sistema operativo, è possibile dividere le tabelle tra più database nella stessa istanza database MySQL. In questo modo potrebbe limitare il numero di file in una directory, ma non risolverà il problema generale.

Quando c'è un deterioramento delle prestazioni a causa di un gran numero di tabelle (più di 10 mila), è causato da MySQL che lavora con i file di archiviazione, inclusa l'apertura e la chiusura. Per risolvere questo problema, è possibile aumentare la dimensione dei parametri `table_open_cache` e `table_definition_cache`. Tuttavia, aumentare i valori di tali parametri potrebbe aumentare significativamente la quantità di memoria utilizzata da MySQL e potrebbe persino utilizzare tutta la memoria disponibile. Per ulteriori informazioni, consulta Apertura e chiusura delle tabelle in MySQL nella documentazione di MySQL.

Inoltre, troppe tabelle possono influenzare significativamente il tempo di avvio di MySQL. Possono essere interessati sia un arresto e un riavvio pulito che un ripristino da arresto anomalo, specialmente nelle versioni precedenti a MySQL 8.0.

Ti consigliamo di avere meno di 10.000 tabelle totali in tutti i database in un'istanza database. Per un caso d'uso con un gran numero di tabelle in un database MySQL, consulta Un milione di tabelle in MySQL 8.0.

Motore di storage

Le funzionalità di point-in-time ripristino e ripristino delle istantanee di Amazon RDS for MySQL richiedono un motore di storage ripristinabile in caso di arresto anomalo. Queste funzionalità sono supportate solo per il motore di archiviazione InnoDB. Sebbene MySQL supporti più motori di storage con funzionalità diverse, non tutti sono ottimizzati per il recupero da arresto anomalo e per la durata dei dati. Ad esempio, il motore di archiviazione MyISAM non supporta un ripristino affidabile in caso di arresto anomalo e potrebbe impedire il corretto funzionamento di point-in-time un ripristino o di uno snapshot. Ciò può causare la perdita o il danneggiamento dei dati quando si riavvia MySQL dopo un arresto anomalo.

InnoDB è il motore di storage consigliato e supportato per istanze di database MySQL su Amazon RDS. Inoltre, a differenza delle istanze MyISAM, è possibile effettuare la migrazione delle istanze InnoDB su Aurora. Tuttavia, MyISAM offre prestazioni migliori di InnoDB quando sono richieste solide funzionalità di ricerca full-text. Se scegli comunque di usare MyISAM con Amazon RDS, le fasi illustrate in Backup automatici con motori di storage MySQL non supportati possono essere utili in determinati scenari per la funzionalità di ripristino da uno snapshot.

Se desideri convertire le tabelle MyISAM esistenti in tabelle InnoDB, puoi utilizzare la procedura illustrata nella documentazione MySQL. MyISAM e InnoDB offrono diversi vantaggi e svantaggi, per cui dovresti valutare attentamente l'impatto di questa conversione sulle tue applicazioni prima di eseguirla.

Inoltre, il Federated Storage Engine non è attualmente supportato da Amazon RDS for MySQL.

Best practice per l'utilizzo di MariaDB

Sia le dimensioni delle tabelle che il numero di tabelle in un database MariaDB possono influire sulle prestazioni.

Dimensione della tabella

In genere, i vincoli del sistema operativo sulle dimensioni dei file determinano la dimensione massima effettiva della tabella per i database MariaDB. Quindi, i limiti di solito non sono determinati dai vincoli interni MariaDB.

In un'istanza database MariaDB, evita che le tabelle nel database diventino troppo grandi. Sebbene il limite di storage generale sia 64 TiB, i limiti di storage forniti limitano la dimensione massima di un file di tabella MariaDB a 16 TiB. Suddividi le tabelle più grandi in file di dimensioni ben al di sotto del limite di 16 TiB. Ciò può anche migliorare le prestazioni e i tempi di recupero.

Le tabelle di dimensioni molto grandi (maggiori di 100 GB) possono influire negativamente sulle prestazioni di lettura e scrittura (incluse le istruzioni DML e in particolare le istruzioni DDL). Gli indici sulle tabelle di grandi dimensioni possono migliorare significativamente le prestazioni selezionate, ma possono anche ridurre le prestazioni delle istruzioni DML. Le istruzioni DDL, ad esempio `ALTER TABLE`, possono essere significativamente più lente per le tabelle di grandi dimensioni perché tali operazioni potrebbero ricostruire completamente una tabella in alcuni casi. Queste istruzioni DDL potrebbero bloccare le tabelle per la durata dell'operazione.

La quantità di memoria richiesta da MariaDB per le letture e le scritture dipende dalle tabelle coinvolte nelle operazioni. È una best practice avere almeno abbastanza RAM per contenere gli indici delle tabelle utilizzate attivamente . Per trovare le dieci tabelle e gli indici più grandi in un database, utilizzare la seguente query:

```
select table_schema, TABLE_NAME, dat, idx from
(SELECT table_schema, TABLE_NAME,
       ( data_length ) / 1024 / 1024 as dat,
       ( index_length ) / 1024 / 1024 as idx
FROM information_schema.TABLES
order by 3 desc ) a
order by 3 desc
limit 10;
```

Numero di tabelle

Il file system sottostante può avere un limite al numero di file che rappresentano le tabelle. Tuttavia, MariaDB non ha limiti per il numero di tabelle. Ciononostante, il numero totale di tabelle nel motore di archiviazione MariaDB InnoDB può contribuire al deterioramento delle prestazioni, indipendentemente dalle dimensioni di tali tabelle. Per limitare l'impatto del sistema operativo, è

possibile dividere le tabelle tra più database nella stessa istanza database di MariaDB. In questo modo potrebbe limitare il numero di file in una directory, ma non risolve il problema generale.

Il deterioramento delle prestazioni a causa di un gran numero di tabelle (più di 10.000) è provocato da MariaDB che lavora con i file di archiviazione, compresa l'apertura e la chiusura dei file di archiviazione in MariaDB. Per risolvere questo problema, è possibile aumentare la dimensione dei parametri `table_open_cache` e `table_definition_cache`. Tuttavia, aumentando i valori di tali parametri potrebbe aumentare significativamente la quantità di memoria utilizzata da MariaDB e potrebbe anche utilizzare tutta la memoria disponibile. Per ulteriori informazioni, consulta Ottimizzazione di table_open_cache nella documentazione di MariaDB.

Inoltre, troppe tabelle possono influenzare significativamente il tempo di avvio di MariaDB. Possono essere interessati sia un arresto e un riavvio pulito che un ripristino di arresto anomalo del sistema. Si consiglia di avere meno di diecimila tabelle totali in tutti i database in un'istanza database.

Motore di storage

Le funzionalità di point-in-time ripristino e ripristino delle istantanee di Amazon RDS for MariaDB richiedono un motore di storage ripristinabile in caso di arresto anomalo. Sebbene MariaDB supporti più motori di storage con funzionalità diverse, non tutti sono ottimizzati per il recupero da arresto anomalo e per la durata dei dati. Ad esempio, sebbene Aria sia un sostituto sicuro di MyISAM, potrebbe comunque impedire che un point-in-time ripristino o un ripristino di istantanee funzionino come previsto. Ciò può causare la perdita o il danneggiamento dei dati quando si riavvia MariaDB dopo un arresto anomalo. InnoDB è il motore di storage consigliato e supportato per istanze di database MariaDB su Amazon RDS. Se scegli comunque di usare Aria con Amazon RDS, le fasi illustrate in Backup automatici con motori di storage MariaDB non supportati possono essere utili in determinati scenari per la funzionalità di ripristino da uno snapshot.

Se desideri convertire le tabelle MyISAM esistenti in tabelle InnoDB, puoi utilizzare la procedura illustrata nella documentazione di MariaDB. MyISAM e InnoDB offrono diversi vantaggi e svantaggi, per cui dovresti valutare attentamente l'impatto di questa conversione sulle tue applicazioni prima di eseguirla.

Best practice per l'utilizzo di Oracle

Per informazioni sulle best practice per l'uso di Amazon RDS for Oracle, consulta la pagina relativa alle best practice per l'esecuzione di Oracle Database su Amazon Web Services.

Un workshop AWS virtuale del 2020 includeva una presentazione sull'esecuzione dei database Oracle di produzione su Amazon RDS. Un video della presentazione è disponibile qui.

Best practice per l'utilizzo di PostgreSQL

Esistono due momenti importanti in cui puoi migliorare le prestazioni con RDS per PostgreSQL: uno è quando carichi i dati in un'istanza database e l'altro è quando usi la funzione di vacuum automatico di PostgreSQL. Le sezioni seguenti illustrano alcune delle prassi suggerite per queste aree.

Per informazioni su come Amazon RDS implementa altre attività comuni del DBA PostgreSQL, consulta Attività DBA comuni per Amazon RDS for PostgreSQL.

Caricamento di dati in un'istanza di database PostgreSQL

Durante il caricamento di dati in un'istanza database Amazon RDS per PostgreSQL, dovresti modificare le impostazioni dell'istanza database e i valori del gruppo di parametri del database per consentire un'importazione più efficace dei dati nell'istanza database.

Modifica i parametri dell'istanza database come segue:

- Disabilità i backup dell'istanza di database (imposta backup_retention su 0)
- Disabilita Multi-AZ.

Modifica il gruppo di parametri del database in modo da includere le seguenti impostazioni. Per individuare le impostazioni più efficienti per la tua istanza database è necessario testare le impostazioni dei parametri.

- Incrementa il valore del parametro `maintenance_work_mem`. Per ulteriori informazioni sui parametri relativi al consumo di risorse di PostgreSQL, consulta la documentazione di PostgreSQL.
- Incrementa il valore dei parametri `max_wal_size` e `checkpoint_timeout` per ridurre il numero di scritture nel registro WAL (write-ahead log).
- Disabilitare il parametro `synchronous_commit`
- Disabilita il parametro di eliminazione automatica di PostgreSQL.
- Verifica che tutte le tabelle che stai importando siano registrate. I dati memorizzati in tabelle non registrate potrebbero andare smarriti durante un failover. Per ulteriori informazioni, consulta CREATE TABLE UNLOGGED (CREA TABELLA NON REGISTRATA).

Utilizza i comandi `pg_dump -Fc` (compresso) o `pg_restore -j` (parallelo) con queste impostazioni.

Al termine dell'operazione di caricamento, ripristina le normali impostazioni dell'istanza database e dei parametri database.

Utilizzo della funzione di eliminazione automatica di PostgreSQL

La funzione di eliminazione automatica per i database di PostgreSQL è una funzione vivamente consigliata per mantenere l'integrità dell'istanza di database di PostgreSQL. La funzione di eliminazione automatica automatizza l'esecuzione dei comandi VACUUM e ANALYZE. Il suo utilizzo è richiesto da PostgreSQL, non imposto da Amazon RDS, ed è essenziale per garantire buone prestazioni. La funzione è abilitata per impostazione predefinita per tutte le nuove istanze database Amazon RDS per PostgreSQL e i relativi parametri di configurazione sono impostati in modo appropriato per impostazione predefinita.

L'amministratore del database è tenuto a conoscere e a comprendere questa operazione di manutenzione. Per la documentazione PostgreSQL sull'autovacuum, consulta The Autovacuum Daemon (Daemon di autovacuum).

La funzione di eliminazione automatica non è un'operazione "priva di risorse", ma viene eseguita in background e, per quanto possibile, assegna la precedenza alle operazioni dell'utente. Se abilitata, verifica la presenza di tabelle con un numero elevato di tuple aggiornate o eliminate. Inoltre, protegge dalla perdita di dati molto vecchi in seguito a wraparound dell'ID transazione. Per ulteriori informazioni, consulta Impedire gli errori di wraparound dell'ID transazione.

La funzione di eliminazione automatica non deve essere considerata un'operazione dal sovraccarico elevato che può essere limitata per migliorare le prestazioni. Al contrario, le tabelle con un'elevata velocità di aggiornamento ed eliminazione si deteriorano rapidamente nel tempo se non viene eseguita la funzione di eliminazione automatica.

> **Important**
>
> La mancata esecuzione della funzione di eliminazione automatica potrebbe comportare l'interruzione delle attività per eseguire un'operazione di eliminazione molto più intrusiva. In alcuni casi, l'istanza database RDS per PostgreSQL potrebbe non essere disponibile a causa di un uso troppo conservativo del vacuum automatico. In questi casi, il database PostgreSQL si arresta per proteggersi. A quel punto, Amazon RDS deve eseguire un vuoto single-user-mode completo direttamente sull'istanza DB. la quale può comportare un'interruzione delle

attività di diverse ore. Pertanto, è consigliabile non disattivare la funzione di eliminazione automatica, che è abilitata per impostazione predefinita.

I parametri della funzione di eliminazione automatica determinano quando viene eseguita e con quale intensità. I parametri `autovacuum_vacuum_threshold` e `autovacuum_vacuum_scale_factor` determinano quando viene eseguita la funzione di eliminazione automatica. I parametri `autovacuum_max_workers`, `autovacuum_nap_time`, `autovacuum_cost_limit` e `autovacuum_cost_delay` determinano con quale intensità viene eseguita la funzione di eliminazione automatica. Per ulteriori informazioni sull'autovacuum, su quando viene eseguito e sui parametri richiesti, consulta la Routine Vacuuming (Vacuum di routine) nella documentazione di PostgreSQL.

La seguente query mostra il numero di tuple "inattive" in una tabella denominata table1:

```
SELECT relname, n_dead_tup, last_vacuum, last_autovacuum FROM
pg_catalog.pg_stat_all_tables
WHERE n_dead_tup > 0 and relname = 'table1';
```

I risultati della query saranno simili a quelli nell'esempio seguente:

```
relname | n_dead_tup | last_vacuum | last_autovacuum
---------+------------+-------------+-----------------
 tasks   |   81430522 |             |
(1 row)
```

Video su best practice di Amazon RDS for PostgreSQL

La conferenza AWS re:Invent 2020 ha incluso una presentazione sulle nuove funzionalità e le best practice per lavorare con PostgreSQL su Amazon RDS. Un video della presentazione è disponibile qui.

Best practice per l'utilizzo di SQL Server

Le best practice per un'implementazione Multi-AZ con un'istanza di database di SQL Server sono le seguenti:

- Utilizza eventi di database Amazon RDS per monitorare i failover. Ad esempio, puoi ricevere un avvio via sms o e-mail in caso di failover di un'istanza di database. Per ulteriori informazioni sugli eventi di Amazon RDS, consulta Utilizzo della notifica degli eventi di Amazon RDS.
- Se l'applicazione memorizza nella cache i valori DNS, imposta un time to live (TTL) inferiore a 30 secondi. Impostare il TTL in questo modo è utile in caso di failover. In un failover, l'indirizzo IP può cambiare e il valore memorizzato nella cache non essere più in uso.
- Si consiglia di non abilitare le seguenti modalità in quanto disattivano il log delle transazioni, necessario per l'implementazione Multi-AZ:
  - Modalità di recupero semplice
  - Modalità offline
  - Modalità di sola lettura
- Esegui un test per determinare il tempo di failover dell'istanza di database. Il tempo di failover può variare in base al tipo di database, alla classe dell'istanza e al tipo di storage che utilizzi. Dovresti inoltre verificare la capacità dell'applicazione di continuare a funzionare in caso di failover.
- Per ridurre il tempo di failover, procedi come indicato di seguito:
  - Assicurati di disporre di sufficienti IOPS assegnati per il carico di lavoro. Un numero di operazioni I/O inadeguato può incrementare il tempo di failover. Il recupero del database richiede operazioni I/O.
  - Utilizza transazioni di piccole dimensioni. Il ripristino del database si basa sulle transazioni, per cui suddividendo le transazioni di grandi dimensioni in più transazioni di dimensioni inferiori, il tempo di failover dovrebbe ridursi.
- Tieni presente che durante un failover le latenze sono elevate. Nell'ambito del processo di failover, Amazon RDS esegue automaticamente la replica dei dati in una nuova istanza di standby. Questa replica significa che viene eseguito il commit dei nuovi dati su due diverse istanze database. Pertanto, fino a quando l'istanza database in standby non raggiunge la nuova istanza database principale, potrebbe verificarsi una latenza.
- Distribuisci le applicazioni in tutte le zone di disponibilità. Se una zona di disponibilità non è raggiungibile, le applicazioni saranno ancora disponibili nelle altre zone di disponibilità.

Quando si utilizza un'implementazione Multi-AZ di SQL Server, ricordarsi che Amazon RDS crea le repliche per tutti i database SQL Server nell'istanza. Se non si desidera che database specifici abbiano repliche secondarie, impostare un'istanza database separata che non utilizza Multi-AZ per questi database.

Video su best practice di Amazon RDS for SQL Server

La conferenza AWS re:Invent 2019 ha incluso una presentazione sulle nuove funzionalità e sulle best practice per lavorare con SQL Server su Amazon RDS. Un video della presentazione è disponibile qui.

Utilizzo di gruppi di parametri di database

È consigliabile provare le modifiche al gruppo di parametri di database su un'istanza di database di test prima di applicarle a istanze di database di produzione. Un'impostazione non corretta dei parametri del motore di un database in un gruppo di parametri di database può avere ripercussioni negative non previste, tra cui prestazioni scadenti e instabilità del sistema. Fai sempre attenzione quando modifichi i parametri del motore di un database ed effettui il backup di un'istanza di database prima di modificare un gruppo di parametri di database.

Per ulteriori informazioni sul backup di un'istanza di database, consulta Backup, ripristino ed esportazione dei dati.

Best practice per automatizzare la creazione di istanze database

È una best practice di Amazon RDS creare un'istanza database con la versione secondaria preferita del motore di database. Puoi utilizzare l' AWS CLI API Amazon RDS o automatizzare la creazione AWS CloudFormation di istanze DB. Quando utilizzi questi metodi, puoi specificare solo la versione principale e Amazon RDS crea automaticamente l'istanza con la versione secondaria preferita. Ad esempio, se PostgreSQL 12.5 è la versione secondaria preferita e se si specifica la versione 12 con `create-db-instance`, l'istanza database sarà la versione 12.5.

Per determinare la versione secondaria preferita, è possibile eseguire il comando `describe-db-engine-versions` con l'opzione `--default-only` come mostrato nell'esempio seguente.

```
aws rds describe-db-engine-versions --default-only --engine postgres

{
    "DBEngineVersions": [
        {
            "Engine": "postgres",
            "EngineVersion": "12.5",
            "DBParameterGroupFamily": "postgres12",
            "DBEngineDescription": "PostgreSQL",
```

```
            "DBEngineVersionDescription": "PostgreSQL 12.5-R1",
            ...some output truncated...
        }
    ]
}
```

Per informazioni sulla creazione di istanze database a livello di programmazione, vedere le seguenti risorse:

- Utilizzando il — AWS CLI create-db-instance
- Utilizzo dell'API Amazon RDS – CreateDBInstance
- Utilizzo di AWS CloudFormation — AWS: :RDS: :dbInstance

Video sulle nuove funzionalità di Amazon RDS

La conferenza AWS re:Invent del 2023 ha incluso una presentazione sulle nuove funzionalità di Amazon RDS. Un video della presentazione è disponibile qui.

Configurazione di un'istanza database Amazon RDS

Questa sezione illustra come configurare l'istanza database Amazon RDS. Prima di creare un'istanza database, decidere la classe di istanza database che eseguirà l'istanza database. Inoltre, decidi dove verrà eseguita l'istanza DB scegliendo una AWS regione. Quindi, creare l'istanza database.

È possibile configurare un'istanza database con un gruppo di opzioni e un gruppo di parametri del database.

- Un gruppo di opzioni può specificare le caratteristiche, denominate opzioni, disponibili per una determinata istanza database Amazon RDS.
- Un gruppo di parametri database agisce come un container per i valori di configurazione di un motore applicati a una o più istanze database.

Le opzioni e i parametri disponibili dipendono dal motore del database e dalla versione del motore del database. Puoi specificare un gruppo di opzioni e un gruppo di parametri database quando si crea un'istanza database oppure puoi modificare un'istanza database per specificarli.

Argomenti

Creazione di un'istanza database Amazon RDS

L'elemento di base di Amazon RDS è l'istanza database, in cui si creano i database. Quando si crea un'istanza database, si scelgono le caratteristiche specifiche del motore. È inoltre possibile scegliere la capacità di archiviazione, la CPU, la memoria e così via dell' AWS istanza su cui viene eseguito il server di database.

Argomenti

Prerequisiti per l'istanza database

> **Important**
>
> Completa le attività indicate nella sezione Configurazione di Amazon RDS prima di poter creare un'istanza database.

Di seguito sono indicati i prerequisiti per creare un'istanza database RDS.

Argomenti

Configurazione della rete per l'istanza database

Puoi creare un'istanza database Amazon RDS solo in un cloud privato virtuale (VPC) basato sul servizio Amazon VPC. Inoltre, deve trovarsi in un ambiente Regione AWS con almeno due zone di disponibilità. Il gruppo di sottoreti DB scelto per l'istanza database deve coprire almeno due zone di disponibilità. Questa configurazione garantisce la possibilità di configurare un'implementazione Multi-AZ quando si crea l'istanza database o si passa facilmente a un'istanza in futuro.

La connettività tra la nuova istanza database e un'istanza Amazon EC2 nello stesso VPC può essere configurata durante la creazione dell'istanza database. Per connetterti alla tua istanza database

da risorse diverse dalle istanze EC2 nello stesso VPC, puoi configurare le connessioni di rete manualmente.

Argomenti

Configurazione della connettività di rete automatica con un'istanza EC2

Quando crei un'istanza DB RDS, puoi utilizzare la AWS Management Console per configurare la connettività tra un'istanza EC2 e la nuova istanza DB. In questo caso, RDS configura automaticamente il VPC e le impostazioni di rete. L'istanza database viene creata nello stesso VPC dell'istanza EC2, per consentire all'istanza EC2 di accedere all'istanza database.

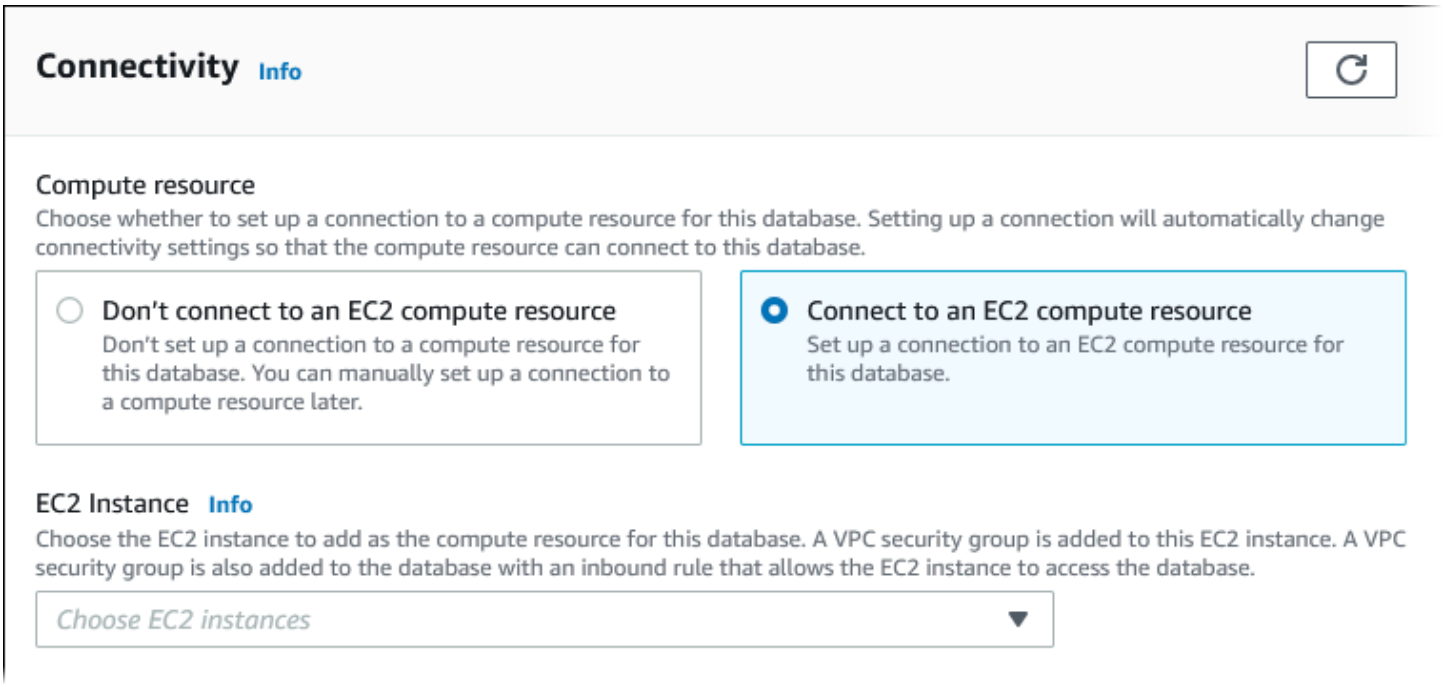
Di seguito sono riportati i requisiti per connettere un'istanza EC2 all'istanza database:

- L'istanza EC2 deve esistere Regione AWS prima di creare l'istanza DB.

  Se non è presente alcuna istanza EC2 Regione AWS, la console fornisce un collegamento per crearne una.
- L'utente che sta creando l'istanza database deve disporre delle autorizzazioni per eseguire le seguenti operazioni:
  - `ec2:AssociateRouteTable`
  - `ec2:AuthorizeSecurityGroupEgress`
  - `ec2:AuthorizeSecurityGroupIngress`
  - `ec2:CreateRouteTable`
  - `ec2:CreateSubnet`
  - `ec2:CreateSecurityGroup`
  - `ec2:DescribeInstances`
  - `ec2:DescribeNetworkInterfaces`
  - `ec2:DescribeRouteTables`
  - `ec2:DescribeSecurityGroups`
  - `ec2:DescribeSubnets`
  - `ec2:ModifyNetworkInterfaceAttribute`
  - `ec2:RevokeSecurityGroupEgress`

L'utilizzo di questa opzione crea un'istanza database privata. L'istanza database utilizza un gruppo di sottoreti DB con solo sottoreti private per limitare l'accesso alle risorse all'interno del VPC.

Per connettere un'istanza EC2 all'istanza database, scegli Connect to an EC2 compute resource (Connetti a una risorsa di calcolo EC2) nella sezione Connectivity (Connettività) della pagina Create database (Crea database).



Se scegli Connect to an EC2 compute resource (Connetti a una risorsa di calcolo EC2), RDS imposta automaticamente le seguenti opzioni. Queste impostazioni non possono essere modificate a meno che non si scelga di non configurare la connettività con un'istanza EC2 selezionando Don't connect to an EC2 compute resource (Non connetterti a una risorsa di calcolo EC2).

Opzione Console	Impostazione automatica
Tipo di rete	RDS imposta il tipo di rete su IPv4. Attualmente, la modalità dual-stack non è supportata quando si configura una connessio ne tra un'istanza EC2 e l'istanza database.
Virtual Private Cloud (VPC)	RDS imposta il VPC su quello associato all'istanza EC2.
	RDS richiede un gruppo di sottoreti database con una sottorete privata nella stessa zona di disponibilità dell'istanza EC2. Se

Opzione Console	Impostazione automatica
DB subnet group (Gruppo di sottoreti DB)	esiste un gruppo di sottoreti database che soddisfa questo requisito, RDS lo utilizza. Per impostazione predefinita, questa opzione è impostata su Automatic setup (Configurazione automatica). Quando si sceglie Automatic setup (Configurazione automatica) e non esiste un gruppo di sottoreti database che soddisfi questo requisito, viene eseguita la seguente procedura. RDS utilizza tre sottoreti private disponibili in tre zone di disponibilità di cui una è la stessa dell'istanza EC2. Se una sottorete privata non è disponibile in una zona di disponibilità, RDS crea una sottorete privata nella zona di disponibilità. Quindi RDS crea il gruppo di sottoreti database. Quando è disponibile una sottorete privata, RDS utilizza la tabella di instradamento associata alla sottorete e aggiunge tutte le sottoreti create a questa tabella di instradamento. Quando una sottorete privata non è disponibile, RDS crea una tabella di instradamento senza accesso al gateway Internet e aggiunge le sottoreti create alla tabella di instradamento. RDS consente inoltre di utilizzare i gruppi di sottoreti database esistenti. Seleziona Choose existing (Scegli esistente) se desideri utilizzare un gruppo di sottoreti database esistente.
Accesso pubblico	RDS sceglie No in modo che il cluster database non sia accessibile pubblicamente. Per motivi di sicurezza, come best practice si consiglia di mantenere il database privato e accertarsi che non sia accessibi le da Internet.

Opzione Console	Impostazione automatica
VPC security group (firewall) (Gruppo di sicurezza VPC (firewall)	RDS crea un nuovo gruppo di sicurezza associato all'istanza database. Il gruppo di sicurezza è denominato `rds-ec2-`*n*, dove *n* è un numero. Questo gruppo di sicurezza include una regola in entrata con il gruppo di sicurezza VPC EC2 (firewall) come origine. Questo gruppo di sicurezza associato all'istan za database consente all'istanza EC2 di accedere all'istanza database. RDS crea, inoltre, un nuovo gruppo di sicurezza associato all'istanza EC2. Il gruppo di sicurezza è denominato `ec2-rds-`*n*, dove *n* è un numero. Questo gruppo di sicurezza include una regola in uscita con il gruppo di sicurezza VPC dell'istanza database come origine. Questo gruppo di sicurezza consente all'istanza EC2 di inviare traffico all'istanza database. Puoi aggiungere un nuovo gruppo di sicurezza aggiuntivo scegliendo Create nuovo (Crea nuovo) e digitando il nome del nuovo gruppo di sicurezza. Puoi aggiungere gruppi di sicurezza esistenti scegliendo Choose existing (Scegli esistente) e selezionando i gruppi di sicurezza da aggiungere.

Opzione Console	Impostazione automatica
Zona di disponibilità	Quando si sceglie Single DB instance (Istanza database singola) in Availability & durability (Disponibilità e durabilità) (implementazione Single-AZ), RDS sceglie la zona di disponibilità dell'istanza EC2. Quando si sceglie Multi-AZ DB instance (Istanza database Multi-AZ) in Availability & durability (Disponibilità e durabilità) (implementazione di istanze database Multi-AZ), RDS sceglie la zona di disponibilità dell'istanza EC2 per un'istanza database nell'implementazione. RDS sceglie casualmente una zona di disponibilità diversa per l'altra istanza database. L'istanza database primaria o la replica di standby vengono create nella stessa zona di disponibilità dell'istanza EC2. Quando si sceglie Multi-AZ DB instance (Istanza database Multi-AZ), è possibile che vengano addebitati costi tra zone di disponibilità se l'istanza database e l'istanza EC2 si trovano in zone di disponibilità diverse.

Per ulteriori informazioni su queste impostazioni, consultare Impostazioni per istanze database.

Se dopo la creazione dell'istanza database le impostazioni vengono modificate, le modifiche potrebbero influire sulla connessione tra l'istanza EC2 e l'istanza database.

Configurazione manuale della rete

Per connetterti alla tua istanza database da risorse diverse dalle istanze EC2 nello stesso VPC, puoi configurare le connessioni di rete manualmente. Se utilizzi il AWS Management Console per creare la tua istanza DB, puoi fare in modo che Amazon RDS crei automaticamente un VPC per te. Altrimenti, puoi utilizzare un VPC esistente o crearne uno nuovo per la tua istanza di database . Qualunque sia l'approccio scelto, il VPC deve disporre di almeno una sottorete in ciascuna di almeno due zone di disponibilità per poterlo utilizzare con un'istanza database RDS.

Per impostazione predefinita, Amazon RDS crea automaticamente l'istanza database una zona di disponibilità. Per scegliere una zona di disponibilità specifica, è necessario impostare l'opzione Availability & durability (Disponibilità e durabilità) su Single DB instance (Istanza database singola).

Questo espone un'impostazione Availability Zone (Zona di disponibilità) che consente di scegliere tra le zone di disponibilità nel VPC. Tuttavia, se si sceglie una implementazione Multi-AZ, RDS sceglie automaticamente la zona di disponibilità dell'istanza database primaria o di scrittura e l'opzione Availability Zone (Zona di disponibilità) non viene visualizzata.

In alcuni casi, potresti non avere un VPC predefinito o uno già creato. In questi casi, puoi fare in modo che un VPC venga creato automaticamente da Amazon RDS durante la creazione di un'istanza database utilizzando la console. In caso contrario, eseguire le seguenti operazioni:

- Crea un VPC con almeno una sottorete in ognuna delle almeno due zone di disponibilità nel luogo in Regione AWS cui desideri distribuire l'istanza DB. Per ulteriori informazioni, consulta Uso di un'istanza database in un VPC e Tutorial: Creazione di un Amazon VPC da utilizzare con un'istanza database (solo IPv4).
- Specificare un gruppo di sicurezza VPC che autorizzi le connessioni all'istanza database. Per ulteriori informazioni, consulta Fornisci accesso alla istanza database nel VPC creando un gruppo di sicurezza e Controllo dell'accesso con i gruppi di sicurezza.
- Specificare un gruppo di sottoreti DB RDS che definisca almeno due sottoreti nel VPC che possono essere utilizzate dall'istanza database. Per ulteriori informazioni, consulta Utilizzo di gruppi di sottoreti database.

Se desideri connetterti a una risorsa che non si trova nello stesso VPC dell'istanza database, consulta gli scenari appropriati descritti in Scenari per accedere a un'istanza database in un VPC.

Prerequisiti aggiuntivi

Prima di creare l'istanza database, considera i seguenti prerequisiti aggiuntivi:

- Se ti connetti AWS utilizzando credenziali AWS Identity and Access Management (IAM), il tuo AWS account deve disporre di determinate politiche IAM. che concedono le autorizzazioni necessarie per eseguire le operazioni Amazon RDS. Per ulteriori informazioni, consulta Gestione accessi e identità per Amazon RDS.

  Per utilizzare IAM per accedere alla console RDS, accedi AWS Management Console con le tue credenziali utente IAM. Quindi, passa alla console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
- Per personalizzare i parametri di configurazione per l'istanza database, specifica un gruppo di parametri database con le impostazioni dei parametri richieste. Per informazioni sulla creazione o la modifica di un gruppo di parametri database, consulta Utilizzo di gruppi di parametri.

> **Important**
>
> Se utilizzi il modello BYOL per RDS for Db2, prima di creare un'istanza DB, devi prima creare un gruppo di parametri personalizzato che contenga il tuo and. IBM Site ID IBM Customer ID Per ulteriori informazioni, consulta Porta la tua licenza per Db2.

- Determina il numero di porta TCP/IP da specificare per l'istanza database. I firewall di alcune aziende bloccano le connessioni alle porte predefinite delle istanze database RDS. Se il firewall della tua azienda blocca la porta predefinita, per l'istanza database specifica una porta diversa. Le porte predefinite per i motori Amazon RDS DB sono:

RDS per Db2	RDS per MariaDB	RDS for MySQL	RDS per Oracle	RDS per PostgreSQL.	RDS per SQL Server
50000	3306	3306	1521	5432	1433

  Per RDS per SQL Server, le seguenti porte sono riservate e non è possibile utilizzarle quando si crea un'istanza DB: e. `1234, 1434, 3260, 3343, 3389, 47001, 49152-49156`

Creazione di un'istanza database

Puoi creare un'istanza database Amazon RDS utilizzando l' AWS Management Console API AWS CLI, the o RDS.

> **Note**
>
> Per RDS for Db2, ti consigliamo di configurare gli elementi necessari per il tuo modello di licenza prima di creare un'istanza DB RDS for Db2. Per ulteriori informazioni, consulta Opzioni di licenza Amazon RDS per Db2.

Console

È possibile creare un'istanza DB utilizzando Easy create abilitato o non abilitato. AWS Management Console Con l'opzione Easy create (Creazione rapida) attivata, specifichi solo il tipo di motore del database, la dimensione dell'istanza di database e l'identificatore dell'istanza di database. Easy

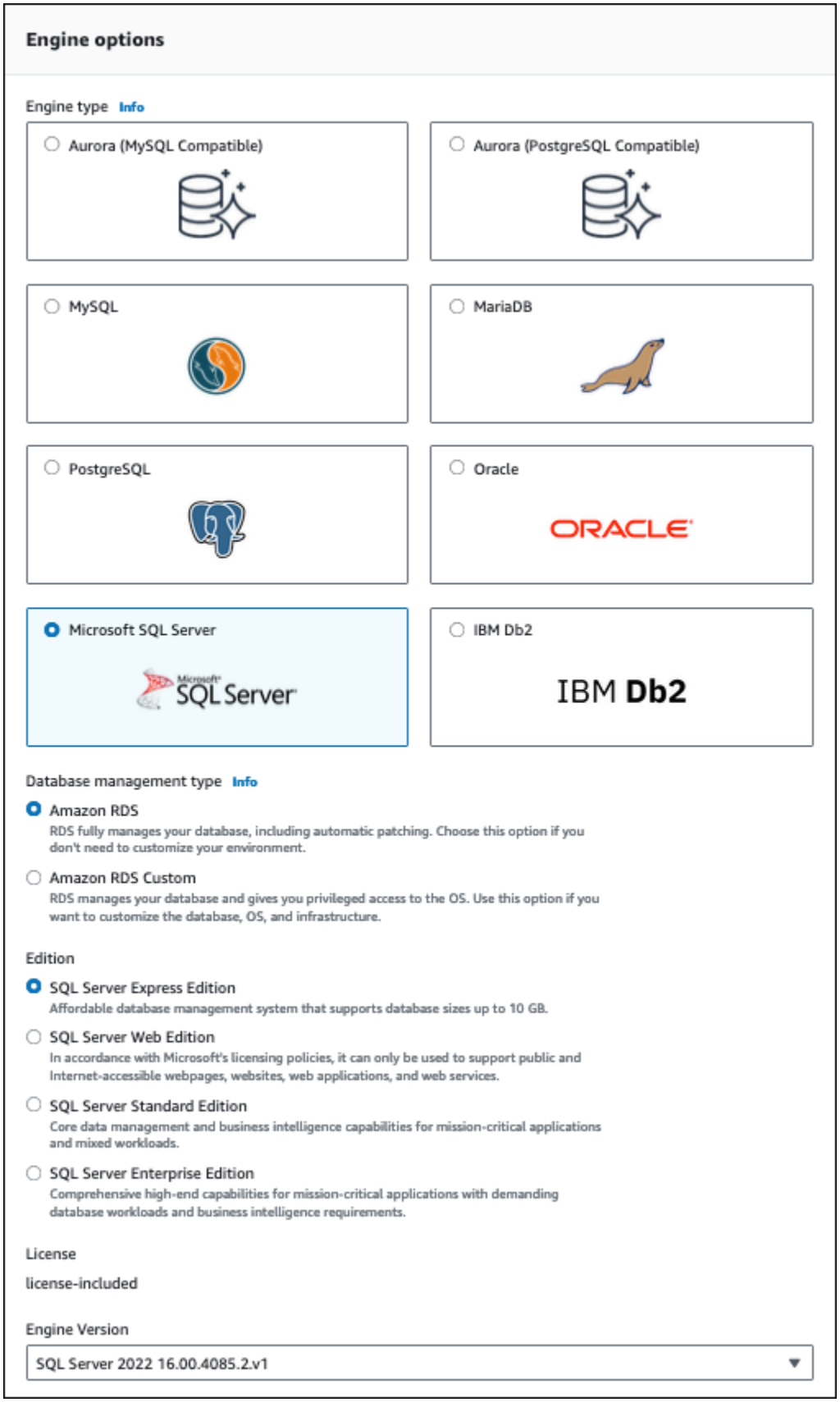
create (Creazione rapida) utilizza l'impostazione predefinita per altre opzioni di configurazione. Con l'opzione Easy create (Creazione rapida) disattivata, specifichi più opzioni di configurazione quando crei un database, comprese quelle relative a disponibilità, sicurezza, backup e manutenzione.

> **Note**
>
> Nella procedura seguente, Standard create (Creazione standard) è attivato e Easy create (Creazione rapida) non è attivata. Questa procedura utilizza Microsoft SQL Server come un esempio.
> Per esempi che utilizzano Easy create (Creazione rapida) per illustrare la creazione e la connessione ad istanze database di esempio per ciascun motore, consulta Nozioni di base su Amazon RDS.

Per creare un'istanza database.

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nell'angolo in alto a destra della console Amazon RDS, scegli la regione AWS in cui desideri creare l'istanza database.
3. Nel riquadro di navigazione, scegliere Databases (Database).
4. Scegli Crea database, quindi Creazione standard.
5. Per il tipo di motore, scegli IBM Db2, MariaDB, Microsoft SQL Server, MySQL, Oracle o PostgreSQL.

   Microsoft SQL Server è visualizzato qui.


Engine options
Engine type Info
Aurora (MySQL Compatible)
Aurora (PostgreSQL Compatible)
MySQL
MariaDB
PostgreSQL
Oracle
ORACLE
Microsoft SQL Server
SQL Server
IBM Db2
IBM Db2
Database management type Info
Amazon RDS
RDS fully manages your database, including automatic patching. Choose this option if you don't need to customize your environment.
Amazon RDS Custom
RDS manages your database and gives you privileged access to the OS. Use this option if you want to customize the database, OS, and infrastructure.
Edition
SQL Server Express Edition
Affordable database management system that supports database sizes up to 10 GB.
SQL Server Web Edition
In accordance with Microsoft's licensing policies, it can only be used to support public and Internet-accessible webpages, websites, web applications, and web services.
SQL Server Standard Edition
Core data management and business intelligence capabilities for mission-critical applications and mixed workloads.
SQL Server Enterprise Edition
Comprehensive high-end capabilities for mission-critical applications with demanding database workloads and business intelligence requirements.
License
license-included
Engine Version
SQL Server 2022 16.00.4085.2.v1

6. Per Tipo di gestione del database, se utilizzi Oracle o SQL Server, scegli Amazon RDS o Amazon RDS Custom.

   Amazon RDS è visualizzato nell'immagine. Per ulteriori informazioni sui RDS Custom, consulta Utilizzo di Amazon RDS Custom.

7. Per Edition, se utilizzi Db2, Oracle o SQL Server, scegli l'edizione del motore DB che desideri utilizzare.

   MySQL dispone di una sola opzione per l'edizione, mentre non è disponibile per MariaDB e PostgreSQL.

8. In Version (Versione), scegliere la versione del motore.

9. In Templates (Modelli), selezionare il modello che corrisponde al proprio caso d'uso. Quando si sceglie Production (Produzione), tutte le opzioni seguenti vengono preselezionate in una fase successiva:

   - Opzione di failover Multi-AZ
   - Opzione di archiviazione SSD per capacità di IOPS allocata (io1)
   - Opzione Enable deletion protection (Abilita protezione da eliminazione)

   Consigliamo queste caratteristiche per qualsiasi ambiente di produzione.

   > **Note**
   >
   > La scelte del modello variano a seconda dell'edizione.

10. Per inserire la password principale, procedere come segue:

    a. Nella sezione Settings (Impostazioni), aprire Credential Settings (Impostazioni credenziali).

    b. Se si desidera specificare una password, deselezionare la casella di spunta Auto generate a password (Genera password automaticamente) se è selezionata.

    c. (Facoltativo) Cambiare il valore Master username (Nome utente master).

    d. Inserisci la stessa password in Master password (Password master) e Confirm password (Conferma password).

11. (Facoltativo) Configura una connessione a una risorsa di calcolo per questa istanza database.

Puoi configurare la connettività tra un'istanza Amazon EC2 e la nuova istanza database durante la creazione dell'istanza database. Per ulteriori informazioni, consulta Configurazione della connettività di rete automatica con un'istanza EC2.

12. Nella sezione Connettività in Gruppo di sicurezza VPC (firewall), se selezioni Crea nuovo, viene creato un gruppo di sicurezza VPC con una regola in entrata che consente all'indirizzo IP del computer locale di accedere al database.
13. Per le restanti sezioni, specifica le impostazioni dell'istanza database. Per informazioni su ciascuna impostazione, consulta Impostazioni per istanze database.
14. Scegliere Create database (Crea database).

    Se scegli di utilizzare una password generata in modo automatico, il pulsante View credential details (Vedi dettagli delle credenziali) appare nella paginaDatabases.

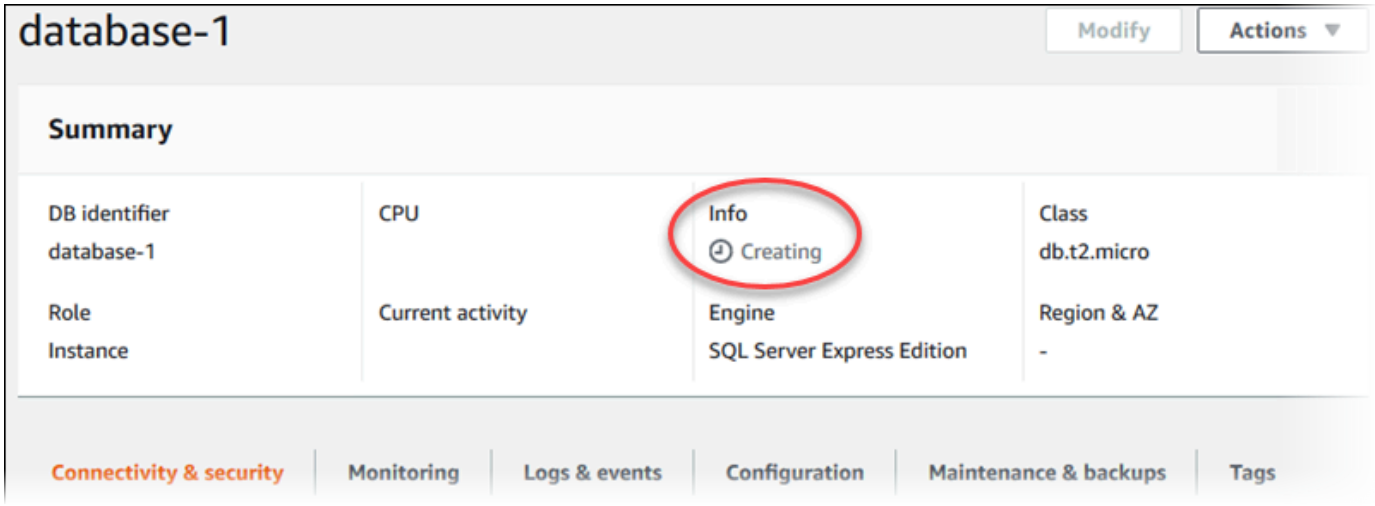
    Per vedere nome utente e password per l'istanza database, seleziona View credential details (Vedi dettagli delle credenziali).

    Per connetterti all'istanza database come utente principale, utilizza il nome utente e la password visualizzati.

    > **Important**
    >
    > Non potrai visualizzare di nuovo la password dell'utente principale. Se non la registri, potresti doverla modificare. Se devi modificare la password dell'utente principale dopo che l'istanza database è disponibile, modifica l'istanza database per eseguire tale operazione. Per ulteriori informazioni sulla modifica di un'istanza database, consulta Modifica di un'istanza database Amazon RDS.

15. Per Databases (Database), seleziona il nome della nuova istanza database.

    Nella console RDS vengono visualizzati i dettagli per la nuova istanza database. L'istanza database rimane nello stato Creating (In creazione) fino a quando non viene creata ed è pronta per l'uso. Quando lo stato cambia in Available (Disponibile) è possibile connettersi all'istanza database. A seconda della classe di istanza database e dello store allocato, potrebbero trascorrere diversi minuti prima che la nuova istanza sia disponibile.



AWS CLI

> **Note**
>
> Se desideri utilizzare la licenza Db2 tramite Marketplace AWS, devi prima abbonarti Marketplace AWS e registrarti con IBM utilizzando il. AWS Management Console Per ulteriori informazioni, consulta Iscrizione alle inserzioni di Db2 Marketplace e registrazione con IBM.

Per creare un'istanza DB utilizzando il AWS CLI, chiama il create-db-instancecomando con i seguenti parametri:

- `--db-instance-identifier`
- `--db-instance-class`
- `--vpc-security-group-ids`
- `--db-subnet-group`
- `--engine`
- `--master-username`
- `--master-user-password`
- `--allocated-storage`
- `--backup-retention-period`

Per informazioni su ciascuna impostazione, consulta Impostazioni per istanze database.

In questo esempio viene utilizzato Microsoft SQL Server.

Example

Per LinuxmacOS, oUnix:

```
aws rds create-db-instance \
    --engine sqlserver-se \
    --db-instance-identifier mymsftsqlserver \
    --allocated-storage 250 \
    --db-instance-class db.t3.large \
    --vpc-security-group-ids mysecuritygroup \
    --db-subnet-group mydbsubnetgroup \
    --master-username masterawsuser \
    --manage-master-user-password \
    --backup-retention-period 3
```

Per Windows:

```
aws rds create-db-instance ^
    --engine sqlserver-se ^
    --db-instance-identifier mydbinstance ^
    --allocated-storage 250 ^
    --db-instance-class db.t3.large ^
    --vpc-security-group-ids mysecuritygroup ^
    --db-subnet-group mydbsubnetgroup ^
    --master-username masterawsuser ^
    --manage-master-user-password ^
    --backup-retention-period 3
```

Questo comando genera un output simile al seguente.

```
DBINSTANCE  mydbinstance  db.t3.large  sqlserver-se  250  sa  creating  3  ****  n
 10.50.2789
SECGROUP  default  active
PARAMGRP  default.sqlserver-se-14  in-sync
```

API RDS

> **Note**
>
> Se desideri utilizzare la licenza Db2 tramite Marketplace AWS, devi prima abbonarti Marketplace AWS e registrarti con IBM utilizzando il. AWS Management Console Per ulteriori informazioni, consulta Iscrizione alle inserzioni di Db2 Marketplace e registrazione con IBM.

Per creare un'istanza database tramite l'API Amazon RDS, chiama l'operazione CreateDBInstance.

Per informazioni su ciascuna impostazione, consulta Impostazioni per istanze database.

Impostazioni per istanze database

Nella tabella seguente sono disponibili i dettagli sulle impostazioni che scegli quando crei un'istanza database. La tabella contiene inoltre i motori di database per i quali ogni impostazione è supportata.

È possibile creare un'istanza DB utilizzando la console, il comando create-db-instanceCLI o l'operazione API CreateDBInstance RDS.

Impostazione della console	Descrizione impostazione	Opzione CLI e parametro API di RDS	Motori di database supportati
Allocated storage (Storage allocato)	La quantità di archiviazione, in gigabyte, da allocare per l'istanza database. In alcuni casi, l'allocazione di una maggiore quantità di storage per l'istanza database rispetto alla dimensione del database può migliorare le prestazioni di I/O. Per ulteriori informazioni, consulta Storage delle istanze di database Amazon RDS.	Opzione CLI: `--allocated-storage` Parametro API: `AllocatedStorage`	Tutti

Impostazione della console	Descrizione impostazione	Opzione CLI e parametro API di RDS	Motori di database supportati
Impostazioni dell'architettura	Se scegli l'architettura multitenant Oracle, RDS per Oracle crea un database container (CDB). Se non si sceglie questa opzione, RDS for Oracle crea un non CDB. Un non CDB utilizza l'architettura di database Oracle tradizionale. Un CDB può contenere database collegabili (PDB) laddove un non CDB non può. Oracle Database 21c utilizza solo l'architettura CDB. Oracle Database 19c può utilizzare l'architettura CDB o non CDB. I rilasci inferiori a Oracle Database 19c utilizzano solo l'architettura non CDB. Per ulteriori informazioni, consulta Panoramica dei database CDB RDS per Oracle.	Opzione CLI: `--engine oracle-ee-cdb` (multitenant Oracle) `--engine oracle-se2-cdb` (multitenant Oracle) `--engine oracle-ee` (tradizionale) `--engine oracle-se2` (tradizionale) Parametro API: `Engine`	Oracle

Impostazione della console	Descrizione impostazione	Opzione CLI e parametro API di RDS	Motori di database supportati
Configurazione dell'architettura	Queste impostazioni sono valide solo quando si sceglie l'architettura multitenant Oracle per Impostazioni dell'architettura. Scegli una delle seguenti impostazioni aggiuntive: • Con la configurazione multi-tenant, l'istanza CDB di RDS per Oracle può contenere da 1 a 30 database tenant, a seconda dell'edizione del database e delle eventuali licenze opzionali richieste. Nel contesto di un database Oracle, il database del tenant è un PDB. I PDB di applicazioni e i PDB di proxy non sono supportati. L'istanza database viene creata con 1 database del tenant iniziale. Scegli i valori per Nome del database tenant, Nome utente principale del database tenant, Password principale del database tenant e Set di caratteri del database tenant. La configurazione multi-tenant è permanente. Pertanto, non è possibile riconvertire la configurazione multi-tenant in configurazione a tenant singolo. L'aggiornamento della versione (RU) minima supportato per la configurazione multi-tenant è 19.0.0.0.ru-2022-01.rur-2022.r1.	Opzione CLI: `--multi-tenant` (configurazione multi-tenant) `--no-multi-tenant` (configurazione a tenant singolo) Parametro API: `MultiTenant`	Oracle

Impostazione della console	Descrizione impostazione	Opzione CLI e parametro API di RDS	Motori di database supportati

> **Note**
>
> La funzionalità Amazon RDS è chiamata "multi-tenant" anziché "multitenant" perché è una funzionalità della piattaforma RDS, non solo del motore di database Oracle. Il termine "Oracle multitenant" si riferisce esclusivamente all'architettura del database Oracle, che è compatibile sia con le implementazioni on-premise che con quelle RDS.

- Con la Configurazione a tenant singolo, il CDB RDS per Oracle contiene 1 PDB. Questa è la configurazione predefinita quando si crea un CDB. Non puoi eliminare il PDB iniziale o aggiungere altri PDB. In seguito è possibile convertire la configurazione a tenant singolo del CDB in configurazione multi-tenant, ma non è possibile riconvertirla nuovamente nella configurazione a tenant singolo.

Indipendentemente dalla configurazione scelta, il CDB contiene un unico PDB

Impostazione della console	Descrizione impostazione	Opzione CLI e parametro API di RDS	Motori di database supportati
	iniziale. Nella configurazione multi-tenant, puoi creare più PDB in un secondo momento utilizzando le API RDS. Per ulteriori informazioni, consulta Panoramica dei database CDB RDS per Oracle.		
Auto minor version upgrade (Aggiornamento automatico della versione secondaria)	Scegli Abilita aggiornamento automatico della versione secondaria per consentire all'istanza DB di ricevere automaticamente gli aggiornamenti delle versioni secondarie preferite del motore DB non appena diventano disponibili. Questo è il comportamento che segue di default. Amazon RDS esegue aggiornamenti automatici di versioni secondarie nella finestra di manutenzione. Se non scegli Abilita l'aggiornamento automatico della versione secondaria, l'istanza DB non viene aggiornata automaticamente quando diventano disponibili nuove versioni secondarie. Per ulteriori informazioni, consulta Aggiornamento automatico della versione secondaria del motore.	Opzione CLI: `--auto-minor-version-upgrade` `--no-auto-minor-version-upgrade` Parametro API: `AutoMinorVersionUpgrade`	Tutti

Impostazione della console	Descrizione impostazione	Opzione CLI e parametro API di RDS	Motori di database supportati
Availability zone (Zona di disponibilità)	La zona di disponibilità per l'istanza database. Utilizza il valore predefinito No Preference (Nessuna preferenza) a meno che non desideri specificare una zona di disponibilità. Per ulteriori informazioni, consulta Regioni, zone di disponibilità e Local Zones.	Opzione CLI: `--availability-zone` Parametro API: `AvailabilityZone`	Tutti
AWS KMS key	Disponibile solo se Encryption (Crittografia) è impostato su Enable encryption (Abilita crittografia). Scegliere AWS KMS key da utilizzare per crittografare questa istanza database. Per ulteriori informazioni, consulta Crittografia delle risorse Amazon RDS.	Opzione CLI: `--kms-key-id` Parametro API: `KmsKeyId`	Tutti
Replica di backup	Scegliere Abilita replica in un'altra regione AWS per creare backup in un'altra regione in caso di ripristino di emergenza. Seleziona quindi la regione di destinazione per i backup aggiuntivi.	Non disponibile durante la creazione di un'istanza database. Per informazioni sull'abilitazione dei backup tra regioni utilizzando l'API AWS CLI o RDS, consulta. Abilitazione dei backup automatici tra regioni	Oracle PostgreSQL SQL Server

Impostazione della console	Descrizione impostazione	Opzione CLI e parametro API di RDS	Motori di database supportati
Backup retention period (Periodo di retention dei backup)	Il numero di giorni in cui desideri eseguire il backup automatico dell'istanza database da mantenere. Per un'istanza database non cruciale, impostare questo valore su **1** o su un valore maggiore. Per ulteriori informazioni, consulta Introduzione ai backup.	Opzione CLI: `--backup-retention-period` Parametro API: `BackupRetentionPeriod`	Tutti
Target Backup	Scegli Cloud AWSdi archiviare i backup automatici e le istantanee manuali nella regione principale. AWS Scegliere Outposts (On-Premise) per archiviarli localmente sul tuo Outpost. Questa impostazione di opzione si applica solo a RDS sugli Outposts. Per ulteriori informazioni, consulta Creazione delle istanze database per Amazon RDS su AWS Outposts.	Opzione CLI: `--backup-target` Parametro API: `BackupTarget`	MySQL, PostgreSQL, SQL Server
Backup window (Finestra di backup)	Il periodo di tempo durante il quale Amazon RDS esegue automaticamente un backup dell'istanza database. A meno che non si abbiano preferenze specifiche per l'ora di esecuzione del backup del database, usare il valore predefinito No Preference (Nessuna preferenza). Per ulteriori informazioni, consulta Introduzione ai backup.	Opzione CLI: `--preferred-backup-window` Parametro API: `PreferredBackupWindow`	Tutti

Impostazione della console	Descrizione impostazione	Opzione CLI e parametro API di RDS	Motori di database supportati
Autorità di certificazione	L'autorità di certificazione (CA) per il certificato del server utilizzato dall'istanza database. Per ulteriori informazioni, consulta .	Opzione CLI: `--ca-certificate-identifier` Parametro API RDS: `CACertificateIdentifier`	Tutti

Impostazione della console	Descrizione impostazione	Opzione CLI e parametro API di RDS	Motori di database supportati
Character set (Set di caratteri)	Set di caratteri per l'istanza database. Il valore predefinito di AL32UTF8 per il set di caratteri del database è per il set di caratteri Unicode 5.0 UTF-8 Universal . Non è possibile modificare il set di caratteri DB dopo aver creato l'istanza DB. In una configurazione single-tenant, un set di caratteri DB non predefinito influisce solo sul PDB, non sul CDB. Per ulteriori informazioni, consulta Configurazione a tenant singolo dell'architettura CDB. Il set di caratteri DB è diverso dal set di caratteri nazionale, denominato set di caratteri NCHAR. A differenza del set di caratteri DB, il set di caratteri NCHAR specifica la codifica per le colonne dei tipi di dati NCHAR (NCHAR, NVARCHAR2 e NCLOB) senza influire sui metadati del database. Per ulteriori informazioni, consulta Set di caratteri RDS for Oracle.	Opzione CLI: `--character-set-name` Parametro API: `CharacterSetName`	Oracle

Impostazione della console	Descrizione impostazione	Opzione CLI e parametro API di RDS	Motori di database supportati
Collation (Regola di confronto)	Una regola di confronto a livello di server per l'istanza database. Per ulteriori informazioni, consulta Regola di confronto a livello di server per Microsoft SQL Server.	Opzione CLI: `--character-set-name` Parametro API: `CharacterSetName`	SQL Server
Copy tags to snapshots (Copia tag in snapshot)	Questa opzione consente di copiare i tag dell'istanza database in uno snapshot database quando si crea uno snapshot. Per ulteriori informazioni, consulta Tagging delle risorse Amazon RDS.	Opzione CLI: `--copy-tags-to-snapshot` `--no-copy-tags-to-snapshot` Parametro API RDS: `CopyTagsToSnapshot`	Tutti

Impostazione della console	Descrizione impostazione	Opzione CLI e parametro API di RDS	Motori di database supportati
Database authentication (Autenticazione del database)	L'opzione di autenticazione del database da utilizzare. Scegliere Password authentication (Autenticazione tramite password) per autenticare gli utenti di database solo con le password del database. Scegli Password and IAM DB authentication (Autenticazione tramite password e database IAM) per autenticare gli utenti del database con le password del database e le credenziali utente tramite utenti e ruoli. Per ulteriori informazioni, consulta Autenticazione del database IAM per MariaDB, MySQL e PostgreSQL. Questa opzione è supportata solo per MySQL e PostgreSQL. Scegli Password e autenticazione Kerberos per autenticare gli utenti del database con password del database e l'autenticazione Kerberos tramite un programma creato con. AWS Managed Microsoft AD AWS Directory Service Quindi, scegli la directory o seleziona Create a new Directory (Crea una nuova directory). Per ulteriori informazioni, consulta uno dei seguenti argomenti: •	IAM/ Opzione CLI: `--enable-iam-database-authentication` `--no-enable-iam-database-authentication` Parametro API RDS: `EnableIAMDatabaseAuthentication` Kerberos: Opzione CLI: `--domain` `--domain-iam-role-name` Parametro API RDS: `Domain` `DomainIAMRoleName`	Varia a seconda del tipo di autenticazione

Impostazione della console	Descrizione impostazione	Opzione CLI e parametro API di RDS	Motori di database supportati
	Utilizzo Kerberos dell'autenticazione per RDS for Db2 • Utilizzo dell'autenticazione Kerberos per MySQL • Configurazione dell'autenticazione Kerberos per Amazon RDS for Oracle • Utilizzo di Autenticazione Kerberos con Amazon RDS for PostgreSQL		
Tipo di gestione del database	Scegliere Amazon RDS se non è necessario personalizzare l'ambiente. Scegliere Amazon RDS Custom se si desidera personalizzare il database, il sistema operativo e l'infrastruttura. Per ulteriori informazioni, consulta Utilizzo di Amazon RDS Custom.	Per CLI e l'API, è necessario specifica re il tipo di motore del atabase.	Oracle SQL Server

Impostazione della console	Descrizione impostazione	Opzione CLI e parametro API di RDS	Motori di database supportati
Database port (Porta del database)	La porta che si desidera utilizzare per all'istanza database. La porta predefinita è visualizzata. **Note** I firewall presso alcune aziende bloccano le connessioni alle porte MariaDB, MySQL e PostgreSQL predefinite. Se il firewall della tua azienda blocca la porta predefinita, inserisci un'altra porta per l'istanza database.	Opzione CLI: `--port` Parametro API RDS: `Port`	Tutti
DB engine version (Versione motore del database	Versione del motore del database da utilizzare.	Opzione CLI: `--engine-version` Parametro API RDS: `EngineVersion`	Tutti

Impostazione della console	Descrizione impostazione	Opzione CLI e parametro API di RDS	Motori di database supportati
DB instance class (Classe istanza database)	La configurazione per l'istanza database. Ad esempio, una classe di istanza database db.t3.small ha 2 GB di memoria, 2 vCPU, 1 core virtuale, una ECU variabile e una capacità I/O moderata. Se possibile, scegliere una classe di istanza database sufficientemente ampia da poter tenere in memoria un tipico set di lavoro di query. Quando i set di lavoro sono conservati in memoria, il sistema può evitare di scrivere sul disco, migliorando le prestazioni. Per ulteriori informazioni, consulta Classi di istanze database. In RDS for Oracle, è possibile selezionare l'opzione Includi configurazioni di memoria aggiuntive. Queste configurazioni sono ottimizzate per un elevato rapporto tra memoria e vCPU. Ad esempio: db.r5.6xlarge.tpc2.mem4x è un'istanza database db.r5.8x che ha 2 thread per core (tpc2) e 4 volte (4x) la memoria di un'istanza standard db.r5.6xlarge. Per ulteriori informazioni, consulta Classi di istanza RDS for Oracle.	Opzione CLI: `--db-instance-class` Parametro API RDS: `DBInstanceClass`	Tutti

Impostazione della console	Descrizione impostazione	Opzione CLI e parametro API di RDS	Motori di database supportati
DB instance identifier (Identificatore istanze DB)	Il nome dell'istanza database. Assegna un nome alle istanze database nello stesso modo in cui assegni un nome ai server in locale. L'identificatore dell'istanza DB può contenere fino a 63 caratteri alfanumerici e deve essere univoco per il tuo account nella regione che hai scelto. AWS	Opzione CLI: `--db-instance-identifier` Parametro API RDS: `DBInstanceIdentifier`	Tutti
DB parameter group (Gruppo di parametri database)	Un gruppo di parametri per l'istanza database. Puoi scegliere il gruppo di parametri predefiniti o puoi creare un gruppo di parametri personalizzato. Se utilizzi il modello BYOL per RDS for Db2, prima di creare un'istanza DB, devi prima creare un gruppo di parametri personalizzato che contenga il tuo and. IBM Site ID IBM Customer ID Per ulteriori informazioni, consulta Porta la tua licenza per Db2. Per ulteriori informazioni, consultare Utilizzo di gruppi di parametri.	Opzione CLI: `--db-parameter-group-name` Parametro API RDS: `DBParameterGroupName`	Tutti

Impostazione della console	Descrizione impostazione	Opzione CLI e parametro API di RDS	Motori di database supportati
DB subnet group (Gruppo di sottoreti DB)	Il gruppo di sottoreti database da utilizzare per il cluster di database. Seleziona Choose existing (Scegli esistente) per utilizzare un gruppo di sottoreti database esistente. Quindi scegli il gruppo di sottoreti richiesto dall'elenco a discesa Existing DB subnet groups (Gruppi di sottoreti DB esistenti). Scegli Automatic setup (Configurazione automatica) per consentire a RDS di selezionare un gruppo di sottoreti database compatibile. Se non ne esiste uno, RDS crea un nuovo gruppo di sottoreti per il cluster. Per ulteriori informazioni, consulta Utilizzo di gruppi di sottoreti database.	Opzione CLI: `--db-subnet-group-name` Parametro API RDS: `DBSubnetGroupName`	Tutti
Volume di log dedicato	Utilizza un volume di log dedicato (DLV, Dedicated Log Volume) per archiviare i log delle transazioni del database su un volume di archiviazione separato dal volume contenente le tabelle del database. Per ulteriori informazioni, consulta Utilizzo di un volume di log dedicato (DLV).	Opzione CLI: `--dedicated-log-volume` Parametro API RDS: `DedicatedLogVolume`	Tutti

Impostazione della console	Descrizione impostazione	Opzione CLI e parametro API di RDS	Motori di database supportati
Deletion protection (Protezione da eliminazione)	L'opzione Enable deletion protection (Abilita protezione da eliminazione) permette di impedire l'eliminazione dell'istanza database. Se si crea un'istanza DB di produzione con AWS Management Console, la protezione da eliminazione è abilitata per impostazione predefinita. Per ulteriori informazioni, consulta Eliminazione di un'istanza database.	Opzione CLI: `--deletion-protection` `--no-deletion-protection` Parametro API RDS: `DeletionProtection`	Tutti
Encryption (Crittografia)	Enable encryption (Abilita crittografia) per abilitare la crittografia dei dati inattivi dell'istanza database. Per ulteriori informazioni, consulta Crittografia delle risorse Amazon RDS.	Opzione CLI: `--storage-encrypted` `--no-storage-encrypted` Parametro API RDS: `StorageEncrypted`	Tutti

Impostazione della console	Descrizione impostazione	Opzione CLI e parametro API di RDS	Motori di database supportati
Enhanced Monitoring	L'opzione Enable enhanced monitoring (Abilita monitoraggio avanzato) permette di raccogliere i parametri in tempo reale per il sistema operativo in cui viene eseguita l'istanza database. Per ulteriori informazioni, consulta Monitoraggio dei parametri del sistema operativo con il monitoraggio avanzato.	Opzioni CLI: `--monitoring-interval` `--monitoring-role-arn` Parametri API RDS: `MonitoringInterval` `MonitoringRoleArn`	Tutti
Tipo di motore	Scegli il motore del database da utilizzare per questa istanza.	Opzione CLI: `--engine` Parametro API RDS: `Engine`	Tutti

Impostazione della console	Descrizione impostazione	Opzione CLI e parametro API di RDS	Motori di database supportati
Initial database name (Nome del database iniziale)	Nome del database nell'istanza database. Se non specifichi un nome, Amazon RDS non crea un database nell'istanza database (eccetto per Oracle e PostgreSQL). Il nome non può essere una parola riservata del motore del database e include altri vincoli a seconda del motore del database. Db2: • Deve contenere da 1 a 8 caratteri alfanumerici. • Deve iniziare con a-z, A-Z, @, $ o # ed essere seguito da a-z, A-Z, 0-9, _, @, # o $. • Non può contenere spazi. • Per ulteriori informazioni, consulta Ulteriori considerazioni. MariaDB e MySQL: • Deve contenere da 1 a 64 caratteri alfanumerici. Oracle: •	Opzione CLI: `--db-name` Parametro API RDS: `DBName`	Tutti tranne SQL Server

Impostazione della console	Descrizione impostazione	Opzione CLI e parametro API di RDS	Motori di database supportati
	Deve contenere da 1 a 8 caratteri alfanumerici. • Non può essere `NULL`. Il valore predefinito è `ORCL`. • Deve iniziare con una lettera. PostgreSQL: • Deve contenere da 1 a 63 caratteri alfanumerici. • Deve iniziare con una lettera o un trattino basso. I caratteri successivi possono essere lettere, trattini bassi o cifre (0-9). • Il nome del database iniziale è `postgres`.		

Impostazione della console	Descrizione impostazione	Opzione CLI e parametro API di RDS	Motori di database supportati
Licenza	Valori validi per il modello di licenza: • bring-your-own-licenseo marketplace-license per Db2. • general-public-licenseper MariadB. • license-included per Microsoft SQL Server. • general-public-licenseper MySQL. • incluso in licenza o per Oracle. bring-your-own-license • postgresql-license per PostgreSQL.	Opzione CLI: `--license-model` Parametro API RDS: `LicenseModel`	Tutti
Log exports (Esportazioni log)	I tipi di file di log del database da pubblicare su Amazon CloudWatch Logs. Per ulteriori informazioni, consulta Pubblicazione di log di database su Amazon CloudWatch Logs.	Opzione CLI: `--enable-cloudwatch-logs-exports` Parametro API RDS: `EnableCloudwatchLogsExports`	Tutti

Impostazione della console	Descrizione impostazione	Opzione CLI e parametro API di RDS	Motori di database supportati
Maintenance window (Finestra di manutenzione)	La finestra di 30 minuti entro cui vengono applicate la modifiche in corso all'istanza database. Se il periodo di tempo non è rilevante, scegli No Preference (Nessuna preferenza). Per ulteriori informazioni, consulta Finestra di manutenzione Amazon RDS.	Opzione CLI: `--preferred-maintenance-window` Parametro API RDS: `PreferredMaintenanceWindow`	Tutti
Gestisci le credenziali principali in AWS Secrets Manager	Seleziona Manage master credentials in AWS Secrets Manager (Gestione credenziali master in AWS Secrets Manager) per gestire la password dell'utente master in un segreto di Secrets Manager. Facoltativamente, scegli la chiave KMS da utilizzare per proteggere il segreto. Scegliere tra le chiavi KMS presenti nell'account o inserire la chiave da un altro account. Per ulteriori informazioni, consulta Gestione delle password con Amazon RDS e AWS Secrets Manager.	Opzione CLI: `--manage-master-user-password | --no-manage-master-user-password` `--master-user-secret-kms-key-id` Parametro API RDS: `ManageMasterUserPassword` `MasterUserSecretKmsKeyId`	Tutti

Impostazione della console	Descrizione impostazione	Opzione CLI e parametro API di RDS	Motori di database supportati
Master password (Password master)	La password dell'account utente master. La password contiene il seguente numero di caratteri ASCII stampabili (escluso /, ", uno spazio e @) a seconda del motore del database: • Db2:8—255 • Oracle: da 8 a 30 • MariaDB and MySQL: da 8 a 41 • SQL Server e PostgreSQL: da 8 a 128	Opzione CLI: `--master-user-password` Parametro API RDS: `MasterUserPassword`	Tutti

Impostazione della console	Descrizione impostazione	Opzione CLI e parametro API di RDS	Motori di database supportati
Master username (Nome utente master)	Nome da utilizzare come nome utente principale per accedere all'istanza database con tutti i privilegi del database. Sono valide le seguenti limitazioni di denominazione: • Il nome può contenere da 1 a 16 caratteri alfanumerici e caratteri di sottolineatura. • Il primo carattere deve essere una lettera. • Il nome non può essere una parola riservata del motore di database. Non è possibile modificare il nome utente principale dopo la creazione dell'istanza database. Per Db2, si consiglia di utilizzare lo stesso nome utente principale del nome dell'istanza Db2 autogestita. Per ulteriori informazioni sui privilegi concessi all'utente master, consultare Privilegi dell'account utente master.	Opzione CLI: `--master-username` Parametro API RDS: `MasterUsername`	Tutti

Impostazione della console	Descrizione impostazione	Opzione CLI e parametro API di RDS	Motori di database supportati
Autenticazione Server Windows Microsoft SQL	Abilitare l'autenticazione Microsoft SQL Server Windows, quindi selezionare Browse Directory (Sfoglia directory) per scegliere la directory in cui consentire agli utenti di dominio autorizzati di autenticarsi con questa istanza di SQL Server utilizzando l'autenticazione Windows.	Opzioni CLI: `--domain` `--domain-iam-role-name` Parametri API RDS: `Domain` `DomainIAMRoleName`	SQL Server
Multi-AZ deployment (Implementazione Multi-AZ)	Create a standby instance (Crea un'istanza standby) per creare una replica secondaria passiva dell'istanza database in un'altra zona di disponibilità per il supporto per il failover. Consigliamo Multi-AZ per carichi di lavoro di produzione e per mantenere alta disponibilità. Per lo sviluppo e il testing, è possibile scegliere Do not create a standby instance (Non creare un'istanza database). Per ulteriori informazioni, consulta Configurazione e gestione di un'implementazione multi-AZ.	Opzione CLI: `--multi-az` `--no-multi-az` Parametro API RDS: `MultiAZ`	Tutti

Impostazione della console	Descrizione impostazione	Opzione CLI e parametro API di RDS	Motori di database supportati
Set di caratteri nazionali (NCHAR)	Il set di caratteri nazionali per l'istanza DB, comunemente chiamato set di caratteri NCHAR. È possibile impostare il set di caratteri nazionale su AL16UTF16 (predefinito) o UTF-8. Non è possibile modificare il set di caratteri nazionali dopo aver creato l'istanza DB. Il set di caratteri nazionale è diverso dal set di caratteri DB. A differenza del set di caratteri DB, il set di caratteri nazionale specifica la codifica solo per le colonne dei tipi di dati NCHAR (NCHAR, NVARCHAR2 e NCLOB) senza influire sui metadati del database. Per ulteriori informazioni, consulta Set di caratteri RDS for Oracle.	Opzione CLI: `--nchar-character-set-name` Parametro API: `NcharCharacterSetName`	Oracle

Impostazione della console	Descrizione impostazione	Opzione CLI e parametro API di RDS	Motori di database supportati
Tipo di rete	I protocolli di indirizzo IP supportati dall'istanza database. IPv4 (impostazione di default) per specificare che le risorse possono comunicare con l'istanza database solo tramite il protocollo di indirizzo IP versione 4 (IPv4). Modalità dual-stack per specificare che le risorse possono comunicare con l'istanza database tramite IPv4, IPv6 o entrambi i protocolli. Utilizza la modalità dual-stack se le risorse devono comunicare con l'istanza database tramite il protocollo di indirizzo IPv6. Inoltre, assicurati di associare un blocco CIDR IPv6 a tutte le sottoreti del gruppo di sottoreti DB specificato. Per ulteriori informazioni, consulta Assegnazione di indirizzi IP in Amazon RDS.	Opzione CLI: `--network-type` Parametro API RDS: `NetworkType`	Tutti
Option group (Gruppo di opzioni)	Un gruppo di opzioni per l'istanza database. Puoi scegliere il gruppo di opzioni predefinite o puoi creare un gruppo di opzioni personalizzate. Per ulteriori informazioni, consulta Uso di gruppi di opzioni.	Opzione CLI: `--option-group-name` Parametro API RDS: `OptionGroupName`	Tutti

Impostazione della console	Descrizione impostazione	Opzione CLI e parametro API di RDS	Motori di database supportati
Approfondimenti sulle prestazioni	Enable Performance Insights (Abilita Performance Insights) per monitorare il carico delle istanze database e consentire l'analisi e la risoluzione dei problemi di prestazioni del database. Seleziona un periodo di mantenimento per stabilire quanta cronologia di dati di Performance Insights conservare. L'impostazione del periodo di conservazione nel livello gratuito è Default (7 days) (Impostazione predefinita (7 giorni)). Per mantenere i dati sulle prestazioni più a lungo, specifica da 1 a 24 mesi. Per altre informazioni sui periodi di conservazione, consulta Prezzi e conservazione dei dati per Performance Insights. Scegliere la chiave KMS da utilizzare per proteggere la chiave usata per crittografare questo volume di database. Scegliere tra le chiavi KMS presenti nell'account o inserire la chiave da un altro account. Per ulteriori informazioni, consulta Monitoraggio del carico DB con Performance Insights su Amazon RDS.	Opzioni CLI: `--enable-performance-insights` `--no-enable-performance-insights` `--performance-insights-retention-period` `--performance-insights-kms-key-id` Parametri API RDS: `EnablePerformanceInsights` `PerformanceInsightsRetentionPeriod` `PerformanceInsightsKMSKeyId`	Tutti tranne Db2

Impostazione della console	Descrizione impostazione	Opzione CLI e parametro API di RDS	Motori di database supportati
IOPS con provisioning	Il valore di Provisioned IOPS (operazioni di I/O al secondo) per l'istanza database. Questa impostazione è disponibile solo se scegli una delle seguenti opzioni per Storage type (Tipo di archiviazione): • SSD per uso generico (gp3) • SSD per capacità di IOPS allocata (io1) • SSD IOPS fornito (io2) Per ulteriori informazioni, consulta Storage delle istanze di database Amazon RDS.	Opzione CLI: `--iops` Parametro API RDS: `Iops`	Tutti

Impostazione della console	Descrizione impostazione	Opzione CLI e parametro API di RDS	Motori di database supportati
Accesso pubblico	Yes (Sì) per assegnare all'istanza database un indirizzo IP pubblico, ovvero renderla accessibile al di fuori del VPC. Per essere accessibile pubblicamente, l'istanza database deve anche trovarsi in una sottorete pubblica nel VPC. Seleziona No per rendere accessibile l'istanza database solo dal VPC. Per ulteriori informazioni, consulta Nascondere istanze database in un VPC da Internet. Per connettersi a un'istanza database dall'esterno del proprio VPC, l'istanza database deve essere accessibile pubblicamente. Inoltre, l'accesso deve essere concesso utilizzando le regole in ingresso del gruppo di sicurezza dell'istanza database e devono essere soddisfatti altri requisiti. Per ulteriori informazioni, consulta Impossibile connettersi all'istanza database di Amazon RDS. Se la tua istanza DB non è accessibile pubblicamente, utilizza una connessione AWS VPN da sito a sito o AWS Direct Connect una connessione per accedervi da una rete privata. Per ulteriori informazioni, consulta Riservatezza del traffico Internet.	Opzione CLI: `--publicly-accessible` `--no-publicly-accessible` Parametro API RDS: `PubliclyAccessible`	Tutti

Impostazione della console	Descrizione impostazione	Opzione CLI e parametro API di RDS	Motori di database supportati
Supporto esteso RDS	Seleziona Enable RDS Extended Support per consentire alle principali versioni del motore supportate di continuare a funzionare oltre la data di fine del supporto standard RDS. Quando crei un'istanza DB, Amazon RDS utilizza per impostazione predefinita RDS Extended Support. Per impedire la creazione di una nuova istanza DB dopo la data di fine del supporto standard di RDS e per evitare addebiti per RDS Extended Support, disabilita questa impostazione. Le istanze DB esistenti non verranno addebitati fino alla data di inizio dei prezzi di RDS Extended Support. Per ulteriori informazioni, consulta Utilizzo dell'estensione del supporto per Amazon RDS.	Opzione CLI: `--engine-lifecycle-support` Parametro API RDS: `EngineLifecycleSupport`	MySQL PostgreSQL
Server proxy per RDS	Scegli Create an RDS Proxy (Crea un server proxy per RDS) per creare un proxy per la tua istanza database. Amazon RDS crea automaticamente per il proxy un ruolo IAM e un segreto in Secrets Manager. Per ulteriori informazioni, consulta Utilizzo di Server proxy per Amazon RDS.	Non disponibile durante la creazione di un'istanza database.	MariaDB MySQL PostgreSQL

Impostazione della console	Descrizione impostazione	Opzione CLI e parametro API di RDS	Motori di database supportati
Storage autoscaling (Auto Scaling dello storage)	Enable storage autoscaling (Abilita Auto Scaling) affinché Amazon RDS aumenti automaticamente lo storage quando necessario, così da evitare che l'istanza database termini lo spazio di storage. Maximum storage threshold (Soglia massima di storage) per impostare il limite superiore affinché Amazon RDS aumenti automaticamente lo storage per l'istanza database. Il valore predefinito è 1.000. Per ulteriori informazioni, consulta Gestione della capacità automaticamente con Auto Scaling dello storage Amazon RDS.	Opzione CLI: `--max-allocated-storage` Parametro API RDS: `MaxAllocatedStorage`	Tutti
Velocità di trasmissione effettiva per archiviazione	Il valore della velocità di trasmissione effettiva per archiviazione dell'istanza database. Questa impostazione è disponibile solo se scegli SSD per generico (gp3) come Tipo di archiviazione. Per ulteriori informazioni, consulta archiviazione gp3 (consigliata).	Opzione CLI: `--storage-throughput` Parametro API RDS: `StorageThroughput`	Tutti

Impostazione della console	Descrizione impostazione	Opzione CLI e parametro API di RDS	Motori di database supportati
Storage Type (Tipo di storage)	Il tipo di archiviazione per l'istanza database. Se scegli SSD per uso generico (gp3), puoi aggiungere capacità di IOPS allocata e velocità di trasmissione effettiva per archiviazione in Advanced settings (Impostazioni avanzate). Se scegli Provisioned IOPS SSD (io1) o Provisioned IOPS SSD (io2), inserisci il valore Provisioned IOPS. Per ulteriori informazioni, consulta Tipi di storage Amazon RDS.	Opzione CLI: `--storage-type` Parametro API RDS: `StorageType`	Tutti
Subnet group (Gruppo di sottoreti)	Gruppo di sottoreti database da associare all'istanza database. Per ulteriori informazioni, consulta Utilizzo di gruppi di sottoreti database.	Opzione CLI: `--db-subnet-group-name` Parametro API RDS: `DBSubnetGroupName`	Tutti

Impostazione della console	Descrizione impostazione	Opzione CLI e parametro API di RDS	Motori di database supportati
Nome del database tenant	Il nome del PDB iniziale nella configurazione multi-tenant dell'architettura Oracle. Questa impostazione è disponibile solo se si sceglie Configurazione multi-tenant per Configurazione dell'architettura. Il nome del database del tenant deve essere diverso dal nome del CDB, che è denominato `RDSCDB`. Non è possibile cambiare il nome del CDB.	Opzione CLI: `--db-name` Parametro API RDS: `DBName`	Oracle

Impostazione della console	Descrizione impostazione	Opzione CLI e parametro API di RDS	Motori di database supportati
Nome utente principale del database tenant	Nome da utilizzare come nome utente principale per accedere al database del tenant (PDB) con tutti i privilegi del database. Questa impostazione è disponibile solo se si sceglie Configurazione multi-tenant per Configurazione dell'architettura. Sono valide le seguenti limitazioni di denominazione: • Il nome può contenere da 1 a 16 caratteri alfanumerici e caratteri di sottolineatura. • Il primo carattere deve essere una lettera. • Il nome non può essere una parola riservata del motore di database. Non puoi eseguire le operazioni indicate di seguito: • Modifica il nome utente principale del tenant dopo aver creato il database del tenant. • Accedi con il nome utente principale del tenant al CDB.	Opzione CLI: `--master-username` Parametro API RDS: `MasterUsername`	Oracle

Impostazione della console	Descrizione impostazione	Opzione CLI e parametro API di RDS	Motori di database supportati
Password principale del database tenant	La password per l'account utente principale del database del tenant (PDB). Questa impostazione è disponibile solo se si sceglie Configurazione multi-tenant per Configurazione dell'architettura. La password contiene 8-30 caratteri ASCII stampabili, esclusi `/`, `"`, lo spazio e `@`.	Opzione CLI: `--master-password` Parametro API RDS: `MasterPassword`	Oracle
Set di caratteri del database tenant	Il set di caratteri del database del tenant iniziale. Questa impostazione è disponibile solo se si sceglie Configurazione multi-tenant per Configurazione dell'architettura. Sono supportate solo le istanze CDB RDS per Oracle. Il valore predefinito di AL32UTF8 per il set di caratteri del database del tenant è per il set di caratteri Unicode 5.0 UTF-8 Universal. È possibile scegliere un set di caratteri del database del tenant diverso dal set di caratteri del CDB. Per ulteriori informazioni, consulta Set di caratteri RDS for Oracle.	Opzione CLI: `--character-set-name` Parametro API RDS: `CharacterSetName`	Oracle

Impostazione della console	Descrizione impostazione	Opzione CLI e parametro API di RDS	Motori di database supportati
Set di caratteri nazionali del database tenant	Il set di caratteri nazionali per il database del tenant, comunemente chiamato set di caratteri NCHAR. Questa impostazione è disponibile solo se si sceglie Configurazione multi-tenant per Configurazione dell'architettura. Sono supportate solo le istanze CDB RDS per Oracle. È possibile impostare il set di caratteri nazionale su AL16UTF16 (predefinito) o UTF-8. Non è possibile modificare il set di caratteri nazionali dopo aver creato il database del tenant. Il set di caratteri nazionale del database del tenant è diverso dal set di caratteri del database del tenant. Il set di caratteri nazionali specifica la codifica solo per le colonne che utilizzano il tipo di dati NCHAR (NCHAR, NVARCHAR2 e NCLOB) e non influisce sui metadati del database. Per ulteriori informazioni, consulta Set di caratteri RDS for Oracle.	Opzione CLI: `--nchar-character-set-name` Parametro API: `NcharCharacterSetName`	Oracle

Impostazione della console	Descrizione impostazione	Opzione CLI e parametro API di RDS	Motori di database supportati
Time zone (Fuso orario)	Il fuso orario per l'istanza database. Se non scegli un fuso orario, l'istanza database utilizzerà il fuso orario predefinito. Non è possibile modificare il fuso orario dopo la creazione dell'istanza database. Per ulteriori informazioni, consulta Fuso orario locale per istanze database Amazon RDS per Db2 e Fuso orario locale per le istanze di database di Microsoft SQL Server.	Opzione CLI: `--timezone` Parametro API RDS: `Timezone`	Db2 SQL Server RDS Custom per SQL Server
Virtual Private Cloud (VPC)	Un VPC basato sul servizio Amazon VPC da associare all'istanza database. Per ulteriori informazioni, consulta VPC di Amazon VPC e Amazon RDS.	Per CLI e API, specificare gli ID del gruppo di sicurezza VPC.	Tutti
VPC security group (firewall) (Gruppo di sicurezza VPC (firewall)	I gruppi di sicurezza da associare alle istanze database. Per ulteriori informazioni, consulta Panoramica dei gruppi di sicurezza VPC.	Opzione CLI: `--vpc-security-group-ids` Parametro API RDS: `VpcSecurityGroupIds`	Tutti

Creazione di risorse Amazon RDS con AWS CloudFormation

Amazon RDS è integrato con AWS CloudFormation, un servizio che ti consente di modellare e configurare le tue risorse AWS in modo da dedicare meno tempo alla creazione e alla gestione delle risorse e dell'infrastruttura. È possibile creare un modello che descrive tutte le AWS risorse desiderate (ad esempio istanze DB e gruppi di parametri DB), il AWS CloudFormation provisioning e la configurazione di tali risorse per l'utente.

Quando usi AWS CloudFormation, puoi riutilizzare il modello per configurare le risorse RDS in modo coerente e continuo. Descrivere le risorse una volta e quindi allestisci le stesse risorse più volte in più regioni e account AWS.

RDS e modelli AWS CloudFormation

Per eseguire l'assegnazione e la configurazione delle risorse per RDS e i servizi correlati, devi conoscere i modelli AWS CloudFormation. I modelli sono file di testo formattati in JSON o YAML. Questi modelli descrivono le risorse di cui intendi effettuare il provisioning negli stack AWS CloudFormation. Se non hai familiarità con JSON o YAML, puoi usare AWS CloudFormation Designer per iniziare a utilizzare i modelli AWS CloudFormation. Per ulteriori informazioni, consulta Che cos'è AWS CloudFormation Designer? nella Guida per l'utente di AWS CloudFormation.

RDS supporta la creazione di risorse in AWS CloudFormation. Per ulteriori informazioni, inclusi esempi di modelli JSON e YAML per le risorse, consulta Riferimento dei tipi di risorse RDS nella Guida per l'utente di AWS CloudFormation.

Ulteriori informazioni su AWS CloudFormation

Per ulteriori informazioni su AWS CloudFormation, consulta le seguenti risorse:

- AWS CloudFormation
- Guida per l'utente di AWS CloudFormation
- Documentazione di riferimento dell'API AWS CloudFormation
- Guida per l'utente dell'interfaccia a riga di comando di AWS CloudFormation

Connessione a un'istanza database Amazon RDS

Prima di poter connettersi a un'istanza database, è necessario creare l'istanza database. Per informazioni, consulta Creazione di un'istanza database Amazon RDS. Dopo che Amazon RDS ha fornito l'istanza database, utilizza una qualsiasi applicazione client o utilità MySQL standard per connetterti all'istanza. Nella stringa di connessione, specifica l'indirizzo DNS dell'endpoint dell'istanza database come parametro host. Puoi inoltre specificare il numero di porta dell'endpoint dell'istanza database come parametro porta.

Argomenti

Ricerca delle informazioni di connessione per un'istanza database Amazon RDS

Le informazioni di connessione per un'istanza database includono l'endpoint, la porta e un utente di database valido, ad esempio l'utente master. Ad esempio, per un'istanza di database MySQL, supponiamo che il valore dell'endpoint sia `mydb.123456789012.us-east-1.rds.amazonaws.com`. In questo caso, il valore della porta è `3306` e l'utente del database è `admin`. Date queste informazioni, è possibile specificare i seguenti valori in una stringa di connessione:

- Per host, nome host o nome DNS, specifica `mydb.123456789012.us-east-1.rds.amazonaws.com`.
- Per la porta, specific `3306`.
- Per l'utente, specifica `admin`.

L'endpoint è univoco per ogni istanza database e i valori della porta e dell'utente possono variare. L'elenco seguente mostra la porta più comune per ogni motore DB:

- dB2 — 50000
- MariaDB – 3306
- Microsoft SQL Server – 1433
- MySQL – 3306
- Oracle – 1521
- PostgreSQL – 5432

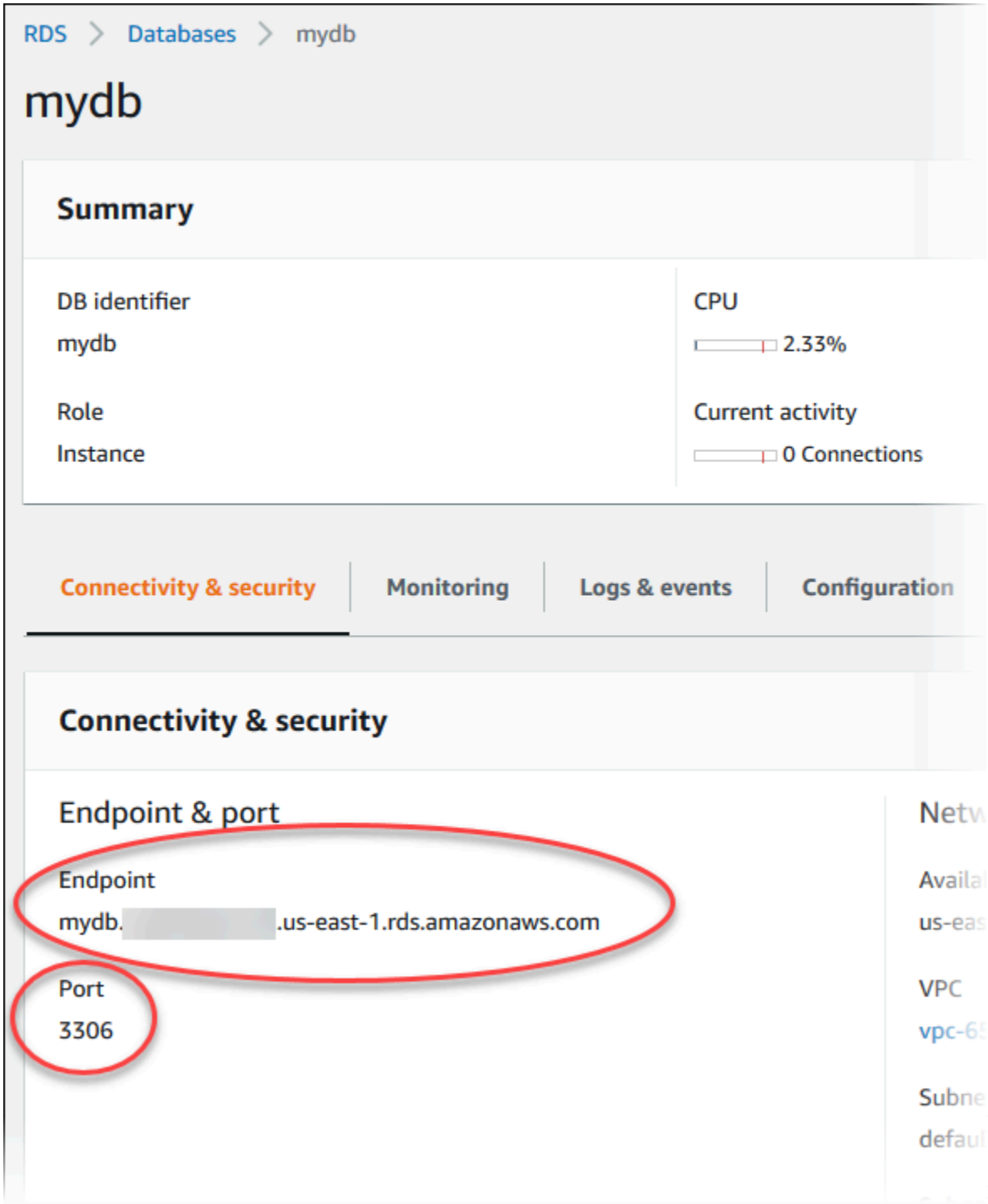
Per connettersi a un'istanza database, utilizzare qualsiasi client per un motore DB. Ad esempio, è possibile utilizzare l'utilità mysql per connettersi a un'istanza database MariaDB o MySQL. È possibile utilizzare Microsoft SQL Server Management Studio per connettersi a un'istanza database di SQL Server. Puoi utilizzare Oracle SQL Developer per connetterti a un'istanza database Oracle. Analogamente puoi utilizzare l'utility a riga di comando psql per connetterti a un'istanza database PostgreSQL.

Per trovare le informazioni di connessione per un'istanza database utilizza la AWS Management ConsoleÈ inoltre possibile utilizzare il describe-db-instancescomando AWS Command Line Interface (AWS CLI) o l'operazione DescribedBInstances dell'API RDS.

Console

Per trovare le informazioni di connessione per un'istanza DB nel AWS Management Console

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di spostamento scegliere Database per visualizzare un elenco delle istanze database.
3. Scegliere il nome dell'istanza database per visualizzarne i dettagli.
4. Nella scheda Connectivity & security (Connettività e sicurezza), copiare l'endpoint. Annotare anche il numero di porta. L'endpoint e il numero di porta sono necessari per la connessione all'istanza database.



5. Se è necessario trovare il nome utente master, scegliere la scheda Configurazione e visualizzare il valore del nome utente principale .

AWS CLI

Per trovare le informazioni di connessione per un'istanza DB utilizzando il AWS CLI, chiama il describe-db-instancescomando. Nella chiamata, eseguire una query per l'ID istanza database, l'endpoint, la porta e il nome utente master.

Per LinuxmacOS, oUnix:

```
aws rds describe-db-instances \
  --query "*[].[DBInstanceIdentifier,Endpoint.Address,Endpoint.Port,MasterUsername]"
```

Per Windows:

```
aws rds describe-db-instances ^
  --query "*[].[DBInstanceIdentifier,Endpoint.Address,Endpoint.Port,MasterUsername]"
```

L'output visualizzato dovrebbe essere simile al seguente.

```
[
    [
        "mydb",
        "mydb.123456789012.us-east-1.rds.amazonaws.com",
        3306,
        "admin"
    ],
    [
        "myoracledb",
        "myoracledb.123456789012.us-east-1.rds.amazonaws.com",
        1521,
        "dbadmin"
    ],
    [
        "mypostgresqldb",
        "mypostgresqldb.123456789012.us-east-1.rds.amazonaws.com",
        5432,
        "postgresadmin"
    ]
]
```

API RDS

Per trovare le informazioni di connessione per un'istanza database utilizzando l'API Amazon RDS, richiamare l'operazione DescribeDBInstances. Nell'output, individuare i valori per l'indirizzo dell'endpoint, la porta dell'endpoint e il nome utente master.

Opzioni di autenticazione del database

Amazon RDS supporta i seguenti modi per autenticare gli utenti del database:

- Autenticazione con password – La tua istanza database esegue tutte le attività di gestione degli account utente. È possibile creare utenti e specificare le password con istruzioni SQL. Le istruzioni SQL che è possibile utilizzare dipendono dal motore DB.
- AWS Identity and Access Management Autenticazione del database (IAM): non è necessario utilizzare una password quando ci si connette a un'istanza DB. Utilizzi invece un token di autenticazione.
- – Autenticazione Kerberos Utilizzare l'autenticazione esterna degli utenti del database utilizzando Kerberos e Microsoft Active Directory. Kerberos è un protocollo di autenticazione di rete che utilizza ticket e crittografia a chiave simmetrica eliminando la necessità di scambiare password sulla rete. È stato integrato in Microsoft Active Directory ed è progettato per autenticare gli utenti su risorse di rete, ad esempio i database.

L’autenticazione con database IAM e Kerberos sono disponibili solo per motori e versioni DB specifici.

Per ulteriori informazioni, consulta Autenticazione del database con Amazon RDS.

Connessioni crittografate

Puoi utilizzare Secure Socket Layer (SSL) o Transport Layer Security (TLS) dall'applicazione per crittografare una connessione a un’istanza database. Ciascun motore database ha il proprio processo per l'implementazione di SSL/TLS. Per ulteriori informazioni, consulta .

Scenari per accedere a un'istanza database in un VPC

Utilizzando Amazon Virtual Private Cloud (Amazon VPC), puoi avviare AWS risorse, come le istanze DB di Amazon RDS, in un cloud privato virtuale (VPC). Quando utilizzi Amazon VPC, hai il controllo

completo sull'ambiente virtuale di rete. Puoi scegliere il tuo intervallo di indirizzi IP, creare sottoreti e configurare liste di routing e di controllo accessi.

Un gruppo di sicurezza VPC controlla l'accesso alle istanze database all'interno di un VPC. Ogni regola del gruppo di sicurezza VPC consente a un'origine specifica di accedere a un'istanza database in un VPC associata a quel gruppo di sicurezza VPC. L'origine può essere una serie di indirizzi (ad esempio, 203.0.113.0/24) oppure un altro gruppo di sicurezza VPC. Specificando un gruppo di sicurezza VPC come origine, consenti il traffico in entrata da tutte le istanze (in genere i server dell'applicazione) che usano il gruppo di sicurezza VPC.

Prima di provare a connettersi all'istanza database, configurare il VPC per il caso d'uso. Di seguito sono riportati gli scenari comuni per accedere a un'istanza database in un VPC:

- Un'istanza database in un VPC a cui si accede da un'istanza Amazon EC2 nello stesso VPC – Un uso comune di un'istanza database in un VPC consiste nel condividere i dati con un server delle applicazioni in esecuzione in un'istanza EC2 nello stesso VPC. L'istanza EC2 potrebbe eseguire un server Web con un'applicazione che interagisce con l'istanza database.
- Un'istanza database in un VPC a cui si accede da un'istanza EC2 in un VPC diverso – In alcuni casi l'istanza database si trova in un VPC diverso dall'istanza EC2 utilizzata per accedervi. In tal caso, puoi utilizzare il peering VPC per accedere all'istanza database.
- Un'istanza database in un VPC a cui si accede da un'applicazione client tramite Internet – Per accedere a un'istanza database in un VPC da un'applicazione client tramite Internet, puoi configurare un VPC con una singola sottorete pubblica e un gateway Internet per consentire la comunicazione tramite Internet.

  Per connettersi a un'istanza database dall'esterno del proprio VPC, l'istanza database deve essere accessibile pubblicamente. Inoltre, l'accesso deve essere concesso utilizzando le regole in ingresso del gruppo di sicurezza dell'istanza database e devono essere soddisfatti altri requisiti. Per ulteriori informazioni, consulta Impossibile connettersi all'istanza database di Amazon RDS.
- Un'istanza database in un VPC a cui si accede da una rete privata: se l'istanza database non è accessibile pubblicamente, puoi utilizzare una delle seguenti opzioni per accedervi da una rete privata:
  - Una AWS connessione VPN da sito a sito
  - AWS Direct Connect Una connessione
  - Una AWS Client VPN connessione

Per ulteriori informazioni, consulta Scenari per accedere a un'istanza database in un VPC.

Connessione alle istanze DB con i driver AWS

La AWS suite di driver è stata progettata per fornire supporto per tempi di switchover e failover più rapidi e l'autenticazione con AWS Secrets Manager, AWS Identity and Access Management (IAM) e Federated Identity. I AWS driver si basano sul monitoraggio dello stato dell'istanza DB e sulla conoscenza della topologia dell'istanza per determinare la nuova istanza primaria. Questo approccio riduce i tempi di switchover e failover a secondi a una cifra, rispetto alle decine di secondi dei driver open source.

La tabella seguente elenca le funzionalità supportate per ciascuno dei driver. Con l'introduzione di nuove funzionalità di servizio, l'obiettivo della AWS suite di driver è disporre di un supporto integrato per tali funzionalità di servizio.

Funzionalità	AWS Driver JDBC	AWS Driver Python
Supporto per il failover	Sì	Sì
Monitoraggio avanzato del failover	Sì	Sì
Suddivisione in lettura/scrittura	Sì	Sì
Connessione ai metadati del driver	Sì	N/D
Telemetria	Sì	Sì
Secrets Manager	Sì	Sì
Autenticazione IAM	Sì	Sì
Identità federata (ADFS)	Sì	Sì
Identità federata (Okta)	Sì	No
Cluster di database Multi-AZ	Sì	Sì

Per ulteriori informazioni sui AWS driver, consulta il driver di lingua corrispondente per la tua istanza DB RDS per MariaDB, RDS per MySQL o RDS per PostgreSQL DB.

> **Note**
>
> Le uniche funzionalità supportate per RDS per MariadB sono l'autenticazione AWS Secrets Manager con AWS Identity and Access Management , (IAM) e l'identità federata.

Connessione a un'istanza DB che esegue un motore DB specifico

Per informazioni sulla connessione a un'istanza database che esegue un motore DB specifico, seguire le istruzioni per il motore DB:

- Connessione all'istanza DB RDS for Db2
- Connessione a un'istanza database che esegue il motore di database MariaDB
- Connessione a un'istanza database che esegua il motore di database di Microsoft SQL Server
- Connessione a un'istanza database che esegue il motore di database di MySQL
- Connessione all'istanza database RDS per Oracle
- Connessione a un'istanza database che esegua il motore di database di PostgreSQL

Gestione delle connessioni con RDS Proxy

Puoi usare il Server proxy per Amazon RDS anche per gestire le connessioni a istanze database RDS per MariaDB, RDS per Microsoft SQL Server, RDS per MySQL o RDS per PostgreSQL. RDS Proxy consente alle applicazioni di effettuare il pool e la condivisione di connessioni di database per migliorare la scalabilità. Per ulteriori informazioni, consulta Utilizzo di Server proxy per Amazon RDS.

Uso di gruppi di opzioni

Alcuni motori di database offrono caratteristiche aggiuntive che semplificano la gestione di dati e database e forniscono ulteriore sicurezza per il database. Amazon RDS utilizza gruppi di opzioni per abilitare e configurare queste funzionalità. Un gruppo di opzioni può specificare le caratteristiche, denominate opzioni, disponibili per una determinata istanza database Amazon RDS. Le opzioni possono includere impostazioni che specificano il funzionamento delle opzioni stesse. Quando associ un'istanza database a un gruppo di opzioni, le opzioni e impostazioni delle opzioni specificate vengono abilitate per l'istanza database in questione.

Amazon RDS supporta opzioni per i motori di database seguenti:

Motore di database	Documentazione di riferimento
`MariaDB`	Opzioni per il motore di database MariaDB
`Microsoft SQL Server`	Opzioni per il motore di database di Microsoft SQL Server
`MySQL`	Opzioni per le istanze database MySQL
`Oracle`	Aggiunta di opzioni alle istanze database Oracle
`PostgreSQL`	PostgreSQL non utilizza opzioni e gruppi di opzioni. PostgreSQL utilizza estensioni e moduli per fornire funzionalità aggiuntive. Per ulteriori informazioni, consulta Versioni con estensione PostgreSQL supportate.

Panoramica dei gruppi di opzioni

Amazon RDS include un gruppo di opzioni predefinito vuoto per ogni nuova istanza database. Non puoi modificare o eliminare questo gruppo di opzioni predefinito, ma qualsiasi nuovo gruppo di opzioni creato deriva le proprie impostazioni dal gruppo di opzioni predefinito. Per applicare un'opzione a un'istanza database, devi eseguire queste operazioni:

1. Creare un nuovo gruppo di opzioni oppure copiare o modificare un gruppo di opzioni esistente.
2. Aggiungere una o più opzioni al gruppo di opzioni.
3. Associare il gruppo di opzioni a questa istanza database.

Per associare un gruppo di opzioni a un'istanza DB, modificare l'istanza DB. Per ulteriori informazioni, consulta Modifica di un'istanza database Amazon RDS.

A un gruppo di opzioni puoi associare sia istanze database sia snapshot del database. In alcuni casi, è possibile eseguire il ripristino da un'istantanea del database o eseguire un point-in-time ripristino per un'istanza DB. In questi casi, il gruppo di opzioni associato allo snapshot del database o all'istanza database viene associato, per default, all'istanza database ripristinata. Puoi associare un gruppo di opzioni diverso a un'istanza database ripristinata. Tuttavia, il nuovo gruppo di opzioni deve contenere tutte le opzioni persistenti o permanenti incluse nel gruppo di opzioni originale. Le opzioni persistenti e permanenti vengono descritte di seguito.

L'esecuzione delle opzioni in una istanza database richiede della memoria aggiuntiva. Per questo motivo, a seconda dell'attuale utilizzo dell'istanza database, potresti dover avviare un'istanza di dimensioni maggiori per usarle. Ad esempio, Oracle Enterprise Manager Database Control utilizza circa 300 MB di RAM. Se abiliti questa opzione per un'istanza DB di piccole dimensioni, potresti riscontrare problemi o out-of-memory errori di prestazioni.

Opzioni persistenti e permanenti

Due tipi di opzioni, le opzioni persistenti e le opzioni permanenti, richiedono un'attenzione speciale quando le aggiungi a un gruppo di opzioni.

Le opzioni persistenti non possono essere rimosse da un gruppo di opzioni mentre le istanze database sono associate al gruppo di opzioni. Un esempio di opzione persistente è l'opzione TDE per la crittografia trasparente dei dati di Microsoft SQL Server. Devi annullare l'associazione di tutte le istanze database dal gruppo di opzioni prima che un'opzione persistente possa essere rimossa dal gruppo di opzioni. In alcuni casi, è possibile ripristinare o eseguire un point-in-time ripristino da un'istantanea del DB. In questi casi, se il gruppo di opzioni associato allo snapshot del database contiene un'opzione persistente, puoi associare al gruppo di opzioni solo l'istanza database ripristinata.

Le opzioni permanenti, come l'opzione TDE per Oracle Advanced Security TDE, non possono mai essere rimosse da un gruppo di opzioni. Puoi modificare il gruppo di opzioni di un'istanza di database che utilizza l'opzione permanente. Tuttavia, il gruppo di opzioni associato all'istanza di database deve includere la stessa opzione permanente. In alcuni casi, è possibile ripristinare o eseguire un point-in-time ripristino da un'istantanea del DB. In questi casi, se il gruppo di opzioni associato allo specifico snapshot del database contiene un'opzione permanente, puoi associare al gruppo di opzioni con tale opzione permanente solo l'istanza database ripristinata.

Per le istanze Oracle DB, puoi copiare gli snapshot DB condivisi che hanno le opzioni `Timezone` o `OLS` (o entrambe). Per eseguire l'operazione, specifica un gruppo di opzioni target che include queste opzioni quando copi lo snapshot DB. L'opzione OLS è permanente e persistente solo per le istanze Oracle DB in esecuzione su Oracle 12.2 o versioni successive. Per ulteriori informazioni su queste opzioni, consulta Fuso orario Oracle e Oracle Label Security.

Considerazioni sul sistema VPC

Il gruppo di opzioni associato all'istanza database è collegato al VPC di tale istanza. Ciò significa che non puoi utilizzare il gruppo di opzioni assegnato a un'istanza database se provi a ripristinare l'istanza in un diverso VPC. Se ripristini un'istanza database su un VPC diverso, puoi eseguire una delle seguenti operazioni:

- Assegnare all'istanza database il gruppo di opzioni predefinito.
- Assegnare un gruppo di opzioni collegato a tale VPC.
- Creare un nuovo gruppo di opzioni e assegnarlo all'istanza database.

Con le opzioni persistenti o permanenti, come Oracle TDE, devi creare un nuovo gruppo di opzioni. Questo gruppo di opzioni deve includere l'opzione persistente o permanente durante il ripristino di un'istanza database in un VPC diverso.

Le impostazioni delle opzioni controllano il comportamento di un'opzione. Ad esempio, l'opzione `NATIVE_NETWORK_ENCRYPTION` di Oracle Advanced Security include un'impostazione che puoi usare per specificare l'algoritmo di crittografia per il traffico di rete da e verso l'istanza database. Alcune impostazioni delle opzioni sono ottimizzate per l'uso con Amazon RDS e non possono essere modificate.

Opzioni reciprocamente esclusive

Alcune opzioni si escludono a vicenda. Puoi usare una o l'altra, ma non entrambe allo stesso tempo. Le opzioni seguenti si escludono a vicenda:

- Oracle Enterprise Manager Database Express e Oracle Management Agent per Enterprise Manager Cloud Control.
- Oracle native network encryption e Oracle Secure Sockets Layer.

Creazione di un gruppo di opzioni

Puoi creare un nuovo gruppo di opzioni che utilizzi le impostazioni del gruppo di opzioni predefinito. Quindi, aggiungi una o più opzioni al nuovo gruppo di opzioni. In alternativa, se esiste già un gruppo di opzioni, puoi copiarlo con tutte le opzioni in un nuovo gruppo di opzioni. Per ulteriori informazioni, consulta Copia di un gruppo di opzioni.

Dopo aver creato un nuovo gruppo di opzioni, questo non include alcuna opzione. Per informazioni su come aggiungere opzioni al gruppo di opzioni, consulta Aggiunta di un'opzione a un gruppo di opzioni. Dopo aver aggiunto le opzioni desiderate, puoi quindi associare il gruppo di opzioni a un'istanza database. In questo modo, le opzioni diventano disponibili nell'istanza database. Per informazioni sull'associazione di un gruppo di opzioni a un'istanza database, consulta la documentazione relativa al motore in uso in Uso di gruppi di opzioni.

Console

Un metodo per creare un gruppo di opzioni consiste nell'usare la AWS Management Console.

Per creare un nuovo gruppo di opzioni tramite la console

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel pannello di navigazione scegliere Option groups (Gruppi di opzioni).
3. Seleziona Create group (Crea gruppo).
4. Nella finestra Create option group (Crea gruppo di opzioni) eseguire queste operazioni:
    a. Per Nome, digita un nome per il gruppo di opzioni univoco all'interno del tuo AWS account. Il nome può includere solo lettere, cifre e trattini.
    b. Per Description (Descrizione) digitare una breve descrizione del gruppo di opzioni. La descrizione viene usata per la visualizzazione.
    c. Per Engine (Motore) scegliere il motore di database desiderato.
    d. Per Major engine version (Versione principale motore) scegliere la versione principale del motore di database desiderato.
5. Per continuare, scegliere Create (Crea). Per annullare l'operazione, invece, scegliere Cancel (Annulla).

AWS CLI

Per creare un gruppo di opzioni, utilizzate il AWS CLI `create-option-group`comando con i seguenti parametri obbligatori.

- `--option-group-name`
- `--engine-name`
- `--major-engine-version`
- `--option-group-description`

Example

L'esempio seguente crea un gruppo di opzioni denominato `testoptiongroup`, associato al motore di database Oracle Enterprise Edition. La descrizione è racchiusa tra virgolette.

Per LinuxmacOS, oUnix:

```
aws rds create-option-group \
    --option-group-name testoptiongroup \
    --engine-name oracle-ee \
    --major-engine-version 12.1 \
    --option-group-description "Test option group"
```

Per Windows:

```
aws rds create-option-group ^
    --option-group-name testoptiongroup ^
    --engine-name oracle-ee ^
    --major-engine-version 12.1 ^
    --option-group-description "Test option group"
```

API RDS

Per creare un gruppo di opzioni, chiamare l'operazione API Amazon RDS `CreateOptionGroup`. Includere i seguenti parametri:

- `OptionGroupName`
- `EngineName`
- `MajorEngineVersion`
- `OptionGroupDescription`

Copia di un gruppo di opzioni

Puoi utilizzare l'API Amazon RDS AWS CLI oppure copiare un gruppo di opzioni. Copiare un gruppo di opzioni può essere utile. Ad esempio quando disponi di un gruppo di opzioni e vuoi includere la maggior parte dei suoi parametri e valori personalizzati in un nuovo gruppo di opzioni. Puoi anche eseguire una copia di un gruppo di opzioni usato nell'ambiente di produzione e quindi modificare la copia per testare altre impostazioni delle opzioni.

> **Note**
>
> Al momento, non puoi copiare un gruppo di opzioni in un'altra AWS regione.

AWS CLI

Per copiare un gruppo di opzioni, usa il AWS CLI copy-option-groupcomando. Includi le seguenti opzioni obbligatorie:

- `--source-option-group-identifier`
- `--target-option-group-identifier`
- `--target-option-group-description`

Example

L'esempio seguente crea un gruppo di opzioni denominato `new-option-group`, che è una copia locale del gruppo di opzioni `my-option-group`.

Per LinuxmacOS, oUnix:

```
aws rds copy-option-group \
    --source-option-group-identifier my-option-group \
    --target-option-group-identifier new-option-group \
```

```
    --target-option-group-description "My new option group"
```

Per Windows:

```
aws rds copy-option-group ^
    --source-option-group-identifier my-option-group ^
    --target-option-group-identifier new-option-group ^
    --target-option-group-description "My new option group"
```

API RDS

Per copiare un gruppo di opzioni, chiama l'CopyOptionGroupoperazione dell'API Amazon RDS. Includi i parametri obbligatori seguenti.

- `SourceOptionGroupIdentifier`
- `TargetOptionGroupIdentifier`
- `TargetOptionGroupDescription`

Aggiunta di un'opzione a un gruppo di opzioni

Puoi aggiungere un'opzione a un gruppo di opzioni esistente. Dopo aver aggiunto le opzioni desiderate, puoi quindi associare il gruppo di opzioni a un'istanza database in modo che le opzioni diventino disponibili nell'istanza database. Per informazioni sull'associazione di un gruppo di opzioni a un'istanza database, consulta la documentazione per il motore di database specifico, elencato in Uso di gruppi di opzioni.

Le modifiche del gruppo di opzioni devono essere applicate immediatamente in due casi:

- Quando aggiungi un'opzione che aggiunge o aggiorna un valore di porta, ad esempio l'opzione `OEM`.
- Quando aggiungi o rimuovi un gruppo di opzioni con un'opzione che include un valore di porta.

In questi casi, scegli l'opzione Apply immediately (Applica immediatamente) nella console. Altrimenti puoi includere l'opzione `--apply-immediately` quando usi la AWS CLI o impostare il parametro `ApplyImmediately` su `true` quando usi l'API di Amazon RDS. Le opzioni che non includono valori di porta possono essere applicate immediatamente oppure durante la finestra di manutenzione successiva per l'istanza database.

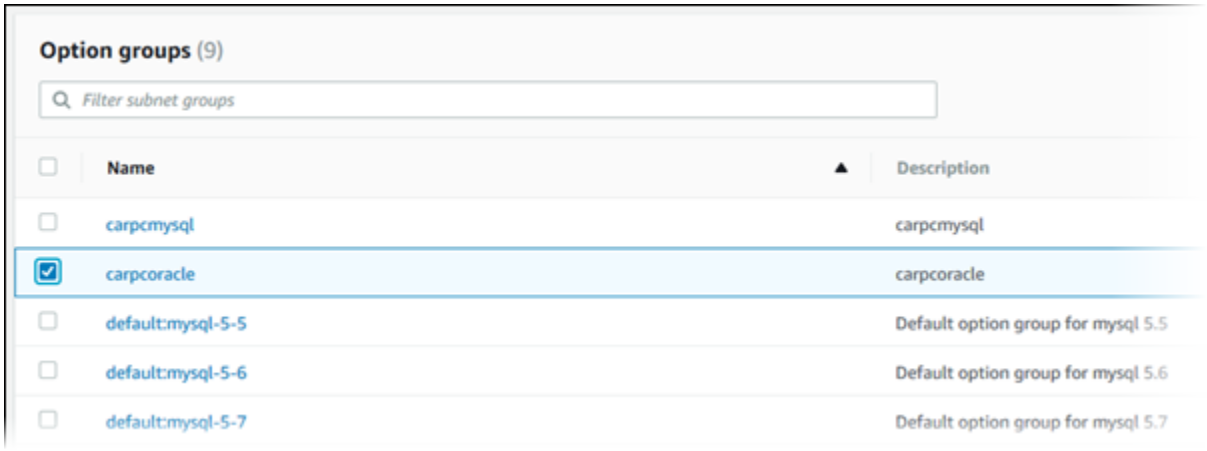
> **Note**
>
> Se specifichi un gruppo di sicurezza come valore per un'opzione di un gruppo di opzioni, puoi gestire il gruppo di sicurezza modificando il gruppo di opzioni. Non è possibile modificare o rimuovere questo gruppo di sicurezza modificando un'istanza database. Inoltre, il gruppo di sicurezza non viene visualizzato nei dettagli dell'istanza DB AWS Management Console né nell'output del AWS CLI comando`describe-db-instances`.

Console

È possibile utilizzare il AWS Management Console per aggiungere un'opzione a un gruppo di opzioni.

Per aggiungere un'opzione a un gruppo di opzioni tramite la console

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione scegliere Option groups (Gruppi di opzioni).
3. Selezionare il gruppo di opzioni che si vuole modificare, quindi scegliere Add Option (Aggiungi opzione).

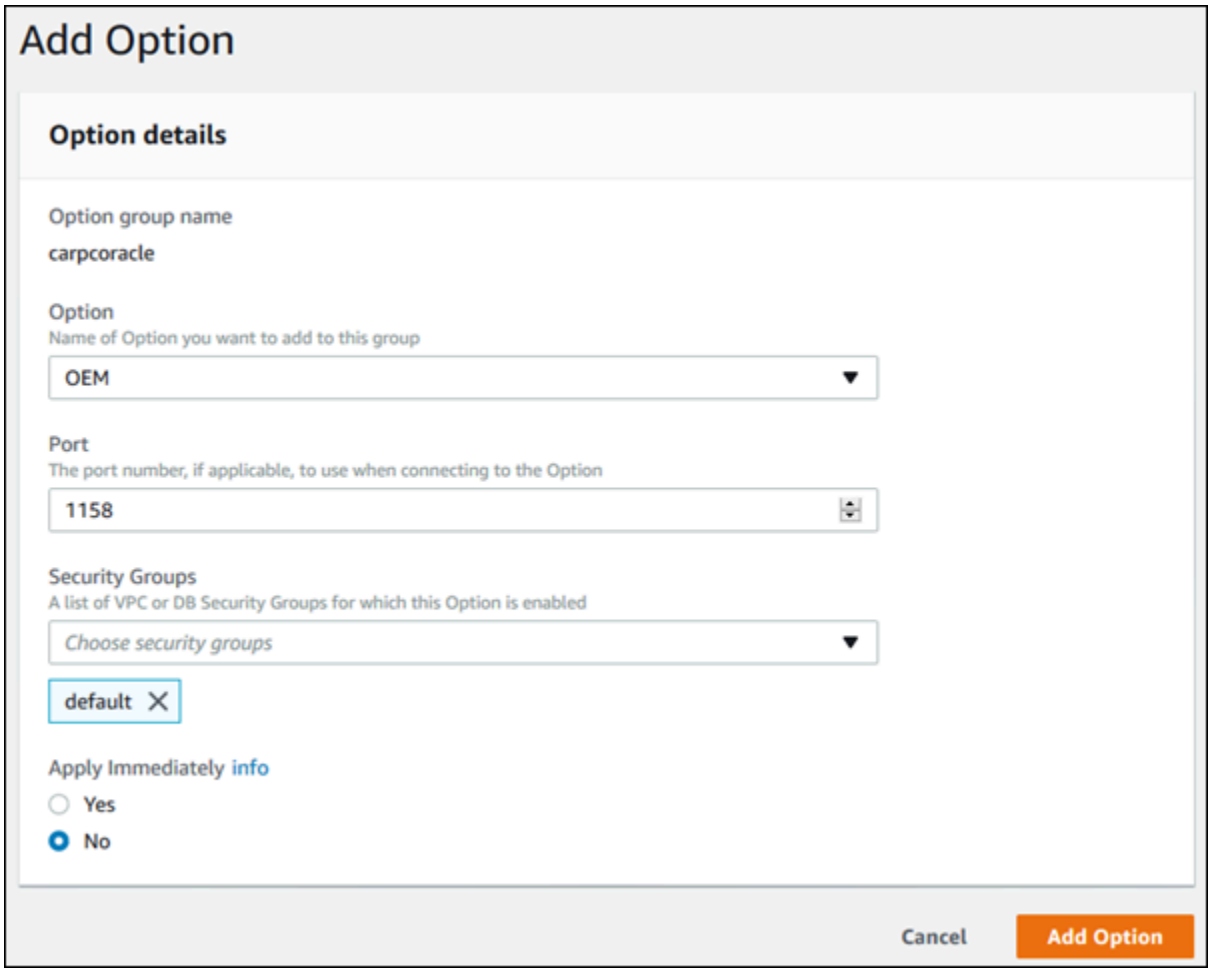


4. Nella finestra Add option (Aggiungi opzione) eseguire queste operazioni:

   a. Scegliere l'opzione che si vuole aggiungere. Potrebbe essere necessario fornire altri valori, a seconda dell'opzione selezionata. Ad esempio, se si sceglie l'opzione `OEM`, è necessario digitare anche un valore di porta e specificare un gruppo di sicurezza.

   b. Per abilitare l'opzione in tutte le istanze database associate non appena viene aggiunta, per Apply Immediately (Applica immediatamente) scegliere Yes (Sì). Se si sceglie No

(impostazione predefinita), l'opzione viene abilitata per ogni istanza database associata durante la finestra di manutenzione successiva.



5. Dopo aver selezionato le impostazioni desiderate, selezionare Add Option (Aggiungi opzione).

AWS CLI

Per aggiungere un'opzione a un gruppo di opzioni, esegui il comando AWS CLI add-option-to-option-group con l'opzione che desideri aggiungere. Per abilitare la nuova opzione immediatamente in tutte le istanze database associate, includi il parametro `--apply-immediately`. Per impostazione predefinita, l'opzione viene abilitata per ogni istanza database associata durante la finestra di manutenzione successiva. Includi il seguente parametro obbligatorio:

- `--option-group-name`

Example

L'esempio seguente aggiunge l'`Timezone`opzione, con l'`America/Los_Angeles`impostazione, a un gruppo di opzioni denominato `testoptiongroup` e la abilita immediatamente.

Per LinuxmacOS, oUnix:

```
aws rds add-option-to-option-group \
    --option-group-name testoptiongroup \
    --options "OptionName=Timezone,OptionSettings=[{Name=TIME_ZONE,Value=America/
Los_Angeles}]" \
    --apply-immediately
```

Per Windows:

```
aws rds add-option-to-option-group ^
    --option-group-name testoptiongroup ^
    --options "OptionName=Timezone,OptionSettings=[{Name=TIME_ZONE,Value=America/
Los_Angeles}]" ^
    --apply-immediately
```

L'output del comando è simile al seguente:

```
...{
       "OptionName": "Timezone",
       "OptionDescription": "Change time zone",
       "Persistent": true,
       "Permanent": false,
       "OptionSettings": [
           {
               "Name": "TIME_ZONE",
               "Value": "America/Los_Angeles",
               "DefaultValue": "UTC",
               "Description": "Specifies the timezone the user wants to change the
 system time to",
               "ApplyType": "DYNAMIC",
               "DataType": "STRING",
               "AllowedValues": "Africa/Cairo,...",
               "IsModifiable": true,
               "IsCollection": false
           }
       ],
       "DBSecurityGroupMemberships": [],
```

```
        "VpcSecurityGroupMemberships": []
    }...
```

Example

Nell'esempio seguente viene aggiunta l'opzione Oracle OEM a un gruppo di opzioni. Vengono specificati inoltre una porta personalizzata e una coppia di gruppi di sicurezza VPC Amazon EC2 da utilizzare per quella porta.

Per LinuxmacOS, oUnix:

```
aws rds add-option-to-option-group \
    --option-group-name testoptiongroup \
    --options OptionName=OEM,Port=5500,VpcSecurityGroupMemberships="sg-test1,sg-test2"
 \
    --apply-immediately
```

Per Windows:

```
aws rds add-option-to-option-group ^
    --option-group-name testoptiongroup ^
    --options OptionName=OEM,Port=5500,VpcSecurityGroupMemberships="sg-test1,sg-test2"
 ^
    --apply-immediately
```

L'output del comando è simile al seguente:

```
OPTIONGROUP  False  oracle-ee  12.1 arn:aws:rds:us-east-1:1234567890:og:testoptiongroup
   Test Option Group  testoptiongroup vpc-test
OPTIONS Oracle 12c EM Express   OEM     False   False   5500
VPCSECURITYGROUPMEMBERSHIPS     active  sg-test1
VPCSECURITYGROUPMEMBERSHIPS     active  sg-test2
```

Example

L'esempio seguente aggiunge l'opzione Oracle `NATIVE_NETWORK_ENCRYPTION` a un gruppo di opzioni e specifica le impostazioni delle opzioni. Se non sono specificate impostazioni, vengono usati i valori predefiniti.

Per LinuxmacOS, oUnix:

```
aws rds add-option-to-option-group \
```

```
    --option-group-name testoptiongroup \
    --options '[{"OptionSettings":
[{"Name":"SQLNET.ENCRYPTION_SERVER","Value":"REQUIRED"},
{"Name":"SQLNET.ENCRYPTION_TYPES_SERVER","Value":"AES256,AES192,DES"}],"OptionName":"NATIVE_NET
 \
    --apply-immediately
```

Per Windows:

```
aws rds add-option-to-option-group ^
    --option-group-name testoptiongroup ^
    --options "OptionSettings"=[{"Name"="SQLNET.ENCRYPTION_SERVER","Value"="REQUIRED"},
{"Name"="SQLNET.ENCRYPTION_TYPES_SERVER","Value"="AES256\,AES192\,DES"}],"OptionName"="NATIVE_N
 ^
    --apply-immediately
```

L'output del comando è simile al seguente:

```
...{
    "OptionName": "NATIVE_NETWORK_ENCRYPTION",
    "OptionDescription": "Native Network Encryption",
    "Persistent": false,
    "Permanent": false,
    "OptionSettings": [
        {
            "Name": "SQLNET.ENCRYPTION_TYPES_SERVER",
             "Value": "AES256,AES192,DES",
            "DefaultValue":
 "RC4_256,AES256,AES192,3DES168,RC4_128,AES128,3DES112,RC4_56,DES,RC4_40,DES40",
            "Description": "Specifies list of encryption algorithms in order of
 intended use",
            "ApplyType": "STATIC",
            "DataType": "STRING",
            "AllowedValues":
 "RC4_256,AES256,AES192,3DES168,RC4_128,AES128,3DES112,RC4_56,DES,RC4_40,DES40",
            "IsModifiable": true,
            "IsCollection": true
        },
        {
            "Name": "SQLNET.ENCRYPTION_SERVER",
            "Value": "REQUIRED",
            "DefaultValue": "REQUESTED",
            "Description": "Specifies the desired encryption behavior",
```

```
            "ApplyType": "STATIC",
            "DataType": "STRING",
            "AllowedValues": "ACCEPTED,REJECTED,REQUESTED,REQUIRED",
            "IsModifiable": true,
            "IsCollection": false
        },...
```

API RDS

Per aggiungere un'opzione a un gruppo di opzioni utilizzando l'API Amazon RDS, chiama l'ModifyOptionGroupoperazione con l'opzione che desideri aggiungere. Per abilitare immediatamente la nuova opzione in tutte le istanze database associate, includi il parametro `ApplyImmediately` e impostalo su `true`. Per impostazione predefinita, l'opzione viene abilitata per ogni istanza database associata durante la finestra di manutenzione successiva. Includi il seguente parametro obbligatorio:

- `OptionGroupName`

Generazione di un elenco delle opzioni e delle impostazioni delle opzioni per un gruppo di opzioni

Puoi elencare tutte le opzioni e le impostazioni delle opzioni per un gruppo di opzioni.

Console

Puoi usare il AWS Management Console per elencare tutte le opzioni e le impostazioni delle opzioni per un gruppo di opzioni.

Per elencare le opzioni e le impostazioni delle opzioni per un gruppo di opzioni

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione scegliere Option groups (Gruppi di opzioni).
3. Scegliere il nome del gruppo di opzioni per visualizzarne i dettagli. Le opzioni e le relative impostazioni nel gruppo di opzioni sono elencate.

AWS CLI

Per elencare le opzioni e le impostazioni delle opzioni per un gruppo di opzioni, usa il AWS CLI `describe-option-groups`comando. Specifica il nome del gruppo di opzioni le cui opzioni e

impostazioni vuoi visualizzare. Se non specifichi il nome del gruppo di opzioni, vengono descritti tutti i gruppi di opzioni.

Example

L'esempio seguente elenca le opzioni e le impostazioni delle opzioni per tutti i gruppi di opzioni.

```
aws rds describe-option-groups
```

Example

L'esempio seguente elenca le opzioni e le impostazioni delle opzioni per un gruppo di opzioni denominato `testoptiongroup`.

```
aws rds describe-option-groups --option-group-name testoptiongroup
```

API RDS

Per elencare le opzioni e le impostazioni delle opzioni per un gruppo di opzioni, utilizzare l'operazione API Amazon RDS `DescribeOptionGroups`. Specifica il nome del gruppo di opzioni le cui opzioni e impostazioni vuoi visualizzare. Se non specifichi il nome del gruppo di opzioni, vengono descritti tutti i gruppi di opzioni.

Modifica di un'impostazione di un'opzione

Dopo aver aggiunto un'opzione che include impostazioni modificabili, puoi modificare le impostazioni in qualsiasi momento. Se modifichi opzioni o impostazioni di opzioni in un gruppo di opzioni, le modifiche vengono applicate a tutte le istanze database associate al gruppo di opzioni. Per ulteriori informazioni sulle impostazioni disponibili per le diverse opzioni, consulta la documentazione per il motore in uso in Uso di gruppi di opzioni.

Le modifiche del gruppo di opzioni devono essere applicate immediatamente in due casi:

- Quando aggiungi un'opzione che aggiunge o aggiorna un valore di porta, ad esempio l'opzione `OEM`.
- Quando aggiungi o rimuovi un gruppo di opzioni con un'opzione che include un valore di porta.

In questi casi, scegli l'opzione Apply immediately (Applica immediatamente) nella console. Altrimenti puoi includere l'opzione `--apply-immediately` quando usi la AWS CLI o impostare il parametro

`ApplyImmediately` su `true` quando usi l'API di RDS. Le opzioni che non includono valori di porta possono essere applicate immediatamente oppure durante la finestra di manutenzione successiva per l'istanza database.

> **Note**
>
> Se si specifica un gruppo di sicurezza come valore per un'opzione in un gruppo di opzioni, è possibile gestire il gruppo di sicurezza modificando il gruppo di opzioni. Non è possibile modificare o rimuovere questo gruppo di sicurezza modificando un'istanza database. Inoltre, il gruppo di sicurezza non viene visualizzato nei dettagli dell'istanza DB AWS Management Console né nell'output del AWS CLI comando`describe-db-instances`.

Console

È possibile utilizzare il AWS Management Console per modificare l'impostazione di un'opzione.

Per modificare un'impostazione di un'opzione tramite la console

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione scegliere Option groups (Gruppi di opzioni).
3. Selezionare il gruppo di opzioni di cui si vuole modificare un'opzione e quindi scegliere Modify option (Modifica opzione).
4. Nella finestra Modify option scegliere l'opzione di cui si vuole modificare un'impostazione in Installed Options (Opzioni installate). Apportare le modifiche desiderate.
5. Per abilitare l'opzione non appena viene aggiunta, per Apply Immediately (Applica immediatamente) scegliere Yes (Sì). Se si sceglie No (impostazione predefinita), l'opzione viene abilitata per ogni istanza database associata durante la finestra di manutenzione successiva.
6. Dopo aver selezionato le impostazioni desiderate, scegliere Modify Option (Modifica opzione).

AWS CLI

Per modificare l'impostazione di un'opzione, usa il AWS CLI `add-option-to-option-group`comando con il gruppo di opzioni e l'opzione che desideri modificare. Per impostazione predefinita, l'opzione viene abilitata per ogni istanza database associata durante la finestra di manutenzione successiva. Per applicare immediatamente la modifica a tutte le istanze database

associate, includi il parametro `--apply-immediately`. Per modificare un'impostazione di un'opzione, usa l'argomento `--settings`.

Example

L'esempio seguente modifica la porta usata da Oracle Enterprise Manager Database Control (OEM) in un gruppo di opzioni denominato `testoptiongroup`, applicando immediatamente la modifica.

Per LinuxmacOS, oUnix:

```
aws rds add-option-to-option-group \
    --option-group-name testoptiongroup \
    --options OptionName=OEM,Port=5432,DBSecurityGroupMemberships=default \
    --apply-immediately
```

Per Windows:

```
aws rds add-option-to-option-group ^
    --option-group-name testoptiongroup ^
    --options OptionName=OEM,Port=5432,DBSecurityGroupMemberships=default ^
    --apply-immediately
```

L'output del comando è simile al seguente:

```
OPTIONGROUP   False  oracle-ee  12.1  arn:aws:rds:us-
east-1:1234567890:og:testoptiongroup   Test Option Group    testoptiongroup
OPTIONS Oracle 12c EM Express   OEM     False   False   5432
DBSECURITYGROUPMEMBERSHIPS   default  authorized
```

Example

L'esempio seguente modifica l'opzione Oracle `NATIVE_NETWORK_ENCRYPTION` e ne modifica le impostazioni.

Per LinuxmacOS, oUnix:

```
aws rds add-option-to-option-group \
    --option-group-name testoptiongroup \
    --options '[{"OptionSettings":
[{"Name":"SQLNET.ENCRYPTION_SERVER","Value":"REQUIRED"},
{"Name":"SQLNET.ENCRYPTION_TYPES_SERVER","Value":"AES256,AES192,DES,RC4_256"}],"OptionName":"NA
 \
```

```
    --apply-immediately
```

Per Windows:

```
aws rds add-option-to-option-group ^
    --option-group-name testoptiongroup ^
    --options "OptionSettings"=[{"Name"="SQLNET.ENCRYPTION_SERVER","Value"="REQUIRED"},
{"Name"="SQLNET.ENCRYPTION_TYPES_SERVER","Value"="AES256\,AES192\,DES
\,RC4_256"}],"OptionName"="NATIVE_NETWORK_ENCRYPTION" ^
    --apply-immediately
```

L'output del comando è simile al seguente:

```
OPTIONGROUP   False  oracle-ee  12.1  arn:aws:rds:us-
east-1:1234567890:og:testoptiongroup   Test Option Group    testoptiongroup

OPTIONS Oracle Advanced Security - Native Network Encryption
 NATIVE_NETWORK_ENCRYPTION       False   False
OPTIONSETTINGS
 RC4_256,AES256,AES192,3DES168,RC4_128,AES128,3DES112,RC4_56,DES,RC4_40,DES40 STATIC
 STRING
   RC4_256,AES256,AES192,3DES168,RC4_128,AES128,3DES112,RC4_56,DES,RC4_40,DES40
 Specifies list of encryption algorithms in order of intended use
   True     True    SQLNET.ENCRYPTION_TYPES_SERVER  AES256,AES192,DES,RC4_256
OPTIONSETTINGS  ACCEPTED,REJECTED,REQUESTED,REQUIRED    STATIC  STRING  REQUESTED
 Specifies the desired encryption behavior  False   True  SQLNET.ENCRYPTION_SERVER
 REQUIRED
OPTIONSETTINGS  SHA1,MD5   STATIC  STRING  SHA1,MD5    Specifies list of
 checksumming algorithms in order of intended use      True    True
 SQLNET.CRYPTO_CHECKSUM_TYPES_SERVER  SHA1,MD5
OPTIONSETTINGS  ACCEPTED,REJECTED,REQUESTED,REQUIRED  STATIC  STRING
 REQUESTED     Specifies the desired data integrity behavior   False   True
 SQLNET.CRYPTO_CHECKSUM_SERVER  REQUESTED
```

API RDS

Per modificare un'impostazione di un'opzione, utilizzare l'operazione API Amazon RDS `ModifyOptionGroup` con il gruppo di opzioni e l'opzione da modificare. Per impostazione predefinita, l'opzione viene abilitata per ogni istanza database associata durante la finestra di manutenzione successiva. Per applicare immediatamente la modifica a tutte le istanze database associate, includi il parametro `ApplyImmediately` e impostalo su `true`.

Rimozione di un'opzione da un gruppo di opzioni

Non tutte le opzioni possono essere rimosse da un gruppo di opzioni. Un'opzione persistente non può essere rimossa da un gruppo di opzioni finché non annulli l'associazione di tutte le istanze database associate al gruppo di opzioni. Un'opzione permanente non può essere mai rimossa da un gruppo di opzioni. Per ulteriori informazioni su quali opzioni sono rimovibili, consulta la documentazione per il motore specifico, elencato in Uso di gruppi di opzioni.

Se rimuovi tutte le opzioni da un gruppo di opzioni, Amazon RDS non elimina il gruppo di opzioni. Le istanze database associate al gruppo di opzioni vuoto continuano a essere associate ad esso, ma semplicemente non hanno alcuna opzione attiva. In alternativa, per rimuovere tutte le opzioni da un'istanza database, puoi associare l'istanza database al gruppo di opzioni (vuoto) predefinito.

Console

È possibile utilizzare il AWS Management Console per rimuovere un'opzione da un gruppo di opzioni.

Per rimuovere un'opzione da un gruppo di opzioni tramite la console

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione scegliere Option groups (Gruppi di opzioni).
3. Selezionare il gruppo di opzioni che si vuole rimuovere e quindi scegliere Delete option (Elimina opzione).
4. Nella finestra Delete option (Elimina opzione) eseguire queste operazioni:
   - Selezionare la casella di controllo per l'opzione che si vuole eliminare.
   - Perché l'eliminazione abbia effetto non appena viene impostata, per Apply Immediately (Applica immediatamente) scegliere Yes (Sì). Se si sceglie No (impostazione predefinita), l'opzione viene eliminata per ogni istanza database associata durante la finestra di manutenzione successiva.

5. Dopo aver selezionato tutte le impostazioni desiderate, scegliere Yes, Delete (Sì, elimina).

AWS CLI

Per rimuovere un'opzione da un gruppo di opzioni, usa il AWS CLI `remove-option-from-option-group`comando con l'opzione che desideri eliminare. Per impostazione predefinita, l'opzione viene rimossa da ogni istanza database associata durante la finestra di manutenzione successiva. Per applicare immediatamente la modifica, includi il parametro `--apply-immediately`.

Example

L'esempio seguente rimuove l'opzione Oracle Enterprise Manager Database Control (OEM) da un gruppo di opzioni denominato `testoptiongroup`, applicando immediatamente la modifica.

Per LinuxmacOS, oUnix:

```
aws rds remove-option-from-option-group \
    --option-group-name testoptiongroup \
    --options OEM \
    --apply-immediately
```

Per Windows:

```
aws rds remove-option-from-option-group ^
    --option-group-name testoptiongroup ^
    --options OEM ^
    --apply-immediately
```

L'output del comando è simile al seguente:

```
OPTIONGROUP    testoptiongroup oracle-ee   12.1    Test option group
```

API RDS

Per rimuovere un'opzione da un gruppo di opzioni, utilizza l'operazione `ModifyOptionGroup` dell'API Amazon RDS. Per impostazione predefinita, l'opzione viene rimossa da ogni istanza database associata durante la finestra di manutenzione successiva. Per applicare immediatamente la modifica, includi il parametro `ApplyImmediately` e impostalo su `true`.

Includere i seguenti parametri:

- `OptionGroupName`
- `OptionsToRemove.OptionName`

Eliminazione di un gruppo di opzioni

È possibile eliminare un gruppo di opzioni solo se soddisfa i seguenti criteri:

- Non è associato a nessuna risorsa Amazon RDS. Un gruppo di opzioni può essere associato a un'istanza database oppure a uno snapshot del database manuale o automatico.
- Non è un gruppo di opzioni predefinito.

Per identificare i gruppi di opzioni utilizzati dalle istanze DB e dagli snapshot DB, è possibile utilizzare i seguenti comandi CLI:

```
aws rds describe-db-instances \
    --query 'DBInstances[*].
[DBInstanceIdentifier,OptionGroupMemberships[].OptionGroupName]'

aws rds describe-db-snapshots | jq -r '.DBSnapshots[] | "\(.DBInstanceIdentifier),
\(.OptionGroupName)"' | sort | uniq
```

Se provi a eliminare un gruppo di opzioni associato a una risorsa RDS, viene restituito un errore simile al seguente.

```
An error occurred (InvalidOptionGroupStateFault) when calling the DeleteOptionGroup
 operation: The option group 'optionGroupName' cannot be deleted because it is in use.
```

Per trovare le risorse Amazon RDS associate a un gruppo di opzioni

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione scegliere Option groups (Gruppi di opzioni).
3. Scegliere il nome del gruppo di opzioni per visualizzarne i dettagli.
4. Controllare la presenza delle risorse Amazon RDS associate nella sezione Associated Instances and Snapshots (Istanze e snapshot associati).

Se al gruppo di opzioni è associata un'istanza database, modificare l'istanza database affinché utilizzi un gruppo di opzioni diverso. Per ulteriori informazioni, consulta Modifica di un'istanza database Amazon RDS.

Se al gruppo di opzioni è associato uno snapshot di database manuale, modifica lo snapshot DB per utilizzare un gruppo di opzioni diverso. Puoi farlo usando il AWS CLI `modify-db-snapshot`comando.

> **Note**
>
> Non puoi modificare il gruppo di opzioni di uno snapshot del database automatico.

Console

Una possibilità per eliminare un gruppo di opzioni consiste nell'usare la AWS Management Console.

Eliminare un gruppo di opzioni utilizzando la console.

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.

2. Nel riquadro di navigazione scegliere Option groups (Gruppi di opzioni).
3. Scegliere il gruppo di opzioni.
4. Scegliere Delete group (Elimina gruppo).
5. Nella pagina di conferma, scegli Delete (Elimina) per completare l'eliminazione del gruppo di opzioni oppure scegliere Cancel (Annulla) per annullare l'eliminazione.

AWS CLI

Per eliminare un gruppo di opzioni, usa il AWS CLI `delete-option-group`comando con il seguente parametro obbligatorio.

- `--option-group-name`

Example

L'esempio seguente illustra come eliminare un gruppo di opzioni denominato `testoptiongroup`.

Per LinuxmacOS, oUnix:

```
aws rds delete-option-group \
    --option-group-name testoptiongroup
```

Per Windows:

```
aws rds delete-option-group ^
    --option-group-name testoptiongroup
```

API RDS

Per eliminare un gruppo di opzioni, richiamare l'operazione API Amazon RDS `DeleteOptionGroup`. Includere il seguente parametro:

- `OptionGroupName`

Utilizzo di gruppi di parametri

Parametri database specificano la modalità di configurazione del database. Ad esempio, i parametri del database possono specificare la quantità di risorse, come la memoria, da allocare a un database.

È possibile gestire la configurazione del database associando le istanze database e i cluster di database Multi-AZ con i gruppi di parametri. Amazon RDS definisce i gruppi di parametri con le impostazioni di default. Puoi definire i gruppi di parametri anche con le impostazioni personalizzate.

> **Note**
>
> Alcuni motori di database offrono funzionalità aggiuntive che è possibile aggiungere al database come opzioni in un gruppo di opzioni. Per informazioni sui gruppi di opzioni, consulta Uso di gruppi di opzioni.

Argomenti

Panoramica dei gruppi di parametri

Un gruppo di parametri database agisce da container per i valori di configurazione del motore che si applicano a una o più istanze database.

I gruppi di parametri cluster database si applicano solo ai cluster database multi-AZ. In un cluster database multi-AZ, le impostazioni del gruppo di parametri cluster database vengono utilizzate per tutte le istanze database nel cluster. Il gruppo di parametri database predefinito per il motore di database e la versione del motore di database viene utilizzato per ogni istanza database nel cluster database.

Argomenti

- Parametri statici e dinamici dell'istanza database
- Parametri statici e dinamici del cluster database
- Parametri del set di caratteri
- Parametri e valori dei parametri supportati

Gruppi di parametri predefiniti e personalizzati

Se decidi di creare un'istanza database senza specificare un gruppo di parametri di database, l'istanza database utilizza un gruppo di parametri predefinito. Allo stesso modo, se crei un cluster di database Multi-AZ senza specificare un gruppo di parametri cluster di database, il cluster di database utilizza un gruppo di parametri cluster di databasedi default. Ogni gruppo di parametri di default contiene le impostazioni predefinite del motore del database e le impostazioni predefinite di sistema di Amazon RDS in base a motore, classe di elaborazione e storage allocato dell'istanza.

Non puoi modificare le impostazioni dei parametri di un gruppo di parametri predefinito. Puoi invece procedere come descritto di seguito:

1. Crea un nuovo set di parametri.
2. Modifica le impostazioni dei parametri desiderati. Non tutti i parametri del motore di database presenti nel gruppo di parametri possono essere modificati.
3. Modifica l'istanza DB o il cluster DB per associare il nuovo gruppo di parametri.

   Quando si associa un nuovo gruppo di parametri DB a un'istanza DB, l'associazione avviene immediatamente. Per ulteriori informazioni sulla modifica di un'istanza database, consulta Modifica di un'istanza database Amazon RDS. Per informazioni sulla modifica di un cluster database multi-AZ, consulta Modifica di un cluster di database Multi-AZ.

   > **Note**
   >
   > Se hai modificato l'istanza database per utilizzare un gruppo di parametri personalizzati e avvii l'istanza database, RDS riavvia automaticamente l'istanza database come parte del processo di avvio.

RDS applica i parametri statici e dinamici modificati in un nuovo gruppo di parametri associato solo dopo il riavvio dell'istanza DB. Tuttavia, se modifichi i parametri dinamici nel gruppo di parametri database associato all'istanza database, tali modifiche vengono applicate immediatamente senza

eseguire il riavvio. Per informazioni sulla modifica del gruppo di parametri database, consulta Modifica di un'istanza database Amazon RDS.

Se aggiorni i parametri all'interno di un gruppo di parametri database, le modifiche si applicano a tutte le istanze database associate al gruppo di parametri. Allo stesso modo, se aggiorni i parametri in un gruppo di parametri cluster database multi-AZ, le modifiche si applicano a tutti i cluster database Aurora associati al gruppo di parametri cluster database.

Se non si desidera creare un gruppo di parametri da zero, è possibile copiare un gruppo di parametri esistente con il AWS CLIcopy-db-parameter-groupcomando o copy-db-cluster-parameter il comando -group. In alcuni casi la copia di un gruppo di parametri è utile. Ad esempio quando devi includere la maggior parte dei valori e dei parametri personalizzati del gruppo di parametri database esistente in un nuovo gruppo di parametri database.

Parametri statici e dinamici dell'istanza database

I parametri di istanza database sono statici o dinamici. Di seguito sono riportate le differenze:

- Quando modifichi un parametro statico e salvi il gruppo parametri del database, la modifica del parametro diventa effettiva al riavvio manuale delle istanze database associate. Per i parametri statici, la console utilizza sempre `pending-reboot` per `ApplyMethod`.
- Quando si modifica un parametro dinamico, per impostazione predefinita la modifica del parametro diventa immediatamente effettiva, senza richiedere il riavvio. Quando si utilizza il AWS Management Console per modificare i valori dei parametri dell'istanza DB, viene sempre utilizzato `immediate` `ApplyMethod` per i parametri dinamici. Per posticipare la modifica dei parametri fino al riavvio di un'istanza DB associata, utilizza l'API AWS CLI o RDS. Quindi, imposta il valore `ApplyMethod` su `pending-reboot` per la modifica del parametro.

> **Note**
>
> L'utilizzo `pending-reboot` con parametri dinamici nell'API AWS CLI o RDS su istanze DB di RDS per SQL Server genera un errore. Utilizza `apply-immediately` su RDS per SQL Server.

Per ulteriori informazioni sull'utilizzo di per modificare il valore AWS CLI di un parametro, vedere. modify-db-parameter-group Per ulteriori informazioni sull'utilizzo dell'API RDS per modificare il valore di un parametro, consulta ParameterGroupModifyDB.

Se l'istanza database non usa le modifiche più recenti apportate al gruppo di parametri database associato, la console mostra il gruppo di parametri database con lo stato pending-reboot. Questo stato non comporta il riavvio automatico durante la successiva finestra di manutenzione. Per applicare le ultime modifiche del parametro su quella istanza database, riavvia manualmente l'istanza database.

Parametri statici e dinamici del cluster database

I parametri di cluster di database sono statici o dinamici. Di seguito sono riportate le differenze:

- Quando modifichi un parametro statico e salvi il gruppo di parametri del cluster di database, la modifica del parametro diventa effettiva al riavvio manuale di ogni istanza database sui cluster di database associati. Per i parametri statici, la console utilizza sempre `pending-reboot` per `ApplyMethod`.
- Quando si modifica un parametro dinamico, per impostazione predefinita la modifica del parametro diventa immediatamente effettiva, senza richiedere il riavvio. Quando si utilizza il AWS Management Console per modificare i valori dei parametri del cluster DB, viene sempre utilizzato `immediate` `ApplyMethod` per i parametri dinamici. Per posticipare la modifica dei parametri fino al riavvio di un cluster DB associato, utilizza l'API AWS CLI o RDS. Quindi, imposta il valore `ApplyMethod` su `pending-reboot` per la modifica del parametro.

Per ulteriori informazioni sull'utilizzo di per modificare il valore AWS CLI di un parametro, consulta - group. modify-db-cluster-parameter Per ulteriori informazioni sull'utilizzo dell'API RDS per modificare il valore di un parametro, consulta ModifyDB. ClusterParameterGroup

Parametri del set di caratteri

Prima di creare l'istanza database o il cluster database multi-AZ imposta tutti i parametri correlati al set di caratteri o alla regola di confronto del database nel gruppo di parametri. prima di creare un database. In questo modo, il database predefinito e i nuovi database usano i valori della regola di confronto e del set di caratteri specificati. Se modifichi parametri di confronto o del set di caratteri, le modifiche dei parametri non vengono applicate a database esistenti.

Per alcuni motori database puoi modificare valori di confronto o del set di caratteri per un database esistente usando il comando `ALTER DATABASE`, ad esempio:

```
ALTER DATABASE database_name CHARACTER SET character_set_name COLLATE collation;
```

Per ulteriori informazioni su come modificare il set di caratteri o i valori di confronto relativi a un database, consulta la documentazione relativa al motore database.

Parametri e valori dei parametri supportati

Per determinare i parametri supportati per il motore del database, vedi i parametri nel gruppo di parametri database e il gruppo di parametri cluster database usato dall'istanza database o dal cluster database. Per ulteriori informazioni, consultare Visualizzazione dei valori dei parametri per un gruppo di parametri del database e Visualizzazione dei valori dei parametri per un gruppo di parametri del cluster database.

In molti casi è possibile specificare valori interi e booleani per i parametri di database utilizzando espressioni, formule e funzioni. Le funzioni possono includere un'espressione logaritmica matematica. Tuttavia, non tutti i parametri supportano espressioni, formule e funzioni per i valori dei parametri. Per ulteriori informazioni, consulta Specifica dei parametri del database.

Un'impostazione errata dei parametri in un gruppo di parametri può avere conseguenze negative impreviste, tra cui il peggioramento delle prestazioni e l'instabilità del sistema. Fai sempre attenzione quando modifichi i parametri database ed esegui il backup dei dati prima di modificare un gruppo di parametri. Prova le modifiche delle impostazioni del gruppo di parametri in un'istanza database o un cluster database di test prima di applicare le modifiche a un'istanza database o un cluster database di produzione.

Utilizzo di gruppi di parametri DB in un'istanza DB

Le istanze database utilizzano gruppi di parametri database. Le sezioni seguenti descrivono la configurazione e la gestione dei gruppi di parametri dell'istanza database.

Argomenti

- Creazione di un gruppo di parametri del database
- Associazione di un gruppo di parametri database a un'istanza database
- Modifica di parametri in un gruppo di parametri del database
- Reimpostazione dei parametri in un gruppo di parametri database sui valori predefiniti
- Copia di un gruppo di parametri database
- Generazione di un elenco di gruppi di parametri del database
- Visualizzazione dei valori dei parametri per un gruppo di parametri del database
- Eliminazione di un gruppo di parametri DB

Creazione di un gruppo di parametri del database

È possibile creare un nuovo gruppo di parametri DB utilizzando AWS Management Console AWS CLI, the o l'API RDS.

Le seguenti limitazioni si applicano al nome del gruppo di parametri database:

- Il nome deve contenere da 1 a 255 lettere, numeri o trattini.

  I nomi del gruppo di parametri predefiniti possono includere un punto, ad esempio `default.mysql8.0`. Tuttavia, i nomi del gruppo di parametri personalizzati non possono includere un punto.
- Il primo carattere deve essere una lettera.
- Il nome non può terminare con un trattino o contenere due trattini consecutivi.

Console

Per creare un gruppo di parametri del database

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel pannello di navigazione, scegliParameter groups (Gruppi di parametri).
3. Scegli Create parameter group (Crea gruppo di parametri).
4. Per il nome del gruppo di parametri, inserisci il nome del nuovo gruppo di parametri DB.
5. In Descrizione, inserisci una descrizione per il tuo nuovo gruppo di parametri DB.
6. Per Tipo di motore, scegli il tuo motore DB.
7. Per Famiglia di gruppi di parametri, scegli una famiglia di gruppi di parametri DB.
8. Per Tipo, se applicabile, scegliete DB Parameter Group.
9. Scegli Create (Crea).

AWS CLI

Per creare un gruppo di parametri DB, utilizzate il AWS CLI `create-db-parameter-group`comando. L'esempio seguente crea un gruppo di parametri database denominato mydbparametergroup per MySQL versione 8.0 con la descrizione "My new parameter group (Il mio nuovo gruppo di parametri)".

Includi i parametri obbligatori seguenti:

- `--db-parameter-group-name`
- `--db-parameter-group-family`
- `--description`

Per elencare tutte le famiglie del gruppo di parametri disponibili, usa il comando seguente:

```
aws rds describe-db-engine-versions --query "DBEngineVersions[].DBParameterGroupFamily"
```

> **Note**
>
> L'output contiene duplicati.

Example

Per LinuxmacOS, oUnix:

```
aws rds create-db-parameter-group \
    --db-parameter-group-name mydbparametergroup \
    --db-parameter-group-family MySQL8.0 \
    --description "My new parameter group"
```

Per Windows:

```
aws rds create-db-parameter-group ^
    --db-parameter-group-name mydbparametergroup ^
    --db-parameter-group-family MySQL8.0 ^
    --description "My new parameter group"
```

Questo comando genera un output simile al seguente:

```
DBPARAMETERGROUP  mydbparametergroup  mysql8.0  My new parameter group
```

API RDS

Per creare un gruppo di parametri database, utilizzare l'operazione API RDS `CreateDBParameterGroup`.

Includi i parametri obbligatori seguenti:

- `DBParameterGroupName`
- `DBParameterGroupFamily`
- `Description`

Associazione di un gruppo di parametri database a un'istanza database

Puoi creare i tuoi gruppi di parametri database con impostazioni personalizzate. È possibile associare un gruppo di parametri DB a un'istanza DB utilizzando l' AWS Management Console API AWS CLI, the o RDS. Ciò è possibile quando crei o modifichi un'istanza database.

Per ulteriori informazioni sulla creazione di un gruppo di parametri del database, consulta Creazione di un gruppo di parametri del database. Per informazioni sulla creazione di un'istanza database, consulta Creazione di un'istanza database Amazon RDS. Per ulteriori informazioni sulla modifica di un'istanza database, consulta Modifica di un'istanza database Amazon RDS.

> **Note**
>
> Quando si associa un nuovo gruppo parametri del database a un'istanza database, i parametri statici e dinamici modificati vengono applicati solo dopo il riavvio dell'istanza database. Tuttavia, se modifichi i parametri dinamici nel gruppo di parametri database associato all'istanza database, tali modifiche vengono applicate immediatamente senza eseguire il riavvio.

Console

Per associare un gruppo di parametri del database a un'istanza database

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, scegliere Databases (Database) e selezionare l'istanza database da modificare.
3. Scegliere Modify (Modifica). Viene visualizzata la pagina Modify DB Instance (Modifica istanza database).
4. Modifica l'impostazione del gruppo di parametri database .

5. Scegliere Continue (Continua) e controllare il riepilogo delle modifiche.
6. (Facoltativo) Scegliere Applica immediatamente per applicare immediatamente le modifiche. In alcuni casi, la chiusura di questa opzione può causare un'interruzione. Per ulteriori informazioni, consulta Impostazione delle modifiche alla pianificazione.
7. Nella pagina di conferma esaminare le modifiche. Se sono corrette, seleziona Modifica istanza database per salvare le modifiche.

   Oppure scegliere Back (Indietro) per cambiare le modifiche o Cancel (Annulla) per annullare le modifiche.

AWS CLI

Per associare un gruppo di parametri DB a un'istanza DB, usa il AWS CLI `modify-db-instance`comando con le seguenti opzioni:

- `--db-instance-identifier`
- `--db-parameter-group-name`

Nell'esempio seguente il gruppo di parametri database `mydbpg` viene associato all'istanza database `database-1`. Le modifiche vengono applicate immediatamente tramite `--apply-immediately`. Utilizza `--no-apply-immediately` per applicare le modifiche durante la successiva finestra di manutenzione. Per ulteriori informazioni, consulta Impostazione delle modifiche alla pianificazione.

Example

Per LinuxmacOS, oUnix:

```
aws rds modify-db-instance \
    --db-instance-identifier database-1 \
    --db-parameter-group-name mydbpg \
    --apply-immediately
```

Per Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier database-1 ^
    --db-parameter-group-name mydbpg ^
    --apply-immediately
```

API RDS

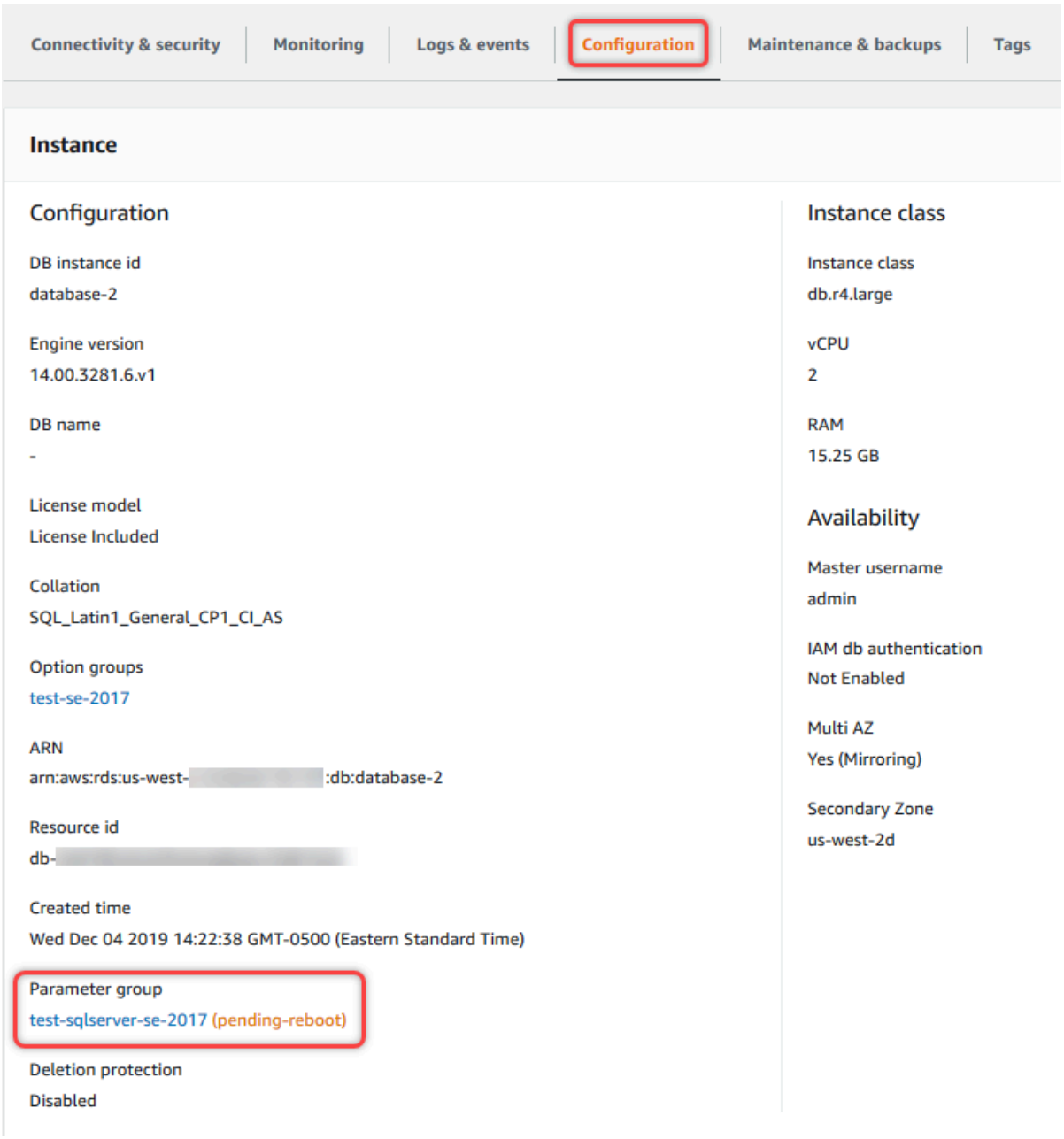
Per associare un gruppo parametri del database a un'istanza database, utilizza l'operazione `ModifyDBInstance` dell'API RDS con i seguenti parametri:

- `DBInstanceName`
- `DBParameterGroupName`

Modifica di parametri in un gruppo di parametri del database

Puoi modificare i valori dei parametri in un gruppo di parametri database creato dal cliente, ma non puoi modificare i valori dei parametri in un gruppo di parametri database predefinito. Le modifiche ai parametri in un gruppo di parametri database creato dal cliente vengono applicate a tutte le istanze database associate al gruppo di parametri database.

Le modifiche apportate ad alcuni parametri vengono applicate all'istanza database immediatamente senza un riavvio. Le modifiche apportate ad altri parametri vengono applicate solo dopo che l'istanza database viene riavviata. La console RDS mostra lo stato del gruppo di parametri database associato a un'istanza database nella scheda Configuration (Configurazione). Supponi, ad esempio che l'istanza database non utilizzi le modifiche più recenti apportate al gruppo di parametri database associato. In questo caso, la console RDS mostra il gruppo di parametri database con lo stato pending-reboot. Per applicare le ultime modifiche del parametro su quella istanza database, riavvia manualmente l’istanza database.



Console

Per modificare i parametri in un gruppo di parametri DB

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione scegliere Parameter groups (Gruppi di parametri).
3. Nell'elenco, scegli il nome del gruppo di parametri che desideri modificare.
4. Per Parameter group actions (Operazioni del gruppo di parametri), scegliere Edit (Modifica).

5. Modificare i valori dei parametri desiderati. È possibile scorrere i parametri usando i tasti freccia in alto a destra nella finestra di dialogo.

   Non è possibile modificare valori nel gruppo di parametri predefinito.

6. Scegli Save changes (Salva modifiche).

AWS CLI

Per modificare un gruppo di parametri DB, utilizzate il AWS CLI `modify-db-parameter-group`comando con le seguenti opzioni richieste:

- `--db-parameter-group-name`
- `--parameters`

L'esempio seguente modifica i valori `max_connections` e `max_allowed_packet` nel gruppo di parametri database denominato mydbparametergroup.

Example

Per LinuxmacOS, oUnix:

```
aws rds modify-db-parameter-group \
    --db-parameter-group-name mydbparametergroup \
    --parameters
 "ParameterName=max_connections,ParameterValue=250,ApplyMethod=immediate" \

 "ParameterName=max_allowed_packet,ParameterValue=1024,ApplyMethod=immediate"
```

Per Windows:

```
aws rds modify-db-parameter-group ^
    --db-parameter-group-name mydbparametergroup ^
    --parameters
 "ParameterName=max_connections,ParameterValue=250,ApplyMethod=immediate" ^

 "ParameterName=max_allowed_packet,ParameterValue=1024,ApplyMethod=immediate"
```

Il comando genera un output simile al seguente:

```
DBPARAMETERGROUP  mydbparametergroup
```

API RDS

Per modificare un gruppo parametri del database, utilizza l'operazione `ModifyDBParameterGroup` dell'API RDS con i seguenti parametri obbligatori:

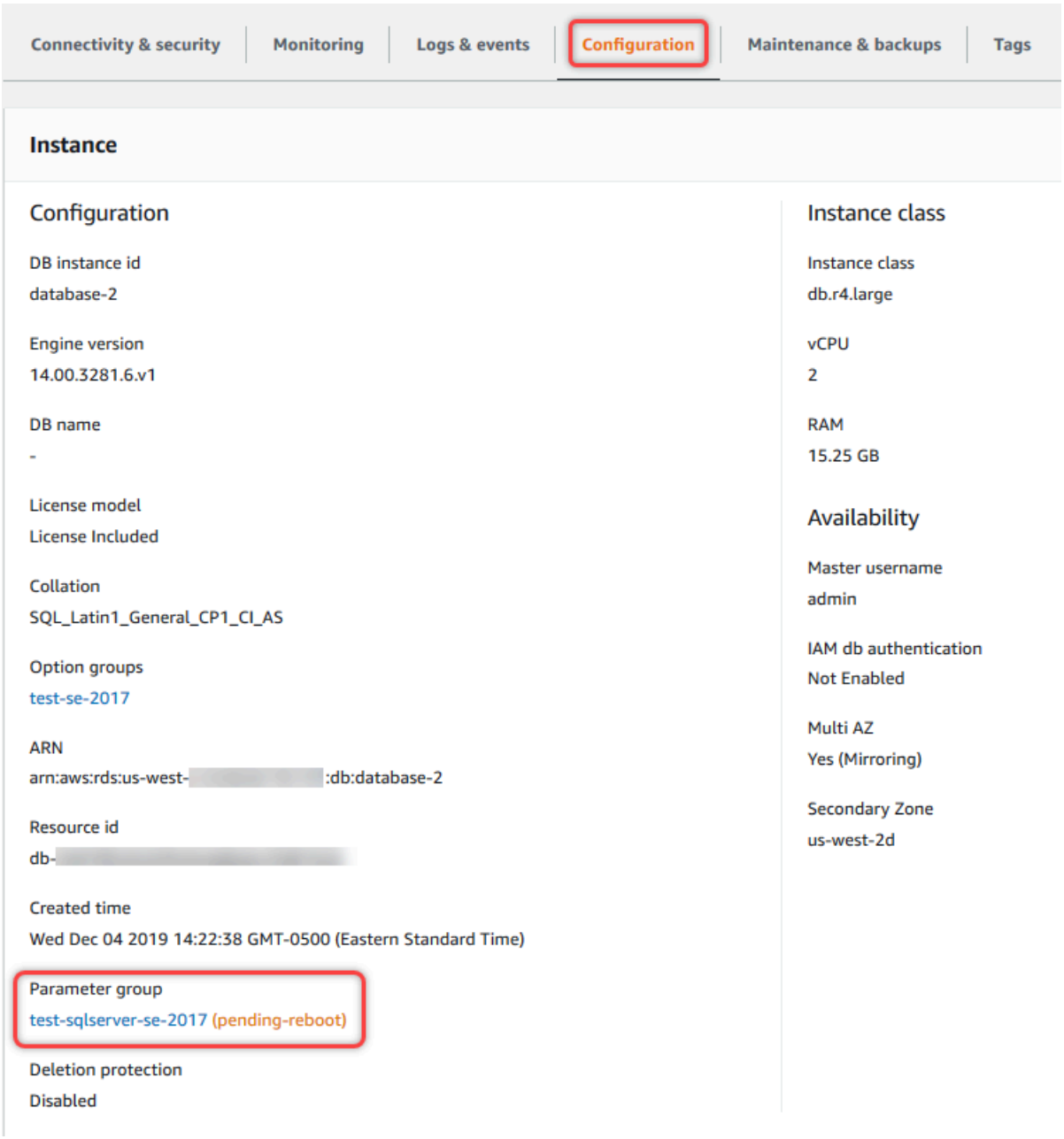
- `DBParameterGroupName`
- `Parameters`

Reimpostazione dei parametri in un gruppo di parametri database sui valori predefiniti

Puoi reimpostare i valori dei parametri in un gruppo di parametri database creato dal cliente sui valori predefiniti. Le modifiche ai parametri in un gruppo di parametri database creato dal cliente vengono applicate a tutte le istanze database associate al gruppo di parametri database.

Quando utilizzi la console, puoi reimpostare specifici parametri sui valori predefiniti. Tuttavia, non è possibile reimpostare altrettanto facilmente tutti i parametri nel gruppo di parametri database contemporaneamente. Quando utilizzi l'API AWS CLI o RDS, puoi ripristinare i valori predefiniti di parametri specifici. Puoi anche reimpostare tutti i parametri del gruppo di parametri database contemporaneamente.

Le modifiche apportate ad alcuni parametri vengono applicate all'istanza database immediatamente senza un riavvio. Le modifiche apportate ad altri parametri vengono applicate solo dopo che l'istanza database viene riavviata. La console RDS mostra lo stato del gruppo di parametri database associato a un'istanza database nella scheda Configuration (Configurazione). Supponi, ad esempio che l'istanza database non utilizzi le modifiche più recenti apportate al gruppo di parametri database associato. In questo caso, la console RDS mostra il gruppo di parametri database con lo stato pending-reboot. Per applicare le ultime modifiche del parametro su quella istanza database, riavvia manualmente l’istanza database.



> **Note**
>
> In un gruppo di parametri database predefinito, i parametri vengono sempre impostati sui valori di default.

Console

Per ripristinare i valori predefiniti dei parametri di un gruppo di parametri database

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione scegliere Parameter groups (Gruppi di parametri).
3. Nell'elenco, scegliere il gruppo di parametri.
4. Per Parameter group actions (Operazioni del gruppo di parametri), scegliere Edit (Modifica).
5. Scegliere i parametri per i quali si desidera ripristinare i valori predefiniti. È possibile scorrere i parametri usando i tasti freccia in alto a destra nella finestra di dialogo.

   Non è possibile reimpostare valori in un gruppo di parametri predefinito.
6. Scegli Reimposta , quindi conferma selezionando Ripristina parametri.

AWS CLI

Per reimpostare alcuni o tutti i parametri in un gruppo di parametri DB, usa il AWS CLI `reset-db-parameter-group`comando con la seguente opzione obbligatoria:`--db-parameter-group-name`.

Per reimpostare tutti i parametri nel gruppo di parametri database, specifica l'opzione `--reset-all-parameters`. Per reimpostare parametri specifici, specifica l'opzione `--parameters`.

Nell'esempio seguente tutti i parametri del gruppo di parametri del database denominato mydbparametergroup vengono reimpostati sui valori predefiniti.

Example

Per LinuxmacOS, oUnix:

```
aws rds reset-db-parameter-group \
    --db-parameter-group-name mydbparametergroup \
    --reset-all-parameters
```

Per Windows:

```
aws rds reset-db-parameter-group ^
    --db-parameter-group-name mydbparametergroup ^
```

```
    --reset-all-parameters
```

Nel seguente esempio le opzioni `max_connections` e `max_allowed_packet` vengono reimpostate sui loro valori predefiniti nel gruppo di parametri database denominato mydbparametergroup.

Example

Per LinuxmacOS, oUnix:

```
aws rds reset-db-parameter-group \
    --db-parameter-group-name mydbparametergroup \
    --parameters "ParameterName=max_connections,ApplyMethod=immediate" \
                 "ParameterName=max_allowed_packet,ApplyMethod=immediate"
```

Per Windows:

```
aws rds reset-db-parameter-group ^
    --db-parameter-group-name mydbparametergroup ^
    --parameters "ParameterName=max_connections,ApplyMethod=immediate" ^
                 "ParameterName=max_allowed_packet,ApplyMethod=immediate"
```

Il comando genera un output simile al seguente:

```
DBParameterGroupName  mydbparametergroup
```

API RDS

Per ripristinare i valori predefiniti dei parametri di un gruppo di parametri database, utilizza il comando `ResetDBParameterGroup` dell'API RDS con il seguente parametro obbligatorio: `DBParameterGroupName`.

Per reimpostare tutti i parametri nel gruppo di parametri database, imposta il parametro `ResetAllParameters` su `true`. Per reimpostare parametri specifici, specifica il parametro `Parameters`.

Copia di un gruppo di parametri database

Puoi copiare i gruppi di parametri database personalizzati che hai creato. La copia di un gruppo di parametri può essere una soluzione utile. Ad esempio quando crei un gruppo di parametri database

e vuoi includere la maggior parte dei parametri e dei valori personalizzati in un nuovo gruppo di parametri del database. È possibile copiare un gruppo di parametri DB utilizzando AWS Management Console. È inoltre possibile utilizzare il AWS CLI copy-db-parameter-groupcomando o l'operazione RDS API CopyDB ParameterGroup.

Dopo aver copiato un gruppo di parametri database, attendi almeno 5 minuti prima di creare la prima istanza database che usa il gruppo di parametri database come predefinito. In questo modo, Amazon RDS può completare l'operazione di copia prima che venga usato il gruppo di parametri. Questo è particolarmente importante per parametri critici durante la creazione del database predefinito per un'istanza database. Un esempio è il set di caratteri per il database predefinito definito dal parametro `character_set_database`. Utilizza l'opzione Parameter Groups della console Amazon RDS o il describe-db-parameterscomando per verificare che il gruppo di parametri DB sia stato creato.

> **Note**
>
> Non puoi copiare un gruppo di parametri predefinito. Tuttavia, puoi creare un nuovo gruppo di parametri basato su un gruppo di parametri predefinito.
> Non puoi copiare un gruppo di parametri DB in un altro Account AWS o Regione AWS.

Console

Per copiare un gruppo di parametri del database

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione scegliere Parameter groups (Gruppi di parametri).
3. Nell'elenco, scegliere il gruppo di parametri personalizzato da copiare.
4. Per Parameter group actions (Operazioni del gruppo di parametri), scegliere Copy (Copia).
5. In New DB parameter group identifier (Identificatore nuovo gruppo di parametri database) immettere un nome per il nuovo gruppo di parametri.
6. Nella casella Description (Descrizione), immettere una descrizione per il nuovo gruppo di parametri.
7. Scegliere Copy (Copia).

AWS CLI

Per copiare un gruppo di parametri DB, usa il AWS CLI `copy-db-parameter-group`comando con le seguenti opzioni richieste:

- `--source-db-parameter-group-identifier`
- `--target-db-parameter-group-identifier`
- `--target-db-parameter-group-description`

L'esempio seguente crea un nuovo gruppo di parametri database denominato `mygroup2`, che è una copia del gruppo di parametri database `mygroup1`.

Example

Per LinuxmacOS, oUnix:

```
aws rds copy-db-parameter-group \
    --source-db-parameter-group-identifier mygroup1 \
    --target-db-parameter-group-identifier mygroup2 \
    --target-db-parameter-group-description "DB parameter group 2"
```

Per Windows:

```
aws rds copy-db-parameter-group ^
    --source-db-parameter-group-identifier mygroup1 ^
    --target-db-parameter-group-identifier mygroup2 ^
    --target-db-parameter-group-description "DB parameter group 2"
```

API RDS

Per copiare un gruppo di parametri del database, utilizza l'operazione API RDS `CopyDBParameterGroup`:

- `SourceDBParameterGroupIdentifier`
- `TargetDBParameterGroupIdentifier`
- `TargetDBParameterGroupDescription`

Generazione di un elenco di gruppi di parametri del database

Puoi elencare i gruppi di parametri DB che hai creato per il tuo AWS account.

> **Note**
>
> I gruppi di parametri predefiniti vengono creati automaticamente da un modello di parametro predefinito quando crei un'istanza database per un motore e una versione di database specifici. Questi gruppi di parametri predefiniti contengono le impostazioni dei parametri preferite e non possono essere modificati. Quando crei un gruppo di parametri personalizzato, puoi modificare le impostazioni dei parametri.

Console

Per elencare tutti i gruppi di parametri DB per un AWS account

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione scegliere Parameter groups (Gruppi di parametri).

   I gruppi di parametri database vengono visualizzati in un elenco.

AWS CLI

Per elencare tutti i gruppi di parametri DB per un AWS account, usa il AWS CLI `describe-db-parameter-groups`comando.

Example

L'esempio seguente elenca tutti i gruppi di parametri database disponibili per un account AWS .

```
aws rds describe-db-parameter-groups
```

Questo comando restituisce una risposta simile alla seguente:

```
DBPARAMETERGROUP  default.mysql8.0     mysql8.0  Default parameter group for MySQL8.0
DBPARAMETERGROUP  mydbparametergroup   mysql8.0  My new parameter group
```

L'esempio seguente descrive il gruppo di parametri mydbparamgroup1.

Per LinuxmacOS, oUnix:

```
aws rds describe-db-parameter-groups \
    --db-parameter-group-name mydbparamgroup1
```

Per Windows:

```
aws rds describe-db-parameter-groups ^
    --db-parameter-group-name mydbparamgroup1
```

Questo comando restituisce una risposta simile alla seguente:

```
DBPARAMETERGROUP  mydbparametergroup1  mysql8.0  My new parameter group
```

API RDS

Per elencare tutti i gruppi di parametri DB per un AWS account, utilizza l'`DescribeDBParameterGroups`operazione API RDS.

Visualizzazione dei valori dei parametri per un gruppo di parametri del database

Puoi ottenere un elenco di tutti i parametri in un gruppo di parametri del database e dei rispettivi valori.

Console

Per visualizzare i valori dei parametri per un gruppo di parametri del database

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione scegliere Parameter groups (Gruppi di parametri).

   I gruppi di parametri database vengono visualizzati in un elenco.
3. Scegliere il nome del gruppo di parametri per visualizzarne l'elenco di parametri.

AWS CLI

Per visualizzare i valori dei parametri per un gruppo di parametri DB, usa il AWS CLI `describe-db-parameters`comando con il seguente parametro obbligatorio.

- `--db-parameter-group-name`

Example

L'esempio seguente elenca i parametri e i valori dei parametri per un gruppo di parametri database denominato mydbparametergroup.

```
aws rds describe-db-parameters --db-parameter-group-name mydbparametergroup
```

Questo comando restituisce una risposta simile alla seguente:

```
DBPARAMETER  Parameter Name            Parameter Value  Source           Data Type
 Apply Type  Is Modifiable
DBPARAMETER  allow-suspicious-udfs                      engine-default   boolean
 static      false
DBPARAMETER  auto_increment_increment                   engine-default   integer
 dynamic     true
DBPARAMETER  auto_increment_offset                      engine-default   integer
 dynamic     true
DBPARAMETER  binlog_cache_size         32768            system           integer
 dynamic     true
DBPARAMETER  socket                    /tmp/mysql.sock  system           string
 static      false
```

API RDS

Per visualizzare i valori dei parametri per un gruppo di parametri del database, utilizza il comando RDS API `DescribeDBParameters` con il seguente parametro obbligatorio.

- `DBParameterGroupName`

Eliminazione di un gruppo di parametri DB

È possibile eliminare un gruppo di parametri DB utilizzando AWS Management Console AWS CLI, o l'API RDS. Un gruppo di parametri è idoneo per l'eliminazione solo se non è associato a un'istanza DB.

Console

Per eliminare un gruppo di parametri DB

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione scegliere Parameter groups (Gruppi di parametri).

   I gruppi di parametri database vengono visualizzati in un elenco.
3. Scegli il nome dei gruppi di parametri da eliminare.
4. Scegli Azioni e poi Elimina.
5. Controllate i nomi dei gruppi di parametri, quindi scegliete Elimina.

AWS CLI

Per eliminare un gruppo di parametri DB, utilizzate il AWS CLI `delete-db-parameter-group`comando con il seguente parametro richiesto.

- `--db-parameter-group-name`

Example

L'esempio seguente elimina un gruppo di parametri DB denominato mydbparametergroup.

```
aws rds delete-db-parameter-group --db-parameter-group-name mydbparametergroup
```

API RDS

Per eliminare un gruppo di parametri DB, utilizzate il `DeleteDBParameterGroup`comando API RDS con il seguente parametro obbligatorio.

- `DBParameterGroupName`

Utilizzo di gruppi di parametri cluster di database per cluster database Multi-AZ

I cluster database Multi-AZ utilizzano gruppi di parametri di cluster di database. Le sezioni seguenti descrivono la configurazione e la gestione dei gruppi di parametri del cluster di database.

Argomenti

Creazione di un gruppo di parametri del cluster database

È possibile creare un nuovo gruppo di parametri del cluster DB utilizzando l'API AWS Management Console AWS CLI, the o RDS.

Dopo aver creato un gruppo di parametri del cluster database, devi attendere almeno 5 minuti prima di creare un cluster database che utilizza tale gruppo. In questo modo, Amazon RDS può completare l'operazione di creazione del gruppo di parametri prima che tale gruppo venga usato dal nuovo cluster database. Puoi utilizzare la pagina Gruppi di parametri nella console Amazon RDS o il describe-db-cluster-parameterscomando per verificare che il gruppo di parametri del cluster DB sia stato creato.

Le seguenti limitazioni si applicano al nome del gruppo di parametri del·cluster database:

- Il nome deve contenere da 1 a 255 lettere, numeri o trattini.

  I nomi del gruppo di parametri predefiniti possono includere un punto, ad esempio `default.aurora-mysql5.7`. Tuttavia, i nomi del gruppo di parametri personalizzati non possono includere un punto.
- Il primo carattere deve essere una lettera.
- Il nome non può terminare con un trattino o contenere due trattini consecutivi.

Console

Per creare un gruppo di parametri del cluster database

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.

2. Nel pannello di navigazione, scegliParameter groups (Gruppi di parametri).
3. Scegli Create parameter group (Crea gruppo di parametri).

   Viene visualizzata la pagina Create parameter group (Crea gruppo di parametri).
4. Nell'elenco Parameter group family (Famiglia gruppo di parametri) selezionare una famiglia del gruppo di parametri database.
5. Nell'elenco Tipo, seleziona il gruppo di parametri del cluster DB.
6. Nella casella Group name (Nome gruppo) immettere il nome del nuovo gruppo di parametri cluster database.
7. Nella casella Description (Descrizione) inserire una descrizione per il nuovo gruppo di parametri cluster database.
8. Scegli Create (Crea).

AWS CLI

Per creare un gruppo di parametri del cluster DB, utilizzare il AWS CLI `create-db-cluster-parameter-group`comando.

L'esempio seguente crea un gruppo di parametri cluster di database denominato mydbclusterparametergroup per RDS for MySQL versione 8.0 con la descrizione "My new cluster parameter group (Il mio nuovo gruppo di parametri cluster)".

Includi i parametri obbligatori seguenti:

- `--db-cluster-parameter-group-name`
- `--db-parameter-group-family`
- `--description`

Per elencare tutte le famiglie del gruppo di parametri disponibili, usa il comando seguente:

```
aws rds describe-db-engine-versions --query "DBEngineVersions[].DBParameterGroupFamily"
```

> **Note**
>
> L'output contiene duplicati.

Example

Per LinuxmacOS, oUnix:

```
aws rds create-db-cluster-parameter-group \
    --db-cluster-parameter-group-name mydbclusterparametergroup \
    --db-parameter-group-family mysql8.0 \
    --description "My new cluster parameter group"
```

Per Windows:

```
aws rds create-db-cluster-parameter-group ^
    --db-cluster-parameter-group-name mydbclusterparametergroup ^
    --db-parameter-group-family mysql8.0 ^
    --description "My new cluster parameter group"
```

Questo comando genera un output simile al seguente:

```
{
    "DBClusterParameterGroup": {
        "DBClusterParameterGroupName": "mydbclusterparametergroup",
        "DBParameterGroupFamily": "mysql8.0",
        "Description": "My new cluster parameter group",
        "DBClusterParameterGroupArn": "arn:aws:rds:us-east-1:123456789012:cluster-
pg:mydbclusterparametergroup2"
    }
}
```

API RDS

Per creare un gruppo di parametri del cluster di database, utilizza l'operazione `CreateDBClusterParameterGroup` dell'API RDS.

Includi i parametri obbligatori seguenti:

- `DBClusterParameterGroupName`
- `DBParameterGroupFamily`
- `Description`

Modifica di parametri in un gruppo di parametri del cluster database

Non puoi modificare i valori di parametri in un gruppo di parametri cluster database creato dal cliente. Non puoi modificare i valori dei parametri in un gruppo di parametri cluster database predefinito. Le modifiche ai parametri in un gruppo di parametri cluster database creato dal cliente vengono applicate a tutti i cluster database associati al gruppo di parametri del cluster database.

Console

Per modificare un gruppo di parametri del cluster database

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione scegliere Parameter groups (Gruppi di parametri).
3. Nell'elenco, selezionare il gruppo di parametri da modificare.
4. Per Parameter group actions (Operazioni del gruppo di parametri), scegliere Edit (Modifica).
5. Modificare i valori dei parametri desiderati. È possibile scorrere i parametri usando i tasti freccia in alto a destra nella finestra di dialogo.

   Non è possibile modificare valori nel gruppo di parametri predefinito.
6. Scegli Save changes (Salva modifiche).
7. Riavvia l'istanza DB principale (writer) nel cluster per applicarvi le modifiche.
8. Quindi riavvia le istanze DB del lettore per applicare le modifiche.

AWS CLI

Per modificare un gruppo di parametri del cluster DB, utilizzate il AWS CLI `modify-db-cluster-parameter-group`comando con i seguenti parametri obbligatori:

- `--db-cluster-parameter-group-name`
- `--parameters`

L'esempio seguente modifica i valori `server_audit_logging` e `server_audit_logs_upload` nel gruppo di parametri cluster database denominato mydbclusterparametergroup.

Example

Per LinuxmacOS, oUnix:

```
aws rds modify-db-cluster-parameter-group \
    --db-cluster-parameter-group-name mydbclusterparametergroup \
    --parameters
 "ParameterName=server_audit_logging,ParameterValue=1,ApplyMethod=immediate" \

 "ParameterName=server_audit_logs_upload,ParameterValue=1,ApplyMethod=immediate"
```

Per Windows:

```
aws rds modify-db-cluster-parameter-group ^
    --db-cluster-parameter-group-name mydbclusterparametergroup ^
    --parameters
 "ParameterName=server_audit_logging,ParameterValue=1,ApplyMethod=immediate" ^

 "ParameterName=server_audit_logs_upload,ParameterValue=1,ApplyMethod=immediate"
```

Il comando genera un output simile al seguente:

```
DBCLUSTERPARAMETERGROUP  mydbclusterparametergroup
```

API RDS

Per modificare un gruppo di parametri cluster database, utilizza l'operazione `ModifyDBClusterParameterGroup` dell'API RDS con i parametri obbligatori seguenti:

- `DBClusterParameterGroupName`
- `Parameters`

Reimpostazione di parametri in un gruppo di parametri cluster database

È possibile reimpostare i parametri ai valori predefiniti in un gruppo di parametri del cluster di database creato dal cliente. Le modifiche ai parametri in un gruppo di parametri cluster database creato dal cliente vengono applicate a tutti i cluster database associati al gruppo di parametri cluster database.

> **Note**
>
> In un gruppo di parametri cluster di database predefinito, i parametri vengono sempre impostati sui valori di default.

Console

Per ripristinare i valori predefiniti dei parametri di un gruppo di parametri cluster database

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione scegliere Parameter groups (Gruppi di parametri).
3. Nell'elenco, scegliere il gruppo di parametri.
4. Per Parameter group actions (Operazioni del gruppo di parametri), scegliere Edit (Modifica).
5. Scegliere i parametri per i quali si desidera ripristinare i valori predefiniti. È possibile scorrere i parametri usando i tasti freccia in alto a destra nella finestra di dialogo.

   Non è possibile reimpostare valori in un gruppo di parametri predefinito.
6. Scegli Reimposta , quindi conferma selezionando Ripristina parametri.
7. Riavviare l'istanza database primaria nel cluster di database per applicare le modifiche a tutte le istanze database del cluster di database.

AWS CLI

Per ripristinare i parametri di un gruppo di parametri del cluster DB ai valori predefiniti, usa il AWS CLI `reset-db-cluster-parameter-group`comando con la seguente opzione obbligatoria:`--db-cluster-parameter-group-name`.

Per reimpostare tutti i parametri nel gruppo di parametri del cluster di database, specificare l'opzione `--reset-all-parameters`. Per reimpostare parametri specifici, specifica l'opzione `--parameters`.

Nell'esempio seguente tutti i parametri del gruppo di parametri del database denominato mydbparametergroup vengono reimpostati sui valori predefiniti.

Example

Per LinuxmacOS, oUnix:

```
aws rds reset-db-cluster-parameter-group \
    --db-cluster-parameter-group-name mydbparametergroup \
    --reset-all-parameters
```

Per Windows:

```
aws rds reset-db-cluster-parameter-group ^
    --db-cluster-parameter-group-name mydbparametergroup ^
    --reset-all-parameters
```

L'esempio seguente reimposta i valori `server_audit_logging` e `server_audit_logs_upload` nel gruppo di parametri cluster database denominato mydbclusterparametergroup.

Example

Per LinuxmacOS, oUnix:

```
aws rds reset-db-cluster-parameter-group \
    --db-cluster-parameter-group-name mydbclusterparametergroup \
    --parameters "ParameterName=server_audit_logging,ApplyMethod=immediate" \
                 "ParameterName=server_audit_logs_upload,ApplyMethod=immediate"
```

Per Windows:

```
aws rds reset-db-cluster-parameter-group ^
    --db-cluster-parameter-group-name mydbclusterparametergroup ^
    --parameters
 "ParameterName=server_audit_logging,ParameterValue=1,ApplyMethod=immediate" ^

 "ParameterName=server_audit_logs_upload,ParameterValue=1,ApplyMethod=immediate"
```

Il comando genera un output simile al seguente:

```
DBClusterParameterGroupName  mydbclusterparametergroup
```

API RDS

Per ripristinare i valori predefiniti dei parametri di un gruppo di parametri del cluster di database, utilizzare il comando RDS API `ResetDBClusterParameterGroup` con il seguente parametro obbligatorio: `DBClusterParameterGroupName`.

Per reimpostare tutti i parametri nel gruppo di parametri del cluster di database, impostare il parametro `ResetAllParameters` su `true`. Per reimpostare parametri specifici, specifica il parametro `Parameters`.

Copia di un gruppo di parametri cluster database

Puoi copiare i gruppi di parametri cluster database personalizzati che hai creato. La copia di un gruppo di parametri è una soluzione pratica quando hai già creato un gruppo di parametri cluster database e vuoi includere la maggior parte dei parametri e valori personalizzati dal gruppo in un nuovo gruppo di parametri cluster database. È possibile copiare un gruppo di parametri del cluster DB utilizzando il comando AWS CLI copy-db-cluster-parameter-group o l'operazione ClusterParameterGroupCopyDB dell'API RDS.

Dopo aver copiato un gruppo di parametri del cluster database, devi attendere almeno 5 minuti prima di creare un cluster database che utilizza tale gruppo. In questo modo, Amazon RDS può completare l'operazione di copia del gruppo di parametri prima che tale gruppo venga usato dal nuovo cluster database. Puoi utilizzare la pagina Gruppi di parametri nella console Amazon RDS o il describe-db-cluster-parameterscomando per verificare che il gruppo di parametri del cluster DB sia stato creato.

> **Note**
>
> Non puoi copiare un gruppo di parametri predefinito. Tuttavia, puoi creare un nuovo gruppo di parametri basato su un gruppo di parametri predefinito.
> Non puoi copiare un gruppo di parametri del cluster DB in un altro Account AWS o Regione AWS.

Console

Per copiare un gruppo di parametri cluster database

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione scegliere Parameter groups (Gruppi di parametri).
3. Nell'elenco, scegliere il gruppo di parametri personalizzato da copiare.
4. Per Parameter group actions (Operazioni del gruppo di parametri), scegliere Copy (Copia).
5. In New DB parameter group identifier (Identificatore nuovo gruppo di parametri database) immettere un nome per il nuovo gruppo di parametri.
6. Nella casella Description (Descrizione), immettere una descrizione per il nuovo gruppo di parametri.
7. Scegliere Copy (Copia).

AWS CLI

Per copiare un gruppo di parametri del cluster DB, usa il AWS CLI [copy-db-cluster-parameter-group](#)comando con i seguenti parametri obbligatori:

- `--source-db-cluster-parameter-group-identifier`
- `--target-db-cluster-parameter-group-identifier`
- `--target-db-cluster-parameter-group-description`

L'esempio seguente crea un nuovo gruppo di parametri cluster database denominato `mygroup2`, che è una copia del gruppo di parametri cluster database `mygroup1`.

Example

Per LinuxmacOS, oUnix:

```
aws rds copy-db-cluster-parameter-group \
    --source-db-cluster-parameter-group-identifier mygroup1 \
    --target-db-cluster-parameter-group-identifier mygroup2 \
    --target-db-cluster-parameter-group-description "DB parameter group 2"
```

Per Windows:

```
aws rds copy-db-cluster-parameter-group ^
    --source-db-cluster-parameter-group-identifier mygroup1 ^
    --target-db-cluster-parameter-group-identifier mygroup2 ^
    --target-db-cluster-parameter-group-description "DB parameter group 2"
```

API RDS

Per copiare un gruppo di parametri cluster database, utilizza l'operazione API RDS [CopyDBClusterParameterGroup](#) con i parametri obbligatori seguenti:

- `SourceDBClusterParameterGroupIdentifier`
- `TargetDBClusterParameterGroupIdentifier`
- `TargetDBClusterParameterGroupDescription`

Generazione di un elenco di gruppi di parametri del cluster database

Puoi elencare i gruppi di parametri del cluster DB che hai creato per il tuo AWS account.

> **Note**
>
> I gruppi di parametri predefiniti vengono creati automaticamente da un modello di parametro predefinito quando crei un cluster database per un motore e una versione di database specifici. Questi gruppi di parametri predefiniti contengono le impostazioni dei parametri preferite e non possono essere modificati. Quando crei un gruppo di parametri personalizzato, puoi modificare le impostazioni dei parametri.

Console

Per elencare tutti i gruppi di parametri del cluster DB per un AWS account

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione scegliere Parameter groups (Gruppi di parametri).

   Il gruppo di parametri cluster database viene visualizzato nell'elenco con Type (Tipo) impostato su DB cluster parameter group (Gruppo di parametri cluster database).

AWS CLI

Per elencare tutti i gruppi di parametri del cluster DB per un AWS account, usa il AWS CLI `describe-db-cluster-parameter-groups`comando.

Example

Nell'esempio seguente sono elencati tutti i gruppi di parametri del cluster database disponibili per un account AWS .

```
aws rds describe-db-cluster-parameter-groups
```

L'esempio seguente descrive il gruppo di parametri mydbclusterparametergroup.

Per LinuxmacOS, oUnix:

```
aws rds describe-db-cluster-parameter-groups \
    --db-cluster-parameter-group-name mydbclusterparametergroup
```

Per Windows:

```
aws rds describe-db-cluster-parameter-groups ^
    --db-cluster-parameter-group-name mydbclusterparametergroup
```

Questo comando restituisce una risposta simile alla seguente:

```
{
    "DBClusterParameterGroups": [
        {
            "DBClusterParameterGroupName": "mydbclusterparametergroup2",
            "DBParameterGroupFamily": "mysql8.0",
            "Description": "My new cluster parameter group",
            "DBClusterParameterGroupArn": "arn:aws:rds:us-east-1:123456789012:cluster-
pg:mydbclusterparametergroup"
        }
    ]
}
```

API RDS

Per elencare tutti i gruppi di parametri del cluster DB per un AWS account, utilizza l'`DescribeDBClusterParameterGroups`azione API RDS.

Visualizzazione dei valori dei parametri per un gruppo di parametri del cluster database

Puoi ottenere un elenco di tutti i parametri in un gruppo di parametri del cluster database e dei rispettivi valori.

Console

Per visualizzare i valori dei parametri per un gruppo di parametri del cluster database

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione scegliere Parameter groups (Gruppi di parametri).

Il gruppo di parametri cluster database viene visualizzato nell'elenco con Type (Tipo) impostato su DB cluster parameter group (Gruppo di parametri cluster database).

3. Scegliere il nome del gruppo di parametri cluster di database per visualizzare il relativo elenco di parametri.

AWS CLI

Per visualizzare i valori dei parametri per un gruppo di parametri del cluster DB, usa il AWS CLI `describe-db-cluster-parameters`comando con il seguente parametro obbligatorio.

- `--db-cluster-parameter-group-name`

Example

L'esempio seguente elenca i parametri e i valori dei parametri per un gruppo di parametri cluster di database denominato mydbclusterparametergroup, in formato JSON.

Questo comando restituisce una risposta simile alla seguente:

```
aws rds describe-db-cluster-parameters --db-cluster-parameter-group-
name mydbclusterparametergroup
```

```
{
    "Parameters": [
        {
            "ParameterName": "activate_all_roles_on_login",
            "ParameterValue": "0",
            "Description": "Automatically set all granted roles as active after the
 user has authenticated successfully.",
            "Source": "engine-default",
            "ApplyType": "dynamic",
            "DataType": "boolean",
            "AllowedValues": "0,1",
            "IsModifiable": true,
            "ApplyMethod": "pending-reboot",
            "SupportedEngineModes": [
                "provisioned"
            ]
        },
```

```
        {
            "ParameterName": "allow-suspicious-udfs",
            "Description": "Controls whether user-defined functions that have only an
 xxx symbol for the main function can be loaded",
            "Source": "engine-default",
            "ApplyType": "static",
            "DataType": "boolean",
            "AllowedValues": "0,1",
            "IsModifiable": false,
            "ApplyMethod": "pending-reboot",
            "SupportedEngineModes": [
                "provisioned"
            ]
        },
...
```

API RDS

Per visualizzare i valori dei parametri per un gruppo di parametri cluster database, utilizza il comando API RDS `DescribeDBClusterParameters` con il parametro obbligatorio seguente.

- `DBClusterParameterGroupName`

In alcuni casi, i valori consentiti per un parametro non vengono visualizzati. Questi sono sempre parametri in cui l'origine è l'impostazione predefinita del motore di database.

Per visualizzare i valori di questi parametri, puoi eseguire le seguenti istruzioni SQL:

- MySQL:

```
-- Show the value of a particular parameter
mysql$ SHOW VARIABLES LIKE '%parameter_name%';

-- Show the values of all parameters
mysql$ SHOW VARIABLES;
```

- PostgreSQL:

```
-- Show the value of a particular parameter
postgresql=> SHOW parameter_name;

-- Show the values of all parameters
```

```
postgresql=> SHOW ALL;
```

Eliminazione di un gruppo di parametri del cluster DB

È possibile eliminare un gruppo di parametri del cluster DB utilizzando AWS Management Console AWS CLI, o l'API RDS. Un gruppo di parametri del gruppo di parametri del cluster DB può essere eliminato solo se non è associato a un cluster DB.

Console

Per eliminare i gruppi di parametri

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione scegliere Parameter groups (Gruppi di parametri).

   I gruppi di parametri vengono visualizzati in un elenco.
3. Scegli il nome dei gruppi di parametri del cluster DB da eliminare.
4. Scegli Azioni e poi Elimina.
5. Controllate i nomi dei gruppi di parametri, quindi scegliete Elimina.

AWS CLI

Per eliminare un gruppo di parametri del cluster DB, utilizzate il AWS CLI `delete-db-cluster-parameter-group`comando con il seguente parametro richiesto.

- `--db-parameter-group-name`

Example

L'esempio seguente elimina un gruppo di parametri del cluster DB denominato mydbparametergroup.

```
aws rds delete-db-cluster-parameter-group --db-parameter-group-name mydbparametergroup
```

API RDS

Per eliminare un gruppo di parametri del cluster DB, utilizzate il `DeleteDBClusterParameterGroup`comando API RDS con il seguente parametro obbligatorio.

- `DBParameterGroupName`

Confronto di gruppi di parametri database

È possibile utilizzare il AWS Management Console per visualizzare le differenze tra due gruppi di parametri DB.

I gruppi di parametri specificati devono essere entrambi gruppi di parametri database o gruppi di parametri cluster database, anche se il motore e la versione del database sono uguali. Ad esempio, non è possibile confrontare un gruppo di parametri DB `aurora-mysql8.0` (Aurora MySQL versione 3) e un gruppo di parametri del cluster DB. `aurora-mysql8.0`

È possibile confrontare i gruppi di parametri database Aurora MySQL e RDS per MySQL, anche per versioni diverse, ma non è possibile confrontare i gruppi di parametri database Aurora PostgreSQL e RDS per PostgreSQL.

Per confrontare due gruppi di parametri DB

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione scegliere Parameter groups (Gruppi di parametri).
3. Nell'elenco, scegliere i due gruppi di parametri da confrontare.

   > **Note**
   >
   > Per confrontare un gruppo di parametri predefinito con un gruppo di parametri personalizzato, scegli prima il gruppo di parametri predefinito nella scheda Predefinito, quindi scegli il gruppo di parametri personalizzati nella scheda Personalizzato.

4. Da Azioni, scegliete Confronta.

Specifica dei parametri del database

I tipi di parametri database comprendono:

- Numero intero
- Boolean

- Stringa
- Long
- Doppio
- Timestamp
- Oggetto di altri tipi di dati definiti
- Array di valori di tipo integer, booleano, string, long, double, timestamp o oggetto

È inoltre possibile specificare parametri interi e booleani utilizzando espressioni, formule e funzioni.

Per il motore Oracle, è possibile utilizzare la formula `DBInstanceClassHugePagesDefault` per specificare un parametro database booleano. Per informazioni, consulta Variabili di formula dei parametri database.

Per il motore PostgreSQL, per specificare un parametro database booleano puoi utilizzare un'espressione. Per informazioni, consulta Espressioni dei parametri database booleani.

Indice

Formule dei parametri database

Una formula per un parametro database è un'espressione che restituisce un valore intero o un valore booleano. L'espressione va racchiusa tra parentesi graffe: {}. Puoi utilizzare una formula per un valore di parametro database o come argomento per una funzione di parametro database.

Sintassi

```
{FormulaVariable}
{FormulaVariable*Integer}
{FormulaVariable*Integer/Integer}
```

```
{FormulaVariable/Integer}
```

Variabili di formula dei parametri database

Ogni variabile di formula restituisce un valore intero o booleano. I nomi delle variabili fanno distinzione tra maiuscole e minuscole.

AllocatedStorage

Restituisce un numero intero che rappresenta la dimensione, in byte, del volume di dati.

DB InstanceClassHugePagesDefault

Restituisce un valore booleano. Al momento, è supportata solo per i motori Oracle.

Per ulteriori informazioni, consulta Attivazione di HugePages per un'istanza RDS per Oracle.

DB InstanceClassMemory

Restituisce un numero intero per il numero di byte di memoria disponibili per il processo del database. Questo numero viene calcolato internamente a partire dalla quantità totale di memoria per la classe di istanza database. Da questo valore, il calcolo sottrae la memoria riservata al sistema operativo e ai processi RDS che gestiscono l'istanza. Pertanto, il numero è sempre leggermente inferiore alle figure di memoria mostrate nelle tabelle della classe di istanza in Classi di istanze database. Il valore esatto dipende da una combinazione di fattori: come classe di istanza, motore di database e se si applica a un'istanza RDS o a un'istanza che fa parte di un cluster Aurora.

DBInstanceVCPU

Restituisce un numero intero che rappresenta il numero di unità di elaborazione centrali virtuali (vCPU) utilizzate da Amazon RDS per gestire l'istanza. Attualmente, è supportato solo per il motore RDS per PostgreSQL.

EndPointPort

Restituisce un numero intero che rappresenta la porta utilizzata durante la connessione all'istanza database.

TrueIfReplica

Restituisce 1 se l'istanza database è una replica di lettura e `0` in caso contrario. Questo è il valore predefinito per il parametro `read_only` in MySQL.

Operatori delle formule dei parametri database

Le formule dei parametri database supportano due operatori, di divisione e di moltiplicazione.

Operatore di divisione: /

Divide il dividendo per il divisore, restituendo un quoziente intero. I decimali nel quoziente vengono troncati, non arrotondati.

Sintassi

```
dividend / divisor
```

Gli argomenti del dividendo e del divisore devono essere espressioni intere.

Operatore di moltiplicazione: *

Moltiplica le espressioni, restituendone il prodotto. I decimali nelle espressioni vengono troncati, non arrotondati.

Sintassi

```
expression * expression
```

Entrambe le espressioni devono essere valori interi.

Funzioni dei parametri database

Puoi specificare gli argomenti delle funzioni dei parametri database come numeri interi o formule. Ogni funzione deve avere almeno un argomento. Specifica più argomenti come elenco separato da virgole. L'elenco non può includere membri vuoti, come argomento1,,argomento3. I nomi di funzione non fanno distinzione tra maiuscole e minuscole.

IF

Restituisce un argomento.

Al momento, è supportata solo per i motori Oracle e l'unico primo argomento supportato è `{DBInstanceClassHugePagesDefault}`. Per ulteriori informazioni, consulta Attivazione di HugePages per un'istanza RDS per Oracle.

Sintassi

```
IF(argument1, argument2, argument3)
```

Restituisce il secondo argomento se il primo argomento restituisce true. In caso contrario, restituisce il terzo argomento.

GREATEST

Restituisce il valore più grande da un elenco di valori interi o formule di parametro.

Sintassi

```
GREATEST(argument1, argument2,...argumentn)
```

Restituisce un integer.

LEAST

Restituisce il valore più piccolo da un elenco di valori interi o formule di parametro.

Sintassi

```
LEAST(argument1, argument2,...argumentn)
```

Restituisce un integer.

SUM

Aggiunge i valori delle formule di parametro o dei numeri interi specificati.

Sintassi

```
SUM(argument1, argument2,...argumentn)
```

Restituisce un integer.

Espressioni dei parametri database booleani

Un'espressione di parametro database booleano viene risolta in un valore booleano 1 o 0. L'espressione è racchiusa tra virgolette.

> **Note**
>
> Le espressioni dei parametri database booleani sono supportate solo per il motore PostgreSQL.

Sintassi

```
"expression operator expression"
```

Entrambe le espressioni devono essere risolte in numeri interi. Un'espressione può essere la seguente:

- Costante intera
- Formula di parametro database
- Funzione di parametro database
- Variabile di parametro database

Le espressioni dei parametri database booleani supportano i seguenti operatori di disuguaglianza:

L'operatore maggiore di: **>**

Sintassi

```
"expression > expression"
```

L'operatore minore di: **<**

Sintassi

```
"expression < expression"
```

Gli operatori maggiore o uguale: **>=**, **=>**

Sintassi

```
"expression >= expression"
"expression => expression"
```

Gli operatori minore o uguale: <=, =<

Sintassi

```
"expression <= expression"
"expression =< expression"
```

Example utilizzo di un'espressione del parametro database booleano

Nell'esempio di espressione del parametro database booleano seguente viene confrontato il risultato di una formula di parametro con un numero intero per modificare il parametro database booleano `wal_compression` per un'istanza database PostgreSQL. L'espressione del parametro confronta il numero di vCPU con il valore 2. Se il numero di vCPU è maggiore di 2, il parametro database `wal_compression` è impostato su true.

```
aws rds modify-db-parameter-group --db-parameter-group-name group-name \
--parameters "ParameterName=wal_compression,ParameterValue=\"{DBInstanceVCPU} > 2\" "
```

Espressioni di log di parametri database

Puoi impostare un valore del parametro database intero in una espressione di log. L'espressione va racchiusa tra parentesi graffe: {}. Ad esempio:

```
{log(DBInstanceClassMemory/8187281418)*1000}
```

La funzione `log` rappresenta la base di log 2. In questo esempio viene utilizzato anche la variabile di formula `DBInstanceClassMemory`. Per informazioni, consulta Variabili di formula dei parametri database.

> **Note**
>
> Al momento, non puoi specificare il parametro `innodb_log_file_size` MySQL con un valore diverso da un numero intero.

Esempi di valori dei parametri database

Questi esempi illustrano l'utilizzo di formule, funzioni ed espressioni per i valori dei parametri database.

> **Warning**
>
> L'impostazione errata dei parametri in un gruppo di parametri database può avere effetti negativi non intenzionali. Questi potrebbero includere prestazioni ridotte e instabilità del sistema. Presta sempre attenzione quando modifichi i parametri database ed esegui il backup dei dati prima di modificare il gruppo di parametri database. Prova le modifiche ai gruppi di parametri su un'istanza DB di test, creata utilizzando point-in-time-restores, prima di applicare tali modifiche al gruppo di parametri alle istanze DB di produzione.

Example utilizzando la funzione del parametro database GREATEST

È possibile specificare la funzione `GREATEST` in un parametro del processo Oracle. Usala per impostare il numero di processi utente su un valore maggiore di 80 o `DBInstanceClassMemory` diviso per 9.868.951.

```
GREATEST({DBInstanceClassMemory/9868951},80)
```

Example utilizzando la funzione di parametro database LEAST

È possibile specificare la funzione `LEAST` in in valore del parametro `max_binlog_cache_size` MySQL. Usalo per impostare la dimensione massima della cache che una transazione può utilizzare in un'istanza MySQL su un valore minore di 1 MB o `DBInstanceClass`/256.

```
LEAST({DBInstanceClassMemory/256},10485760)
```

Creazione di una ElastiCache cache Amazon utilizzando le impostazioni dell'istanza database di

ElastiCache è un servizio di caching in memoria completamente gestito che fornisce latenze di lettura e scrittura in microsecondi che supportano casi d'uso flessibili e in tempo reale. ElastiCache può aiutarti ad accelerare le prestazioni di applicazioni e database. È possibile utilizzarlo ElastiCache come archivio dati primario per casi d'uso che non richiedono la durabilità dei dati, ad esempio classiifiche di gioco, streaming e analisi dei dati. ElastiCache aiuta a rimuovere la complessità associata all'implementazione e alla gestione di un ambiente di elaborazione distribuito. Per ulteriori informazioni, consulta Casi ElastiCache d'uso comuni e How ElastiCache Can Help for Memcached and Common ElastiCache Use Cases e How ElastiCache Can Help for Redis. Puoi utilizzare la console Amazon RDS per creare ElastiCache cache.

Puoi utilizzare Amazon ElastiCache in due formati. Puoi iniziare con una cache serverless o scegliere di progettare il tuo cluster di cache. Se scegli di progettare il tuo cluster di cache, ElastiCache funziona con entrambi i motori Redis e Memcached. Se non sei sicuro del motore da usare, consulta Confronto tra Memcached e Redis. Per ulteriori informazioni su Amazon ElastiCache, consulta la Amazon ElastiCache User Guide.

Argomenti

- Panoramica della creazione di ElastiCache cache con le impostazioni dell'istanza RDS
- Creazione di una ElastiCache cache con impostazioni da un'

Panoramica della creazione di ElastiCache cache con le impostazioni dell'istanza RDS

Puoi creare una ElastiCache cache da Amazon RDS utilizzando le stesse impostazioni di configurazione di un'istanza DB RDS del cluster appena creata o esistente.

Alcuni casi d'uso per associare una ElastiCache cache all'istanza :

- È possibile risparmiare sui costi e migliorare le prestazioni utilizzando ElastiCache RDS anziché eseguendo solo RDS.

  Ad esempio, puoi risparmiare fino al 55% sui costi e ottenere prestazioni di lettura fino a 80 volte più veloci utilizzando ElastiCache RDS for MySQL anziché solo RDS per MySQL.

- È possibile utilizzare la ElastiCache cache come archivio dati principale per applicazioni che non richiedono la durabilità dei dati. Le applicazioni che utilizzano Redis o Memcached possono essere utilizzate quasi ElastiCache senza modifiche.

Quando si crea una ElastiCache cache da RDS, la ElastiCache cache eredita le seguenti impostazioni dall'istanza DB RDS del :

- ElastiCache impostazioni di connettività
- ElastiCache impostazioni di sicurezza

È inoltre possibile configurare le impostazioni di configurazione della cache in base alle proprie esigenze.

Configurazione ElastiCache nelle tue applicazioni

Le applicazioni devono essere configurate per utilizzare la ElastiCache cache. È inoltre possibile ottimizzare e migliorare le prestazioni della cache configurando le applicazioni in modo che utilizzino strategie di memorizzazione nella cache in base alle proprie esigenze.

- Per accedere alla ElastiCache cache e iniziare, consulta Guida introduttiva ad Amazon ElastiCache per Redis e Guida introduttiva ad Amazon ElastiCache for Memcached.
- Per ulteriori informazioni sulle strategie di caching, consulta Best practice e strategie di caching per Memcached e Best practice e strategie di caching per Redis.
- Per ulteriori informazioni sull'alta disponibilità nei ElastiCache cluster Redis, consulta Alta disponibilità con gruppi di replica.
- Potrebbero essere sostenuti costi associati allo storage di backup, al trasferimento dei dati all'interno o tra regioni o all'uso di. AWS Outposts Per i dettagli sui prezzi, consulta la pagina ElastiCache dei prezzi di Amazon.

Creazione di una ElastiCache cache con impostazioni da un'

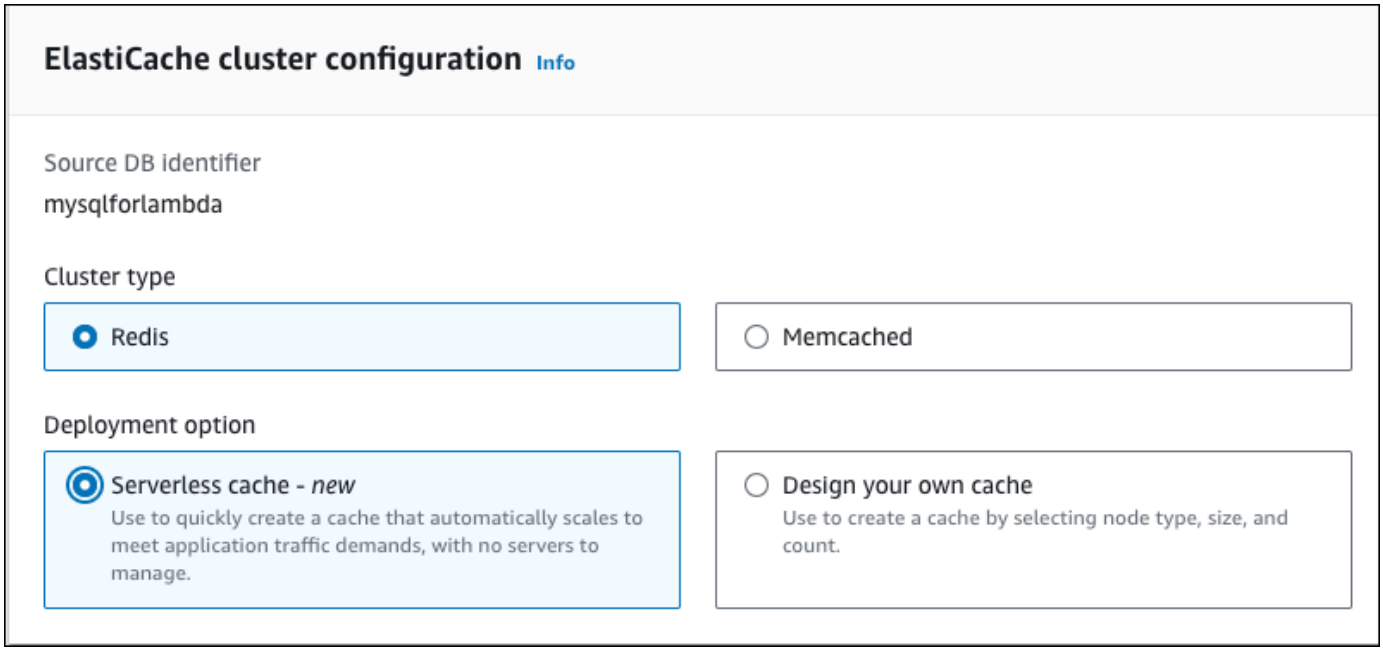
1. Per creare un'istanza database, segui le istruzioni riportate in Creazione di un'istanza database Amazon RDS.

2. Dopo aver creato un'istanza DB, la console visualizza la finestra Componenti aggiuntivi consigliati. Seleziona Crea un ElastiCache cluster da RDS utilizzando le impostazioni del database.

   Per un database esistente, nella pagina Database, seleziona l'istanza del DB richiesta. Nel menu a discesa Azioni, scegli Crea ElastiCache cluster per creare una ElastiCache cache in RDS con le stesse impostazioni dell'istanza DB RDS del .

   Nella sezione di ElastiCache configurazione, l'identificatore Source DB mostra da quale istanza del DB la cache eredita le ElastiCache impostazioni.

3. Scegli se desideri creare un cluster Redis o Memcached. Per ulteriori informazioni, consulta Confronto tra Memcached e Redis.



4. Dopodiché, scegli se vuoi creare una cache serverless o progettare la tua cache. Per ulteriori informazioni, consulta Scelta tra le opzioni di distribuzione.

   Se scegli Serverless cache:

   a. Nelle impostazioni della cache, inserisci i valori per Nome e Descrizione.

   b. In Visualizza impostazioni predefinite, lascia le impostazioni predefinite per stabilire la connessione tra la cache e l'istanza DB.

   c. Puoi anche modificare le impostazioni predefinite scegliendo Personalizza impostazioni predefinite. Seleziona le impostazioni di ElastiCache connettività, le impostazioni ElastiCache di sicurezza e i limiti massimi di utilizzo.

5. Se scegli Progetta la tua cache:

   a. Se hai scelto il cluster Redis, scegli se vuoi mantenere la modalità cluster abilitata o disabilitata. Per ulteriori informazioni, consulta Replica: Redis (modalità cluster disabilitata) e Redis (modalità cluster abilitata).

   b. Immetti i valori per Nome, Descrizione e Versione del motore.

      Per Versione del motore, il valore predefinito consigliato è la versione più recente del motore. Puoi anche scegliere una versione di Engine per la ElastiCache cache che meglio soddisfa i tuoi requisiti.

   c. Scegli il tipo di nodo per l'opzione Tipo di nodo. Per ulteriori informazioni, consulta Gestione di nodi.

      Se scegli di creare un cluster Redis con la Modalità cluster impostata su Abilitato, inserisci il numero di partizioni (partizioni/gruppi di nodi) per l'opzione Numero di partizioni.

      Immetti il numero di repliche di ogni partizione per l'opzione Numero di repliche.

      > **Note**
      >
      > Il tipo di nodo selezionato, il numero di shard e il numero di repliche influiscono tutti sulle prestazioni della cache e sui costi delle risorse. Assicurati che queste impostazioni corrispondano alle esigenze del tuo database. Per informazioni sui prezzi, consulta la pagina ElastiCache dei prezzi di Amazon.

   d. Seleziona le impostazioni di ElastiCache connettività e le impostazioni ElastiCache di sicurezza. È possibile mantenere le impostazioni predefinite o personalizzarle in base alle proprie esigenze.

6. Verifica le impostazioni predefinite ed ereditate della ElastiCache cache. Alcune impostazioni non possono essere modificate dopo la creazione.

   > **Note**
   >
   > RDS potrebbe modificare la finestra di backup della ElastiCache cache per soddisfare il requisito minimo di 60 minuti. La finestra di backup del database di origine rimane invariata.

7. Quando sei pronto, scegli Crea ElastiCache cache.

La console visualizza un banner di conferma per la creazione ElastiCache della cache. Segui il link contenuto nel banner verso la ElastiCache console per visualizzare i dettagli della cache. La ElastiCache console visualizza la ElastiCache cache appena creata.

Gestione di un'istanza database di Amazon RDS

Di seguito, è possibile trovare le istruzioni per la dell'istanza database Amazon RDS.

Argomenti

Arresto temporaneo di un'istanza database Amazon RDS

Puoi interrompere un'istanza DB a intermittenza per test temporanei o per un'attività di sviluppo quotidiana. Il caso d'uso più comune è l'ottimizzazione dei costi.

> **Note**
>
> In alcuni casi, è necessario molto tempo per arrestare un'istanza DB. Per arrestare l'istanza DB e riavviarla immediatamente, riavvia l'istanza DB. Per ulteriori informazioni, consulta Riavvio di un'istanza database.

Argomenti

Casi d'uso per arrestare l'istanza DB

L'arresto e l'avvio di un'istanza DB sono più rapidi rispetto alla creazione di uno snapshot DB, all'eliminazione dell'istanza DB e al ripristino dello snapshot quando si desidera accedere all'istanza. I casi d'uso più comuni per l'arresto di un'istanza includono i seguenti:

- Ottimizzazione dei costi: per i database non di produzione, puoi interrompere temporaneamente l'istanza DB di Amazon RDS per risparmiare denaro. Mentre l'istanza è interrotta, non ti vengono addebitati costi per le ore dell'istanza DB.

  > **Important**
  >
  > Mentre l'istanza database è arrestata, ti viene addebitato l'archivio assegnato, inclusa la capacità di IOPS allocata. Ti viene addebitato anche l'archivio dei backup, incluso quello

per gli snapshot manuali e i backup automatici all'interno della finestra di conservazione specificata. Tuttavia, non è previsto alcun costo per le ore dell'istanza database. Per ulteriori informazioni, consulta le domande frequenti sulla fatturazione.

- Sviluppo quotidiano: se gestisci un'istanza DB per scopi di sviluppo, puoi avviare l'istanza quando è necessario e poi chiuderla quando non è necessaria.
- Test: potrebbe essere necessaria un'istanza DB temporanea per testare le procedure di backup e ripristino, le migrazioni, gli aggiornamenti delle applicazioni o le attività correlate. In questi casi d'uso, è possibile interrompere l'istanza DB quando non è necessaria.
- Formazione: se stai svolgendo un corso di formazione in RDS, potresti dover avviare le istanze DB durante la sessione di formazione e chiuderle in seguito.

Motori di database, classi di istanza e regioni supportati

Puoi arrestare e avviare le istanze database Amazon RDS in esecuzione nei motori seguenti:

- Db2
- MariaDB
- Microsoft SQL Server, incluso RDS Custom per SQL Server
- MySQL
- Oracle
- PostgreSQL

L'arresto e l'avvio di un'istanza database sono supportati per tutte le classi di istanza database e in tutte le regioni AWS .

Arresto di un'istanza database in una implementazione multi-AZ

È possibile interrompere e avviare un'istanza DB in una distribuzione Multi-AZ. Nota i seguenti limiti:

- È possibile creare una distribuzione Multi-AZ solo se il motore di database la supporta. Per ulteriori informazioni sul supporto del motore, vedereRegioni e motori DB supportati per cluster DB Multi-AZ in Amazon RDS.
- RDS per SQL Server non supporta l'arresto di un'istanza DB in una distribuzione Multi-AZ. Per ulteriori informazioni, consulta Limitazioni, note e suggerimenti per l'implementazione di Multi-AZ di Microsoft SQL Server.

- Potrebbe essere necessario molto tempo per arrestare un'istanza DB. Se si dispone di almeno un backup dopo un failover precedente, è possibile velocizzare l'operazione di arresto eseguendo un riavvio con operazione di failover. Per ulteriori informazioni, consulta Riavvio di un'istanza database.

Arresto di un'istanza database

L'operazione di arresto si verifica nelle seguenti fasi:

1. L'istanza database avvia il normale processo di arresto.

   Lo stato dell'istanza database diventa `stopping`.
2. L'istanza smette di funzionare, fino a un massimo di 7 giorni consecutivi.

   Lo stato dell'istanza database diventa `stopped`.

Caratteristiche di un'istanza DB interrotta

Quando si trova in uno stato interrotto, l'istanza DB presenta le seguenti caratteristiche:

- L'istanza DB interrotta mantiene quanto segue:
  - ID istanza
  - Endpoint DNS (Domain Name Server)
  - Gruppo di parametri
  - Gruppo di sicurezza
  - Option group (Gruppo di opzioni)
  - Registri delle transazioni di Amazon S3 (necessari per il ripristino) point-in-time

  Quando avvii un'istanza database, la configurazione è uguale a quella presente al momento dell'arresto.
- Tutti i volumi di storage restano collegati all'istanza database e i dati vengono conservati. RDS elimina eventuali dati archiviati nella RAM dell'istanza database.

  Mentre l'istanza database è arrestata, ti viene addebitato l'archivio assegnato, inclusa la capacità di IOPS allocata. Ti viene addebitato anche l'archivio dei backup, incluso quello per gli snapshot manuali e i backup automatici all'interno della finestra di conservazione specificata.

- RDS rimuove le azioni in sospeso, inclusi gli aggiornamenti di manutenzione pianificati, ad eccezione delle azioni in sospeso per il gruppo di opzioni o il gruppo di parametri DB dell'istanza DB.

> **Note**
>
> Occasionalmente, un'istanza database RDS for PostgreSQL non si arresta in modo pulito. Se ciò accade, si vede che l'istanza passa attraverso un processo di ripristino quando viene riavviata in un secondo momento. Questo comportamento è previsto dal motore del database destinato a proteggere l'integrità del database. Alcune statistiche e contatori basati sulla memoria non conservano la cronologia e vengono reinizializzati dopo il riavvio, per acquisire il carico di lavoro operativo che avanza.

Riavvio automatico di un'istanza DB interrotta

Se non avvii manualmente l'istanza database dopo sette giorni consecutivi di arresto, RDS avvia automaticamente l'istanza database. In questo modo, l'istanza non rimane indietro rispetto agli aggiornamenti di manutenzione richiesti. Per informazioni su come arrestare e avviare l'istanza in base a una pianificazione, consulta Come posso utilizzare Step Functions per interrompere un'istanza Amazon RDS per più di 7 giorni?.

Limitazioni relative all'arresto dell'istanza database

Di seguito sono elencate alcune limitazioni relative all'arresto e all'avvio di un'istanza database:

- Non è possibile interrompere un'istanza DB RDS for SQL Server in una distribuzione Multi-AZ.
- Non è possibile arrestare un'istanza database che dispone di una replica di lettura o che costituisce una replica di lettura.
- Non è possibile modificare un'istanza database arrestata.
- Non è possibile eliminare un gruppo di opzioni associato a un'istanza database arrestata.
- Non è possibile eliminare un gruppo di parametri database associato a un'istanza database arrestata.
- In una implementazione multi-AZ, le zone di disponibilità primarie e secondarie potrebbero essere cambiate dopo l'avvio dell'istanza database.

Per RDS Custom per SQL Server sono valide limitazioni aggiuntive. Per ulteriori informazioni, consulta Avvio e arresto di un'istanza database RDS Custom per SQL Server.

Considerazioni su gruppi di parametri e opzioni

Non è possibile rimuovere le opzioni persistenti (incluse le opzioni permanenti) da un gruppo di opzioni se sono presenti istanze database associate a tale gruppo di opzioni. Questo aspetto è valido anche per le istanze database con stato `stopping` (arresto in corso), `stopped` (arrestata) o `starting` (avvio in corso).

Puoi modificare il gruppo di opzioni o il gruppo di parametri database associato a un'istanza database arrestata. Tuttavia, la modifica viene applicata solo al successivo avvio dell'istanza database. Se scegli di applicare le modifiche immediatamente, la modifica viene applicata all'avvio dell'istanza database. In caso contrario, la modifica viene applicate durante la finestra di manutenzione successiva dopo l'avvio dell'istanza database.

Considerazioni sugli indirizzi IP pubblici

Quando arresti un'istanza database, l'endpoint DNS viene conservato. Se si arresta un'istanza database che dispone di un indirizzo IP pubblico, Amazon RDS rilascia il suo indirizzo IP pubblico. Quando l'istanza database viene riavviata, avrà un indirizzo IP pubblico diverso.

> **Note**
>
> Stabilisci sempre la connessione a un'istanza database usando l'endpoint DNS e non l'indirizzo IP.

Arresto temporaneo di un'istanza DB: passaggi di base

È possibile interrompere un DB utilizzando l' AWS Management Console AWS CLI, the o l'API RDS.

Console

Per arrestare un'istanza database

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, scegliere Databases (Database) e l'istanza database da arrestare.

3. Per Actions (Operazioni), scegli Stop temporarily (Arresta temporaneamente).
4. Nella finestra Stop DB instance temporarily, (Arresto temporaneo di un'istanza database) seleziona la conferma per il riavvio automatico dell'istanza database dopo 7 giorni.
5. (Facoltativo) Seleziona Save the DB instance in a snapshot (Salva l'istanza database in uno snapshot) e immetti il nome dello snapshot in Snapshot name (Nome snapshot). Scegli questa opzione per creare uno snapshot dell'istanza database prima di arrestarla.
6. Scegli Stop temporarily (Arresta temporaneamente) per arrestare l'istanza database oppure Cancel (Annulla) per annullare l'operazione.

AWS CLI

Per interrompere un'istanza DB utilizzando il AWS CLI, chiama il stop-db-instancecomando con la seguente opzione:

- `--db-instance-identifier` – Nome dell'istanza database.

Example

```
aws rds stop-db-instance --db-instance-identifier mydbinstance
```

API RDS

Per arrestare un'istanza database tramite l'API Amazon RDS, chiamare l'operazione StopDBInstance con il parametro seguente:

- `DBInstanceIdentifier` – Nome dell'istanza database.

Avvio di un'istanza database Amazon RDS arrestata in precedenza

Puoi arrestare temporaneamente l'istanza database Amazon RDS per risparmiare denaro. Dopo avere arrestato l'istanza database, puoi riavviarla per iniziare di nuovo a usarla. Per ulteriori informazioni sull'arresto e sull'avvio delle istanze database, consulta Arresto temporaneo di un'istanza database Amazon RDS.

Quando avvii un'istanza database arrestata in precedenza, l'istanza database conserva determinate informazioni. Queste informazioni sono ID, endpoint DNS (Domain Name Server), gruppo di parametri, gruppo di sicurezza e gruppo di opzioni. Quando avvii un'istanza arrestata, viene addebitato il costo di un'intera ora dell'istanza.

Console

Per avviare un'istanza database

1. Accedi alla AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, scegliere Databases (Database) e l'istanza database da avviare.
3. In Actions (Operazioni), scegliere Start (Avvia).

AWS CLI

Per avviare un'istanza database tramite AWS CLI, chiamare il comando start-db-instance con l'opzione seguente:

- `--db-instance-identifier` – Il nome dell'istanza database.

Example

```
aws rds start-db-instance --db-instance-identifier mydbinstance
```

API RDS

Per avviare un'istanza database tramite l'API Amazon RDS, chiamare l'operazione StartDBInstance con i parametri seguenti:

- `DBInstanceIdentifier` – Il nome dell'istanza database.

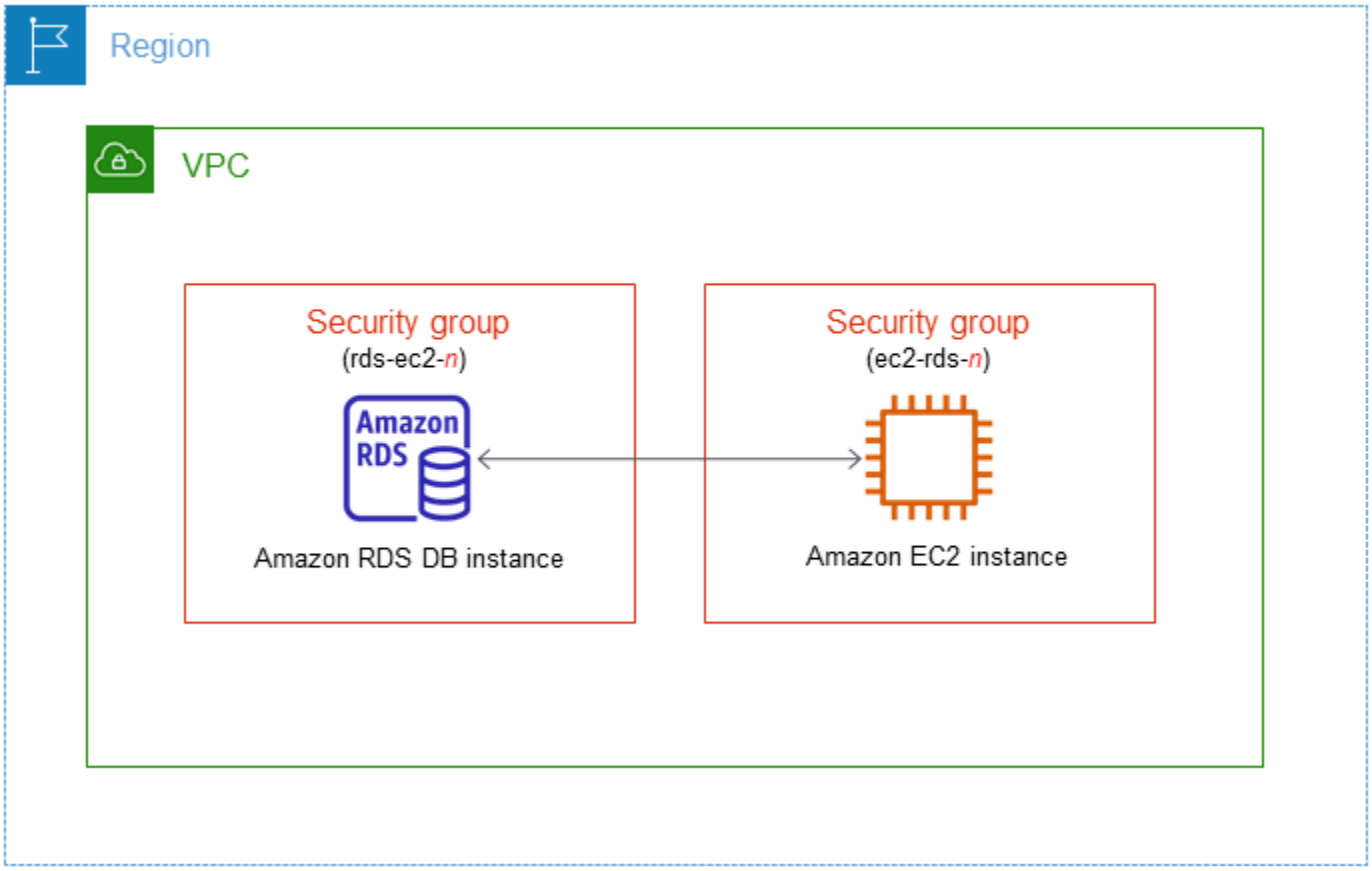
Connessione automatica di una risorsa di calcolo AWS e di un'istanza database

È possibile connettersi automaticamente a un'istanza database e risorse di calcolo AWS quali istanze di Amazon Elastic Compute Cloud (Amazon EC2) e funzioni AWS Lambda.

Argomenti

Connessione automatica di un'istanza EC2 e di un'istanza database

È possibile usare la console Amazon RDS per semplificare l'impostazione di una connessione tra un'istanza Amazon Elastic Compute Cloud (Amazon EC2) e un'istanza database. Spesso, l'istanza database si trova in una sottorete privata e l'istanza EC2 si trova in una sottorete pubblica all'interno di un VPC. È possibile usare un client SQL nell'istanza EC2 per connettersi all'istanza database . L'istanza EC2 può anche eseguire server o applicazioni web che accedono all'istanza database privata . Per istruzioni sulla configurazione di una connessione tra un'istanza EC2 e un cluster database Multi-AZ, consulta the section called “Connessione di un'istanza EC2 e un cluster di database multi-AZ”.



Se desideri connetterti a un'istanza EC2 che non si trova nello stesso VPC dell'istanza database, consulta gli scenari in Scenari per accedere a un'istanza database in un VPC.

Argomenti

- Panoramica della connettività automatica con un'istanza EC2
- Connessione automatica di un'istanza EC2 e di un database RDS
- Visualizzazione delle risorse di calcolo connesse
- Connessione a un'istanza database che esegue un motore DB specifico

Panoramica della connettività automatica con un'istanza EC2

Quando configuri una connessione tra un'istanza EC2 e un database RDS, Amazon RDS configura automaticamente il gruppo di sicurezza VPC per l'istanza EC2 e per il database RDS.

Di seguito sono riportati i requisiti per connettere un'istanza EC2 a un database RDS:

- L'istanza EC2 deve risiedere nello stesso VPC del database RDS.

Se nello stesso VPC non esistono istanze EC2, allora la console fornisce un collegamento per crearne una.

- L'utente che configura la connettività deve disporre delle autorizzazioni per eseguire le seguenti operazioni Amazon EC2:
  - `ec2:AuthorizeSecurityGroupEgress`
  - `ec2:AuthorizeSecurityGroupIngress`
  - `ec2:CreateSecurityGroup`
  - `ec2:DescribeInstances`
  - `ec2:DescribeNetworkInterfaces`
  - `ec2:DescribeSecurityGroups`
  - `ec2:ModifyNetworkInterfaceAttribute`
  - `ec2:RevokeSecurityGroupEgress`

Se l'istanza database e l'istanza EC2 si trovano in zone di disponibilità diverse, è possibile che all'account vengano addebitati costi tra zone di disponibilità.

Quando si configura una connessione a un'istanza EC2, Amazon RDS opera in base alla configurazione corrente dei gruppi di sicurezza associati al database RDS e all'istanza EC2, come descritto nella tabella seguente.

Configurazione del gruppo di sicurezza RDS corrente	Configurazione del gruppo di sicurezza EC2 corrente	Operazione RDS
Esistono uno o più gruppi di sicurezza associati al database RDS con un nome che corrisponde al modello `rds-ec2-`*n* (dove *n* è un numero). Un gruppo di sicurezza che corrispon de al modello non è stato modificato. Questo gruppo di sicurezza include una sola regola in uscita con il gruppo	Esistono uno o più gruppi di sicurezza associati all'istan za EC2 con un nome che corrisponde al modello `ec2-rds-`*n* (dove *n* è un numero). Un gruppo di sicurezza che corrisponde al modello non è stato modificato. Questo gruppo di sicurezza ha una sola regola in uscita con il	RDS non esegue alcuna operazione. Una connessione è già configurata automaticamente tra l'istanza EC2 e il database RDS. Poiché esiste già una connessione tra l'istanza EC2 e il database RDS, i gruppi di sicurezza non vengono modificati.

Configurazione del gruppo di sicurezza RDS corrente	Configurazione del gruppo di sicurezza EC2 corrente	Operazione RDS
di sicurezza VPC dell'istanza EC2 come origine.	gruppo di sicurezza VPC del database RDS come origine.	
Si applica una delle due condizioni seguenti: • Non esiste un gruppo di sicurezza associato al database RDS con un nome che corrisponde al modello `rds-ec2-`*n*. • Esistono uno o più gruppi di sicurezza associati al database RDS con un nome che corrisponde al modello `rds-ec2-`*n*. Tuttavia, Amazon RDS non può utilizzare nessuno di questi gruppi di sicurezza per la connessione all'istanza EC2. Amazon RDS non può utilizzare un gruppo di sicurezza che non dispone di una regola in entrata con il gruppo di sicurezza VPC dell'istanza EC2 come origine. Inoltre, Amazon RDS non può utilizzare un gruppo di sicurezza che è stato modificato. Esempi di modifiche sono l'aggiunta di una regola o la modifica della porta di una regola esistente.	Si applica una delle due condizioni seguenti: • Non esiste un gruppo di sicurezza associato all'istanza EC2 con un nome che corrisponde al modello `ec2-rds-`*n*. • Esistono uno o più gruppi di sicurezza associati all'istanza EC2 con un nome che corrisponde al modello `ec2-rds-`*n*. Tuttavia, Amazon RDS non può utilizzare nessuno di questi gruppi di sicurezza per la connessione al database RDS. Amazon RDS non può utilizzare un gruppo di sicurezza che non dispone di una regola in uscita con il gruppo di sicurezza VPC del database RDS come origine. Inoltre, Amazon RDS non può utilizzare un gruppo di sicurezza che è stato modificato.	RDS action: create new security groups

Configurazione del gruppo di sicurezza RDS corrente	Configurazione del gruppo di sicurezza EC2 corrente	Operazione RDS
Esistono uno o più gruppi di sicurezza associati al database RDS con un nome che corrisponde al modello `rds-ec2-`*n*. Un gruppo di sicurezza che corrispon de al modello non è stato modificato. Questo gruppo di sicurezza include una sola regola in uscita con il gruppo di sicurezza VPC dell'istanza EC2 come origine.	Esistono uno o più gruppi di sicurezza associati all'istan za EC2 con un nome che corrisponde al modello `ec2-rds-`*n*. Tuttavia, Amazon RDS non può utilizzare nessuno di questi gruppi di sicurezza per la connessio ne al database RDS. Amazon RDS non può utilizzare un gruppo di sicurezza che non dispone di una regola in uscita con il gruppo di sicurezza VPC del database RDS come origine. Inoltre, Amazon RDS non può utilizzare un gruppo di sicurezza che è stato modificat o.	RDS action: create new security groups
Esistono uno o più gruppi di sicurezza associati al database RDS con un nome che corrisponde al modello `rds-ec2-`*n*. Un gruppo di sicurezza che corrispon de al modello non è stato modificato. Questo gruppo di sicurezza include una sola regola in uscita con il gruppo di sicurezza VPC dell'istanza EC2 come origine.	Esiste un gruppo di sicurezza EC2 valido per la connessio ne, ma non è associato all'istanza EC2. Questo gruppo di sicurezza ha un nome che corrisponde al modello `ec2-rds-`*n*. Non è stato modificato. Dispone di una sola regola in uscita con il gruppo di sicurezza VPC del database RDS come origine.	RDS action: associate EC2 security group

Configurazione del gruppo di sicurezza RDS corrente	Configurazione del gruppo di sicurezza EC2 corrente	Operazione RDS
Si applica una delle due condizioni seguenti: • Non esiste un gruppo di sicurezza associato al database RDS con un nome che corrisponde al modello `rds-ec2-`*n*. • Esistono uno o più gruppi di sicurezza associati al database RDS con un nome che corrisponde al modello `rds-ec2-`*n*. Tuttavia, Amazon RDS non può utilizzare nessuno di questi gruppi di sicurezza per la connessione all'istanza EC2. Amazon RDS non può utilizzare un gruppo di sicurezza che non dispone di una regola in entrata con il gruppo di sicurezza VPC dell'istanza EC2 come origine. Inoltre, Amazon RDS non può utilizzare un gruppo di sicurezza che è stato modificato.	Esistono uno o più gruppi di sicurezza associati all'istanza EC2 con un nome che corrisponde al modello `ec2-rds-`*n*. Un gruppo di sicurezza che corrisponde al modello non è stato modificato. Questo gruppo di sicurezza ha una sola regola in uscita con il gruppo di sicurezza VPC del database RDS come origine.	RDS action: create new security groups

RDS: creazione di nuovi gruppi di sicurezza

Amazon RDS esegue le seguenti operazioni:

- Crea un nuovo gruppo di sicurezza che corrisponde al modello `rds-ec2-`*n*. Questo gruppo di sicurezza include una regola in uscita con il gruppo di sicurezza VPC dell'istanza EC2 come

origine. Questo gruppo di sicurezza è associato al database RDS e consente all'istanza EC2 di accedere al database RDS.

- Crea un nuovo gruppo di sicurezza che corrisponde al modello `ec2-rds-`*n*. Questo gruppo di sicurezza ha una regola in uscita con il gruppo di sicurezza VPC del cluster RDS come destinazione. Questo gruppo di sicurezza è associato all'istanza EC2 e consente all'istanza EC2 di inviare il traffico al database RDS.

Azione RDS: associazione del gruppo di sicurezza EC2

Amazon RDS associa il gruppo di sicurezza EC2 esistente, valido all'istanza EC2. Questo gruppo di sicurezza consente all'istanza EC2 di inviare traffico al database RDS.

Connessione automatica di un'istanza EC2 e di un database RDS

Prima di configurare una connessione tra un'istanza EC2 e un database RDS assicurati di aver soddisfatto i requisiti descritti in Panoramica della connettività automatica con un'istanza EC2.

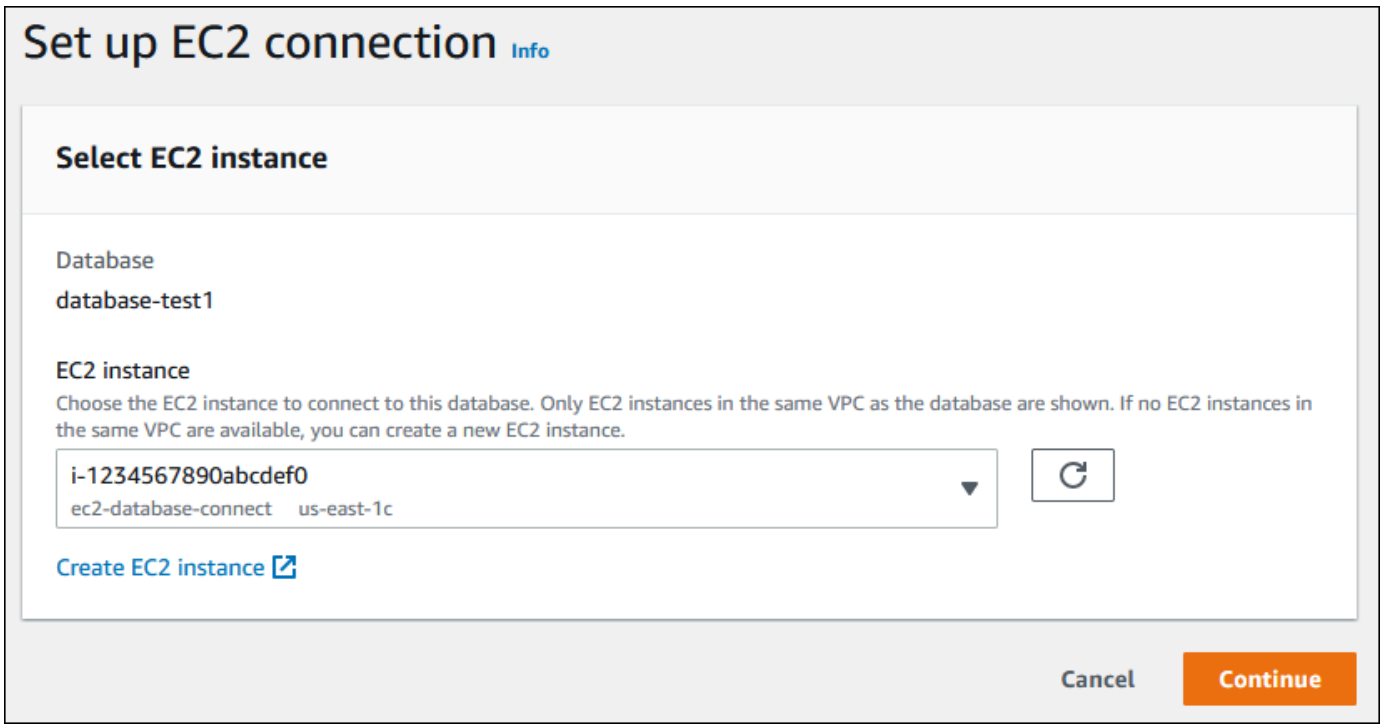
Se modifichi i gruppi di sicurezza dopo avere configurato la connettività, le modifiche potrebbero influenzare la connessione tra l'istanza EC2 e il database RDS.

> **Note**
>
> È possibile configurare automaticamente una connessione tra un'istanza EC2 e un database RDS solo utilizzando la AWS Management Console. Non è possibile configurare automaticamente una connessione con l'API AWS CLI o RDS.

Per connettere automaticamente un'istanza EC2 e un database RDS

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, scegli Databases (Database), quindi seleziona il database RDS.
3. In Operazioni, scegli Configura connessione EC2.

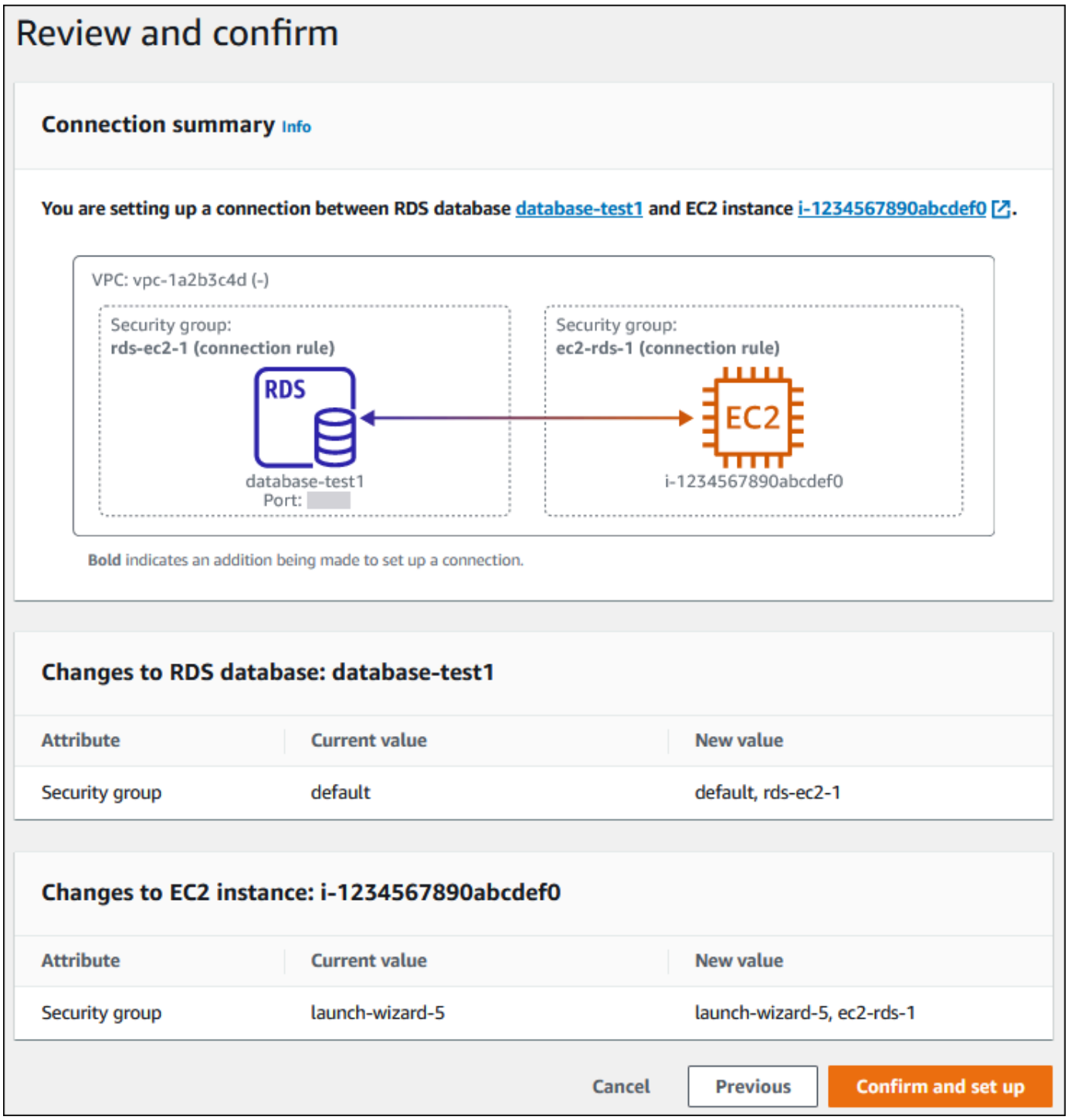
   Viene visualizzata la pagina Set up EC2 connection (Configura connessione EC2).
4. Nella pagina Set up EC2 connection (Configura connessione EC2), scegli l'istanza EC2.



Se nello stesso VPC non esistono istanze EC2, scegli Create EC2 instance (Crea istanza EC2) per crearne una. In questo caso, assicurati che la nuova istanza EC2 si trovi nello stesso VPC del database RDS.

5. Scegli Continua.

   Viene visualizzata la pagina Review and confirm (Rivedi e conferma).



6. Nella pagina Review and confirm (Rivedi e conferma), esamina le modifiche che RDS apporterà per configurare la connettività con l'istanza EC2.

   Se le modifiche sono corrette, scegli Conferma e configura.

   Se le modifiche non sono corrette, scegli Previous (Precedente) o Cancel (Annulla).

Visualizzazione delle risorse di calcolo connesse

È possibile utilizzare il AWS Management Console per visualizzare le risorse di calcolo connesse a un cluster DB di database RDS. Le risorse mostrate includono le connessioni delle risorse di calcolo configurate automaticamente. È possibile configurare la connettività delle risorse di calcolo automaticamente nei modi seguenti:

- È possibile selezionare la risorsa di calcolo quando si crea il database.

  Per ulteriori informazioni, consultare Creazione di un'istanza database Amazon RDS e Creazione di un cluster di database Multi-AZ.
- È possibile configurare la connettività tra un database esistente e una risorsa di calcolo.

  Per ulteriori informazioni, consulta Connessione automatica di un'istanza EC2 e di un database RDS.

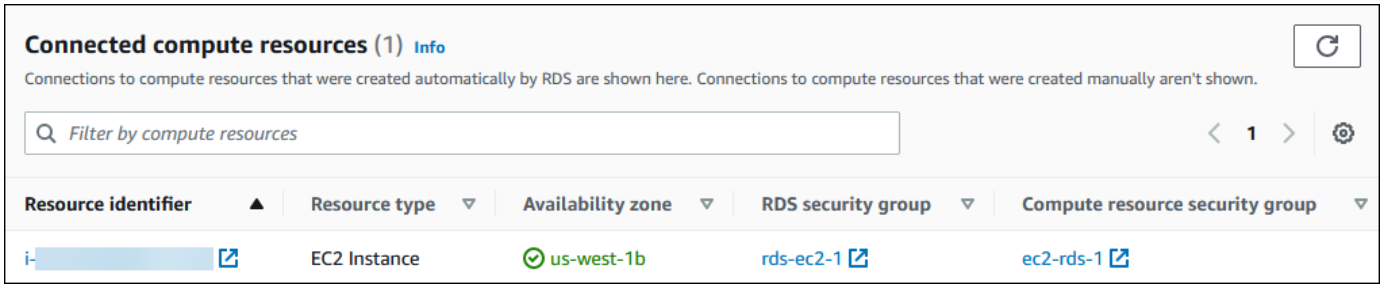
Le risorse di calcolo elencate non includono quelle connesse al database manualmente. Ad esempio, è possibile consentire manualmente a una risorsa di calcolo di accedere a un database aggiungendo una regola al gruppo di sicurezza VPC associato al database.

Per garantire la presenza della risorsa di calcolo nell'elenco, è necessario che siano soddisfatte le condizioni elencate di seguito.

- Il nome del gruppo di sicurezza associato alla risorsa di calcolo corrisponde al modello `ec2-rds-`*n* (dove *n* è un numero).
- Il gruppo di sicurezza associato alla risorsa di calcolo ha una regola in uscita con l'intervallo di porte impostato sulla porta utilizzata dal database RDS.
- Il gruppo di sicurezza associato alla risorsa di calcolo ha una regola in uscita con l'origine impostata su un gruppo di sicurezza associato al database RDS.
- Il nome del gruppo di sicurezza associato al database RDS corrisponde al modello `rds-ec2-`*n* (dove *n* è un numero).
- Il gruppo di sicurezza associato al database RDS ha una regola in entrata con l'intervallo di porte impostato sulla porta utilizzata dal database RDS.
- Il gruppo di sicurezza associato al database RDS ha una regola in entrata con l'origine impostata su un gruppo di sicurezza associato alla risorsa di calcolo.

Per visualizzare le risorse di calcolo connesse a un database RDS

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, scegli Databases (Database), quindi seleziona il nome del database RDS.
3. Nella scheda Connectivity & security (Connettività e sicurezza), visualizza le risorse di calcolo in Connected compute resources (Risorse di calcolo connesse).



Connessione a un'istanza database che esegue un motore DB specifico

Per informazioni sulla connessione a un'istanza database che esegue un motore DB specifico, seguire le istruzioni per il motore DB:

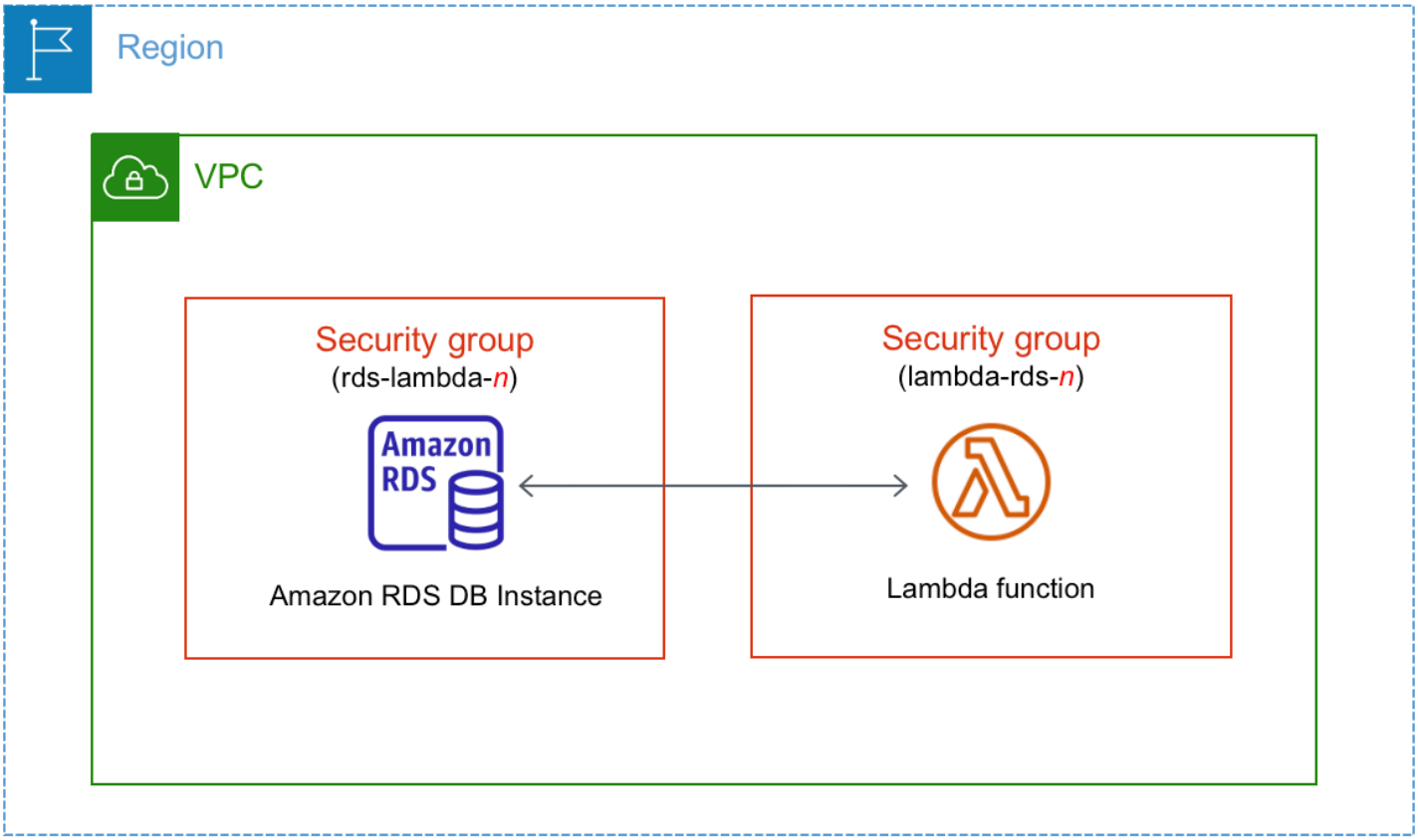
- Connessione a un'istanza database che esegue il motore di database MariaDB
- Connessione a un'istanza database che esegua il motore di database di Microsoft SQL Server
- Connessione a un'istanza database che esegue il motore di database di MySQL
- Connessione all'istanza database RDS per Oracle
- Connessione a un'istanza database che esegua il motore di database di PostgreSQL

Connessione automatica di una funzione Lambda e di un'istanza database

È possibile utilizzare la console Amazon RDS per semplificare la configurazione di una connessione tra una funzione Lambda e un'istanza database. Spesso, l'istanza database si trova in una sottorete privata all'interno di un VPC. La funzione Lambda può essere utilizzata dalle applicazioni per accedere all'istanza database privata.

Per istruzioni sulla configurazione di una connessione tra una funzione Lambda e un cluster database Multi-AZ, consulta the section called “Connessione di una funzione Lambda e di un cluster database Multi-AZ”.

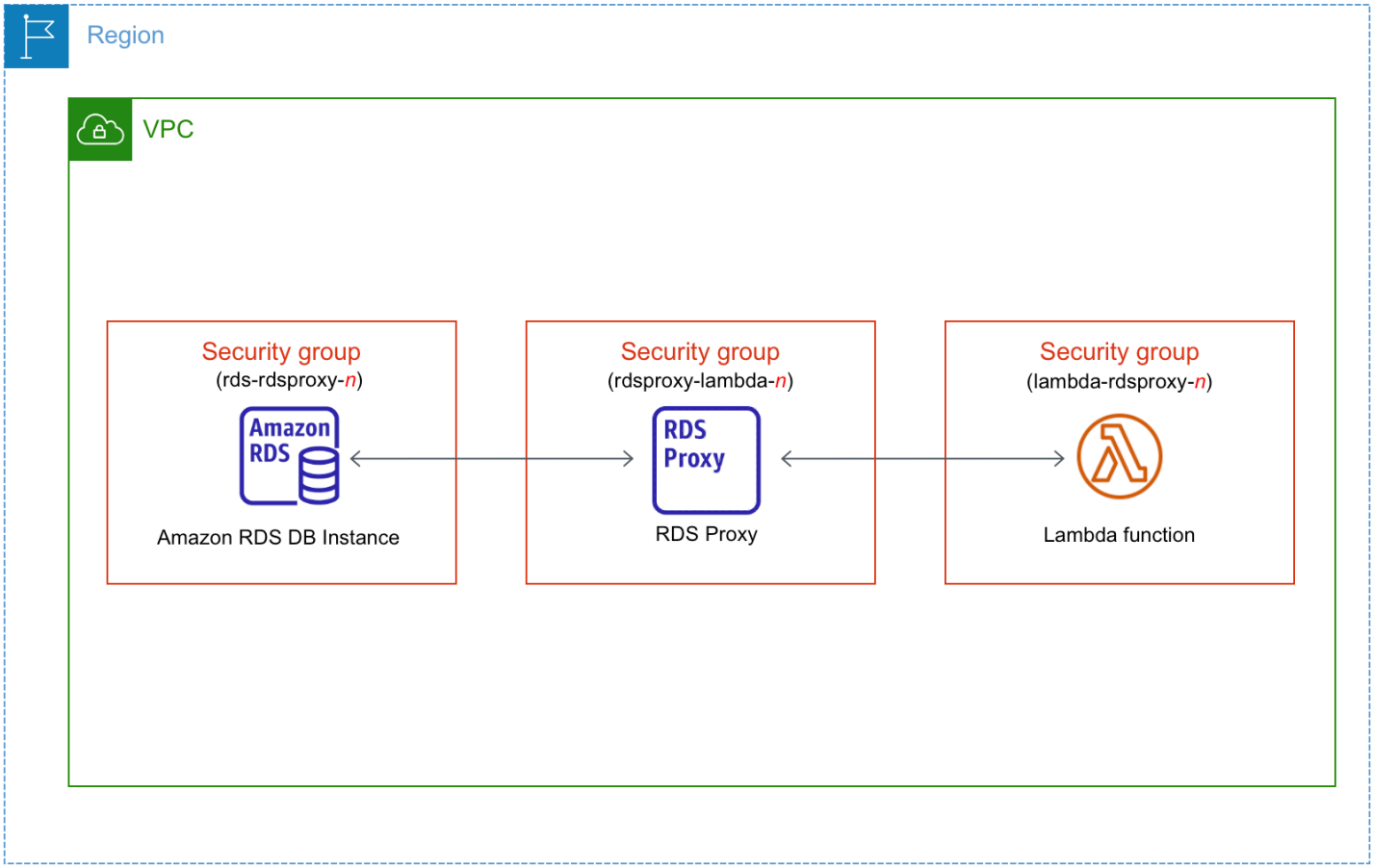
L'immagine seguente mostra una connessione diretta tra l'istanza database e la funzione Lambda.



È possibile configurare la connessione tra la funzione Lambda e l'istanza database tramite RDS Proxy per migliorare le prestazioni e la resilienza del database. Spesso, le funzioni Lambda effettuano connessioni database brevi e frequenti che traggono vantaggio dal pool di connessioni offerto da RDS Proxy. È possibile sfruttare qualsiasi autenticazione AWS Identity and Access Management (IAM) già disponibile per le funzioni Lambda, anziché gestire le credenziali del database nel codice dell'applicazione Lambda. Per ulteriori informazioni, consulta Utilizzo di Server proxy per Amazon RDS.

Quando si utilizza la console per connettersi a un proxy esistente, Amazon RDS aggiorna il gruppo di sicurezza proxy per consentire le connessioni dall'istanza database e la funzione Lambda.

È anche possibile creare un nuovo proxy dalla stessa pagina della console. Quando si crea un proxy nella console, per accedere all'istanza database, occorre inserire le credenziali del database o selezionare un segreto AWS Secrets Manager.



Argomenti

- Panoramica della connettività automatica a una funzione Lambda
- Connessione automatica di un funzione Lambda e di un database RDS
- Visualizzazione delle risorse di calcolo connesse

Panoramica della connettività automatica a una funzione Lambda

Di seguito sono riportati i requisiti per connettere una funzione Lambda a un'istanza database RDS:

- La funzione Lambda deve esistere nello stesso VPC dell'istanza database.
- L'utente che configura la connettività deve disporre delle autorizzazioni per eseguire le seguenti operazioni Amazon RDS, Amazon EC2, Lambda, Secrets Manager e IAM:
  - Amazon RDS
    - `rds:CreateDBProxies`
    - `rds:DescribeDBInstances`

- `rds:DescribeDBProxies`
- `rds:ModifyDBInstance`
- `rds:ModifyDBProxy`
- `rds:RegisterProxyTargets`

- Amazon EC2
  - `ec2:AuthorizeSecurityGroupEgress`
  - `ec2:AuthorizeSecurityGroupIngress`
  - `ec2:CreateSecurityGroup`
  - `ec2:DeleteSecurityGroup`
  - `ec2:DescribeSecurityGroups`
  - `ec2:RevokeSecurityGroupEgress`
  - `ec2:RevokeSecurityGroupIngress`
- Lambda
  - `lambda:CreateFunctions`
  - `lambda:ListFunctions`
  - `lambda:UpdateFunctionConfiguration`
- Secrets Manager
  - `secretsmanager:CreateSecret`
  - `secretsmanager:DescribeSecret`
- IAM
  - `iam:AttachPolicy`
  - `iam:CreateRole`
  - `iam:CreatePolicy`
- AWS KMS
  - `kms:describeKey`

> **Note**
>
> Se l'istanza database e la funzione Lambda si trovano in zone di disponibilità diverse, potrebbero essere addebitati costi tra zone di disponibilità all'account.

Quando si configura una connessione tra una funzione Lambda e un database RDS, Amazon RDS configura il gruppo di sicurezza VPC per la funzione e per l'istanza database. Se si utilizza il proxy RDS, Amazon RDS configura anche il gruppo di sicurezza VPC per il proxy. Amazon RDS opera in base alla configurazione corrente dei gruppi di sicurezza associati all'istanza database, alla funzione Lambda e al proxy, come descritto nella tabella seguente.

Configurazione del gruppo di sicurezza RDS corrente	Configurazione del gruppo di sicurezza Lambda corrente	Configurazione del gruppo di sicurezza proxy corrente	Operazione RDS
Esistono uno o più gruppi di sicurezza associati all'istanza database con un nome che corrisponde al modello `rds-lambda-` *n* o se un proxy è già connesso all'istanza database, RDS verifica se il parametro `TargetHealth` di un proxy associato è `AVAILABLE`. Un gruppo di sicurezza che corrisponde al modello non è stato modificato. Questo gruppo di sicurezza include una sola regola in entrata con il gruppo di sicurezza VPC della funzione Lambda o del proxy come l'origine.	Esistono uno o più gruppi di sicurezza associati alla funzione Lambda con un nome che corrisponde al modello `lambda-rds-` *n* o `lambda-rdsproxy-` *n* (dove *n* è un numero). Un gruppo di sicurezza che corrisponde al modello non è stato modificato. Questo gruppo di sicurezza dispone di una sola regola in uscita con il gruppo di sicurezza VPC dell'istanza database o il proxy come la destinazione.	Esistono uno o più gruppi di sicurezza associati al proxy con un nome che corrisponde al modello `rdsproxy-lambda-` *n* (dove *n* è un numero). Un gruppo di sicurezza che corrisponde al modello non è stato modificato. Questo gruppo di sicurezza dispone di regole in entrata e in uscita con i gruppi di sicurezza VPC della funzione Lambda e l'istanza database.	Amazon RDS non esegue alcuna azione. Una connessione era già stata configurata automaticamente tra la funzione Lambda, il proxy (opzionale) e l'istanza database. Poiché esiste già una connessione tra la funzione, il proxy e il database, i gruppi di sicurezza non vengono modificati.

Configurazione del gruppo di sicurezza RDS corrente	Configurazione del gruppo di sicurezza Lambda corrente	Configurazione del gruppo di sicurezza proxy corrente	Operazione RDS
Si applica una delle due condizioni seguenti: • Non esiste alcun gruppo di sicurezza associato all'istanza database con un nome che corrisponde al modello `rds-lambda-`*n* o se `TargetHealth` di un proxy associato è `AVAILABLE`. • Esiste almeno un gruppo di sicurezza associato all'istanza database con un nome che corrisponde al modello `rds-lambda-`*n* o se `TargetHealth` di un proxy associato è `AVAILABLE`. Tuttavia, nessuno di questi gruppi di sicurezza può essere utilizzato per la connessione alla funzione Lambda.	Si applica una delle due condizioni seguenti: • Non esiste un gruppo di sicurezza associato alla funzione Lambda con un nome che corrisponde al modello `lambda-rds-`*n* o `lambda-rdsproxy-`*n*. • Esistono uno o più gruppi di sicurezza associati alla funzione Lambda con un nome che corrisponde al modello `lambda-rds-`*n* o `lambda-rdsproxy-`*n*. Tuttavia, Amazon RDS non può utilizzare nessuno di questi gruppi di sicurezza per la connessione all'istanza database.	Si applica una delle due condizioni seguenti: • Non esiste un gruppo di sicurezza associato al proxy con un nome che corrisponde al modello `rdsproxy-lambda-`*n*. • Esistono uno o più gruppi di sicurezza associati al proxy con un nome che corrisponde al modello `rdsproxy-lambda-`*n*. Tuttavia, Amazon RDS non può utilizzare nessuno di questi gruppi di sicurezza per la connessione all'istanza database o alla funzione Lambda. Amazon RDS non può utilizzare un	RDS action: create new security groups

Configurazione del gruppo di sicurezza RDS corrente	Configurazione del gruppo di sicurezza Lambda corrente	Configurazione del gruppo di sicurezza proxy corrente	Operazione RDS
Amazon RDS non può utilizzare un gruppo di sicurezza che non dispone di una regola in entrata con il gruppo di sicurezza VPC della funzione Lambda o del proxy come l'origine. Inoltre, Amazon RDS non può utilizzare un gruppo di sicurezza che è stato modificato. Esempi di modifiche sono l'aggiunta di una regola o la modifica della porta di una regola esistente.	Amazon RDS non può utilizzare un gruppo di sicurezza che non dispone di una regola in uscita con il gruppo di sicurezza VPC dell'istanza database o del proxy come destinazione. Inoltre, Amazon RDS non può utilizzare un gruppo di sicurezza che è stato modificat o.	gruppo di sicurezza che non dispone di una regola in entrata e in uscita con il gruppo di sicurezza VPC dell'istanza database e la funzione Lambda. Inoltre, Amazon RDS non può utilizzare un gruppo di sicurezza che è stato modificat o.	

Configurazione del gruppo di sicurezza RDS corrente	Configurazione del gruppo di sicurezza Lambda corrente	Configurazione del gruppo di sicurezza proxy corrente	Operazione RDS
Esiste almeno un gruppo di sicurezza associato all'istanza database con un nome che corrisponde al modello `rds-lambda-` *n* o se `TargetHealth` di un proxy associato è `AVAILABLE` . Un gruppo di sicurezza che corrisponde al modello non è stato modificato. Questo gruppo di sicurezza include una sola regola in entrata con il gruppo di sicurezza VPC della funzione Lambda o del proxy come l'origine.	Esistono uno o più gruppi di sicurezza associati alla funzione Lambda con un nome che corrisponde al modello `lambda-rds-` *n* o `lambda-rdsproxy-` *n*. Tuttavia, Amazon RDS non può utilizzare nessuno di questi gruppi di sicurezza per la connessione all'istanza database. Amazon RDS non può utilizzare un gruppo di sicurezza che non dispone di una regola in uscita con il gruppo di sicurezza VPC dell'istanza database o del proxy come destinazione. Inoltre, Amazon RDS non può utilizzare un gruppo di sicurezza che è stato modificato.	Esistono uno o più gruppi di sicurezza associati al proxy con un nome che corrisponde al modello `rdsproxy-lambda-` *n*. Tuttavia, Amazon RDS non può utilizzare nessuno di questi gruppi di sicurezza per la connessione all'istanza database o alla funzione Lambda. Amazon RDS non può utilizzare un gruppo di sicurezza che non dispone di una regola in entrata e in uscita con il gruppo di sicurezza VPC dell'istanza database e la funzione Lambda. Inoltre, Amazon RDS non può utilizzare un gruppo di sicurezza che è stato modificato.	RDS action: create new security groups

Configurazione del gruppo di sicurezza RDS corrente	Configurazione del gruppo di sicurezza Lambda corrente	Configurazione del gruppo di sicurezza proxy corrente	Operazione RDS
Esiste almeno un gruppo di sicurezza associato all'istanza database con un nome che corrisponde al modello `rds-lambda-` *n* o se `TargetHealth` di un proxy associato è `AVAILABLE`. Un gruppo di sicurezza che corrisponde al modello non è stato modificato. Questo gruppo di sicurezza include una sola regola in entrata con il gruppo di sicurezza VPC della funzione Lambda o del proxy come l'origine.	Esiste un gruppo di sicurezza Lambda valido per la connessione, ma non è associato alla funzione Lambda. Questo gruppo di sicurezza ha un nome che corrisponde al modello `lambda-rds-` *n* o `lambda-rdsproxy-` *n*. Non è stato modificato. Dispone di una sola regola in uscita con il gruppo di sicurezza VPC dell'istanza database o del proxy come la destinazione.	Esiste un gruppo di sicurezza proxy valido per la connessione, ma non è associato al proxy. Questo gruppo di sicurezza ha un nome che corrisponde al modello `rdsproxy-lambda-` *n*. Non è stato modificato. Dispone di regole in entrata e in uscita con i gruppi di sicurezza VPC dell'istanza database e la funzione Lambda.	RDS action: associate Lambda security group

Configurazione del gruppo di sicurezza RDS corrente	Configurazione del gruppo di sicurezza Lambda corrente	Configurazione del gruppo di sicurezza proxy corrente	Operazione RDS
Si applica una delle due condizioni seguenti: • Non esiste alcun gruppo di sicurezza associato all'istanza database con un nome che corrisponde al modello `rds-lambda-` *n* o se `TargetHealth` di un proxy associato è `AVAILABLE` . • Esiste almeno un gruppo di sicurezza associato all'istanza database con un nome che corrisponde al modello `rds-lambda-` *n* o se `TargetHealth` di un proxy associato è `AVAILABLE` . Tuttavia, Amazon RDS non può utilizzare nessuno di questi gruppi di sicurezza per la connessione alla	Esistono uno o più gruppi di sicurezza associati alla funzione Lambda con un nome che corrisponde al modello `lambda-rds-` *n* o `lambda-rdsproxy-` *n*. Un gruppo di sicurezza che corrisponde al modello non è stato modificato. Questo gruppo di sicurezza dispone di una sola regola in uscita con il gruppo di sicurezza VPC dell'istanza database o del proxy come la destinazione.	Esistono uno o più gruppi di sicurezza associati al proxy con un nome che corrisponde al modello `rdsproxy-lambda-` *n*. Un gruppo di sicurezza che corrisponde al modello non è stato modificato. Questo gruppo di sicurezza dispone di regole in entrata e in uscita con il gruppo di sicurezza VPC dell'istanza database e la funzione Lambda.	RDS action: create new security groups

Configurazione del gruppo di sicurezza RDS corrente	Configurazione del gruppo di sicurezza Lambda corrente	Configurazione del gruppo di sicurezza proxy corrente	Operazione RDS
funzione Lambda o al proxy. Amazon RDS non può utilizzare un gruppo di sicurezza che non dispone di una regola in entrata con il gruppo di sicurezza VPC della funzione Lambda o del proxy come l'origine. Inoltre, Amazon RDS non può utilizzare un gruppo di sicurezza che è stato modificato.			

Configurazione del gruppo di sicurezza RDS corrente	Configurazione del gruppo di sicurezza Lambda corrente	Configurazione del gruppo di sicurezza proxy corrente	Operazione RDS
Si applica una delle due condizioni seguenti: • Non esiste alcun gruppo di sicurezza associato all'istanza database con un nome che corrisponde al modello `rds-lambda-`*n* o se `TargetHealth` di un proxy associato è `AVAILABLE`. • Esiste almeno un gruppo di sicurezza associato all'istanza database con un nome che corrisponde al modello `rds-lambda-`*n* o se `TargetHealth` di un proxy associato è `AVAILABLE`. Tuttavia, Amazon RDS non può utilizzare nessuno di questi gruppi di sicurezza per la connessione alla	Si applica una delle due condizioni seguenti: • Non esiste un gruppo di sicurezza associato alla funzione Lambda con un nome che corrisponde al modello `lambda-rds-`*n* o `lambda-rdsproxy-`*n*. • Esistono uno o più gruppi di sicurezza associati alla funzione Lambda con un nome che corrisponde al modello `lambda-rds-`*n* o `lambda-rdsproxy-`*n*. Tuttavia, Amazon RDS non può utilizzare nessuno di questi gruppi di sicurezza per la connessione all'istanza database.	Si applica una delle due condizioni seguenti: • Non esiste un gruppo di sicurezza associato al proxy con un nome che corrisponde al modello `rdsproxy-lambda-`*n*. • Esistono uno o più gruppi di sicurezza associati al proxy con un nome che corrisponde al modello `rdsproxy-lambda-`*n*. Tuttavia, Amazon RDS non può utilizzare nessuno di questi gruppi di sicurezza per la connessione all'istanza database o alla funzione Lambda. Amazon RDS non può utilizzare un	RDS action: create new security groups

Configurazione del gruppo di sicurezza RDS corrente	Configurazione del gruppo di sicurezza Lambda corrente	Configurazione del gruppo di sicurezza proxy corrente	Operazione RDS
funzione Lambda o al proxy. Amazon RDS non può utilizzare un gruppo di sicurezza che non dispone di una regola in entrata con il gruppo di sicurezza VPC della funzione Lambda o del proxy come l'origine. Inoltre, Amazon RDS non può utilizzare un gruppo di sicurezza che è stato modificato.	Amazon RDS non può utilizzare un gruppo di sicurezza che non dispone di una regola in uscita con il gruppo di sicurezza VPC dell'istanza database o del proxy come l'origine. Inoltre, Amazon RDS non può utilizzare un gruppo di sicurezza che è stato modificato.	gruppo di sicurezza che non dispone di una regola in entrata e in uscita con il gruppo di sicurezza VPC dell'istanza database e la funzione Lambda. Inoltre, Amazon RDS non può utilizzare un gruppo di sicurezza che è stato modificato.	

RDS: creazione di nuovi gruppi di sicurezza

Amazon RDS esegue le seguenti operazioni:

- Crea un nuovo gruppo di sicurezza che corrisponde al modello `rds-lambda-`*n* o `rds-rdsproxy-`*n* (se si sceglie di utilizzare RDS Proxy). Questo gruppo di sicurezza dispone di una regola in entrata con il gruppo di sicurezza VPC della funzione Lambda o proxy come l'origine. Questo gruppo di sicurezza è associato all'istanza database e consente alla funzione o al proxy di accedere all'istanza database.
- Crea un nuovo gruppo di sicurezza che corrisponde al modello `lambda-rds-`*n* o `lambda-rdsproxy-`*n*. Questo gruppo di sicurezza dispone di una regola in uscita con il gruppo di sicurezza VPC dell'istanza database o del proxy come la destinazione. Questo gruppo di sicurezza è associato alla funzione Lambda e consente alla funzione di inviare traffico all'istanza database o inviare traffico tramite un proxy.

- Crea un nuovo gruppo di sicurezza che corrisponde al modello `rdsproxy-lambda-`*n*. Questo gruppo di sicurezza dispone di regole in entrata e in uscita con il gruppo di sicurezza VPC dell'istanza database e la funzione Lambda.

Azione RDS: associazione del gruppo di sicurezza Lambda

Amazon RDS associa il gruppo di sicurezza Lambda valido, esistente alla funzione Lambda. Questo gruppo di sicurezza consente alla funzione di inviare traffico all'istanza database o inviare traffico tramite un proxy.

Connessione automatica di un funzione Lambda e di un database RDS

È possibile usare la console Amazon RDS per connettere automaticamente una funzione Lambda all'istanza database. Ciò semplifica il processo di configurazione di una connessione tra queste risorse.

È anche possibile usare RDS Proxy per includere un proxy nella connessione. Funzioni Lambda effettuano connessioni database brevi e frequenti che traggono vantaggio dal pool di connessioni offerto da RDS Proxy. È anche possibile utilizzare qualsiasi autenticazione IAM che è già stata configurata per le funzioni Lambda, anziché gestire le credenziali del database nel codice dell'applicazione Lambda.

È possibile connettere un'istanza database esistente a funzioni Lambda nuove ed esistenti utilizzando la pagina Configurazione della connessione Lambda. Il processo di configurazione consente di impostare automaticamente i gruppi di sicurezza richiesti.

Prima di configurare una connessione tra una funzione Lambda e un'istanza database, assicurati che:

- La funzione Lambda e l'istanza database si trovino nello stesso VPC.
- Disponi delle autorizzazioni corrette per l'account utente. Per ulteriori informazioni sui requisiti, consulta Panoramica della connettività automatica a una funzione Lambda.

Se si modificano i gruppi di sicurezza dopo la configurazione della connettività, le modifiche potrebbero influenzare la connessione tra la funzione Lambda e l'istanza database.

> **Note**
>
> È possibile configurare automaticamente una connessione tra un'istanza database e una funzione Lambda solo nella AWS Management Console. Per connettere una funzione Lambda, l'istanza database deve essere nello stato Disponibile.

Per connettere automaticamente una funzione Lambda e un'istanza database

<result>

Dopo aver confermato la configurazione, Amazon RDS avvia il processo di connessione della funzione Lambda, di RDS Proxy (se hai usato un proxy) e dell'istanza database. La console mostra la finestra di dialogo Dettagli di connessione, in cui sono elencate le modifiche al gruppo di sicurezza che consentono le connessioni tra le risorse.
</result>

1. Accedi alla AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, scegli Database, quindi seleziona l'istanza database che desideri connettere a una funzione Lambda.
3. Per Operazioni, scegli Configura connessione Lambda.
4. Nella pagina Configurazione della connessione Lambda, in Seleziona la funzione Lambda, effettua una delle seguenti operazioni:
   - Se disponi di una funzione Lambda esistente nello stesso VPC dell'istanza database, seleziona Scegli una funzione esistente, quindi seleziona la funzione.
   - Se non disponi di una funzione Lambda nello stesso VPC, seleziona Crea una nuova funzione, quindi inserisci un Nome della funzione. Il runtime predefinito è impostato su Nodejs.18. Puoi modificare le impostazioni per la nuova funzione Lambda nella console Lambda dopo aver completato la configurazione della connessione.
5. (Facoltativo) In RDS Proxy, seleziona Connessione tramite RDS Proxy, quindi esegui una delle seguenti operazioni:
   - Se disponi di un proxy esistente che desideri utilizzare, seleziona Scegli un proxy esistente, quindi seleziona il proxy.
   - Se non disponi di un proxy e desideri che uno venga creato automaticamente da Amazon RDS, seleziona Crea nuovo proxy. Quindi, per Credenziali del database, esegui una delle seguenti operazioni:

a. Seleziona Nome utente e password del database, quindi inserisci Nome utente e Password per l'istanza database.

b. Seleziona Segreti Secrets Manager. Quindi, per Seleziona segreto, scegli un segreto AWS Secrets Manager. Se non disponi di un segreto di Secrets Manager, seleziona Crea nuovo segreto di Secrets Manager per creare un nuovo segreto. Dopo aver creato il segreto, per Seleziona segreto, scegli il nuovo segreto.

Dopo aver creato il nuovo proxy, seleziona Scegli un proxy esistente, quindi seleziona il proxy. Tieni presente che prima che il proxy sia disponibile per la connessione, potrebbe trascorrere del tempo.

6. (Facoltativo) Espandi Riepilogo della connessione e verifica gli aggiornamenti evidenziati per le risorse.
7. Scegliere Set up (Configura).

Visualizzazione delle risorse di calcolo connesse

È possibile usare la AWS Management Console per visualizzare le funzioni Lambda collegate all'istanza database. Le risorse mostrate includono le connessioni delle risorse di calcolo configurate automaticamente da Amazon RDS.

Le risorse di elaborazione elencate non includono quelle connesse manualmente all'istanza database. Ad esempio, è possibile autorizzare manualmente una risorsa di calcolo ad accedere all'istanza database aggiungendo una regola al gruppo di sicurezza VPC associato al database.

Affinché la console elenchi una funzione Lambda, devono essere soddisfatte le seguenti condizioni:

- Il nome del gruppo di sicurezza associato alla risorsa di calcolo corrisponde al modello `lambda-rds-`*n* o `lambda-rdsproxy-`*n* (dove *n* è un numero).
- Il gruppo di sicurezza associato alla risorsa di calcolo dispone di una regola in uscita con l'intervallo di porte impostato sulla porta dell'istanza database o un proxy associato. La destinazione della regola in uscita deve essere impostata su un gruppo di sicurezza associato all'istanza database o a un proxy associato.
- Se la configurazione include un proxy, il nome del gruppo di sicurezza collegato al proxy associato al database corrisponde al modello `rdsproxy-lambda-`*n* (dove *n* è un numero).

- Il gruppo di sicurezza associato alla funzione dispone di una regola in uscita con l'intervallo di porte impostato sulla porta utilizzata dall'istanza database o da un proxy associato. La destinazione deve essere impostata su un gruppo di sicurezza associato all'istanza database o a un proxy associato.

Per visualizzare le risorse di elaborazione connesse automaticamente a un'istanza database

1. Accedi alla AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel pannello di navigazione, scegli Database e seleziona l'istanza database.
3. Nella scheda Connettività e sicurezza, visualizza le risorse di calcolo in Risorse di calcolo connesse.

Modifica di un'istanza database Amazon RDS

Puoi modificare le impostazioni di un'istanza database per eseguire attività come l'aggiunta di ulteriore storage o la modifica della classe dell'istanza database. In questo argomento viene descritto come modificare un'istanza database di Amazon RDS e vengono fornite informazioni sulle impostazioni per le istanze database.

È consigliabile testare eventuali modifiche in un'istanza di test prima di modificare un'istanza di produzione. In questo modo è possibile comprendere appieno l'impatto di ogni modifica. Il test è importante soprattutto in caso di aggiornamento delle versioni di database.

La maggior parte delle modifiche apportate a un'istanza database possono essere applicate immediatamente o ritardate fino alla successiva finestra di manutenzione. Alcune modifiche, ad esempio le modifiche al gruppo di parametri, richiedono il riavvio manuale dell'istanza database per rendere effettiva la modifica.

> **Important**
>
> Alcune modifiche portano a un'interruzione perché Amazon RDS deve riavviare l'istanza database per rendere effettiva la modifica. Esamina l'impatto sul database e le applicazioni prima di modificare le impostazioni della tua istanza di database.

Console

Per modificare un'istanza database

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, scegliere Databases (Database) e selezionare l'istanza database da modificare.
3. Scegliere Modify (Modifica). Viene visualizzata la pagina Modify DB Instance (Modifica istanza database).
4. Modificare le impostazioni desiderate. Per informazioni su ciascuna impostazione, consulta Impostazioni per istanze database.
5. Quando tutte le modifiche sono come le desideri, seleziona Continue (Continua) e controlla il riepilogo delle modifiche.

6. (Facoltativo) Scegliere Applica immediatamente per applicare immediatamente le modifiche. In alcuni casi, la chiusura di questa opzione può causare un'interruzione. Per ulteriori informazioni, consulta Impostazione delle modifiche alla pianificazione.
7. Nella pagina di conferma esaminare le modifiche. Se sono corrette, seleziona Modifica istanza database per salvare le modifiche.

   Oppure scegliere Back (Indietro) per cambiare le modifiche o Cancel (Annulla) per annullare le modifiche.

AWS CLI

Per modificare un'istanza DB utilizzando il AWS CLI, chiama il modify-db-instancecomando. Specifica l'identificatore istanze DB e i valori per le impostazioni da modificare. Per ulteriori informazioni su ciascuna opzione, consulta Impostazioni per istanze database.

Example

Il codice seguente modifica `mydbinstance` impostando il periodo di retention dei backup a 1 settimana (7 giorni). Il codice abilita la protezione da eliminazione utilizzando `--deletion-protection`. Per disabilitare la protezione da eliminazione, utilizza `--no-deletion-protection`. Le modifiche vengono applicate durante la prossima finestra di manutenzione utilizzando `--no-apply-immediately`. Utilizza `--apply-immediately` per applicare immediatamente le modifiche. Per ulteriori informazioni, consulta Impostazione delle modifiche alla pianificazione.

Per LinuxmacOS, oUnix:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --backup-retention-period 7 \
    --deletion-protection \
    --no-apply-immediately
```

Per Windows:

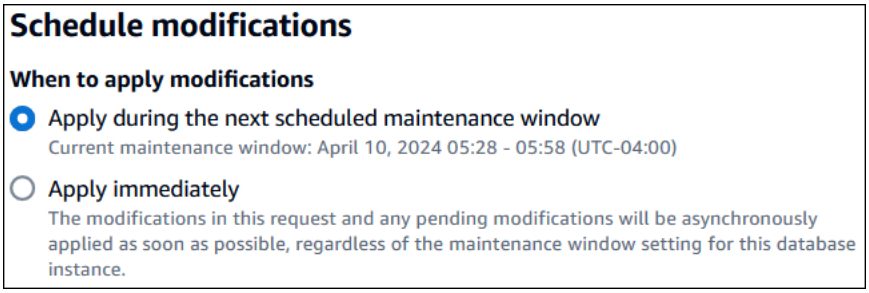
```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --backup-retention-period 7 ^
    --deletion-protection ^
    --no-apply-immediately
```

API RDS

Per modificare un'istanza database tramite l'API di Amazon RDS, chiama l'operazione ModifyDBInstance. Specifica l'identificatore istanze database e i parametri per le impostazioni da modificare. Per informazioni su ciascun parametro, consulta Impostazioni per istanze database.

Impostazione delle modifiche alla pianificazione

Quando modifichi l'istanza DB, sei tu a decidere quando apportare le modifiche.



Per applicare le modifiche immediatamente anziché nella finestra di manutenzione successiva, scegliete l'opzione Applica immediatamente in. AWS Management Console Oppure usi il `--apply-immediately` parametro quando chiami AWS CLI o imposta il `ApplyImmediately` parametro su `true` quando usi l'API Amazon RDS.

Se scegli di non applicare le modifiche immediatamente, RDS le inserisce nella coda delle modifiche in sospeso. Durante la finestra di manutenzione successiva, RDS applica tutte le modifiche in sospeso nella coda. Se scegli di applicare le modifiche immediatamente, verranno applicate le nuove modifiche e tutte le modifiche nella coda delle modifiche in sospeso.

Per vedere le modifiche in sospeso per la prossima finestra di manutenzione, usa il describe-db-instances AWS CLI comando e controlla il campo. `PendingModifiedValues`

> **Important**
>
> Se una qualsiasi delle modifiche in sospeso richiede che l'istanza database non sia temporaneamente disponibile (downtime), la scelta dell'opzione Applica immediatamente può causare tempi di inattività imprevisti.
> Se scegli di applicare subito una modifica, devi tener presente che saranno applicate immediatamente tutte le modifiche, invece che durante la prossima finestra di manutenzione.

Se non vuoi che una modifica in sospeso venga applicata nella prossima finestra di manutenzione, puoi modificare l'istanza database per annullare la modifica. È possibile farlo utilizzando AWS CLI e specificando l'opzione. `--apply-immediately`

Le modifiche ad alcune impostazioni di database vengono applicate immediatamente, anche se scegli di rinviarle. Per vedere come le diverse impostazioni del database interagiscono con l'impostazione Applica immediatamente, consulta Impostazioni per istanze database.

Impostazioni per istanze database

Nella tabella seguente sono disponibili i dettagli relativi alle impostazioni che puoi e non puoi modificare. Puoi anche capire quando è possibile applicare le modifiche e se le modifiche causano tempi di inattività per l'istanza database. Utilizzando le funzionalità di Amazon RDS come Multi-AZ, puoi ridurre al minimo i tempi di inattività se successivamente modifichi l'istanza database. Per ulteriori informazioni, consulta Configurazione e gestione di un'implementazione multi-AZ.

Puoi modificare un'istanza database utilizzando la console, il comando `modify-db-instance` della CLI o l'operazione `ModifyDBInstance` dell'API RDS.

Impostazione e descrizione della console	Opzione CLI e parametro API di RDS	Quando avvengono le modifiche	Note sui tempi di inattività	Motori di database supportati
Allocated storage (Storage allocato) Storage, in gibibyte, da allocare all'istanza database. Lo storage allocato può essere solo aumento, non ridotto. Non è possibile modificare lo storage di alcune istanze database precedenti o delle istanze database ripristinate	Opzione CLI: `--allocated-storage` Parametro API RDS: `AllocatedStorage`	Se si sceglie di applicare la modifica immediatamente, questa si verifica immediatamente. Se non si sceglie di applicare	Durante questa modifica non si verifica un'interruzione. Le prestazioni potrebbero essere inferiori durante la modifica.	Tutti i motori di database

Impostazione e descrizione della console	Opzione CLI e parametro API di RDS	Quando avvengono le modifiche	Note sui tempi di inattività	Motori di database supportati
da snapshot DB precedenti. L'impostazione Allocated storage (Storage allocato) è disabilitata nella console se l'istanza database non è idonea. È possibile verificare se è possibile allocare più spazio di archiviazione utilizzando il comando CLI describe-valid-db-instance-modifications. Questo comando restituisce le opzioni storage valide per l'istanza database. Non puoi modificare lo spazio di archiviazione allocato se lo stato dell'istanza database è storage-optimization (ottimizzazione-archiviazione). Inoltre, non puoi modificare lo spazio di archiviazione allocato per un'istanza database se è già stato modificato nelle ultime sei ore. Lo storage massimo consentito dipende dal motore di database e dal tipo di storage. Per ulteriori informazioni, consulta Storage delle istanze di database Amazon RDS.		la modifica immediatamente, questa si verifica durante la finestra di manutenzione successiva.		

Impostazione e descrizione della console	Opzione CLI e parametro API di RDS	Quando avvengono le modifiche	Note sui tempi di inattività	Motori di database supportati
Configurazione dell'architettura Una configurazione che consente a più database del tenant di risiedere nell'istanza database. Attualmente, solo i database container (CDB) RDS per Oracle supportano questa impostazione. Se il CDB è nella configurazione a tenant singolo, puoi modificarlo per utilizzare la configurazione multi-tenant. In questa configurazione, è possibile utilizzare le API RDS per creare da 1 a 30 database tenant, a seconda dell'edizione del database e delle eventuali licenze opzionali richieste. I PDB di applicazioni e i PDB di proxy non sono supportati. La configurazione multi-tenant è permanente, il che significa che non è possibile riconvertire successivamente il CDB nella configurazione a tenant singolo. **Note** La funzionalità Amazon RDS è chiamata "multi-tenant" anziché "multiten	Opzione CLI: `--multi-tenant` (configurazione multi-tenant dell'architettura CDB) `--no-multi-tenant` (configurazione a tenant singolo dell'architettura CDB) Parametro API: `MultiTenant`	La modifica avviene immediatamente.	Durante questa modifica non si verifica un'interruzione.	Oracle

Impostazione e descrizione della console	Opzione CLI e parametro API di RDS	Quando avvengono le modifiche	Note sui tempi di inattività	Motori di database supportati
ant" perché è una funzionalità della piattaforma RDS, non solo del motore di database Oracle. Il termine "Oracle multitenant" si riferisce esclusivamente all'architettura del database Oracle, che è compatibile sia con le implementazioni on-premise che con quelle RDS. Per ulteriori informazioni, consulta Panoramica dei database CDB RDS per Oracle.				

Impostazione e descrizione della console	Opzione CLI e parametro API di RDS	Quando avvengono le modifiche	Note sui tempi di inattività	Motori di database supportati
Impostazioni dell'architettura L'architettura del database Oracle: CDB o non CDB. Se si sceglie l'architettura multi-tenant Oracle, RDS per Oracle converte il database non CDB in un database CDB che utilizza la configurazione a tenant singolo. Questa impostazione è supportata solo se il database è un database non CDB che esegue Oracle Database 9c con RU di aprile 2021 o versioni successive. Dopo la conversione, il CDB contiene un database collegabile (PDB) iniziale. La modifica dell'architettura è permanente, il che significa che non puoi riconvertire il tuo CDB in un database non CDB. **Note** Per convertire un CDB nella configurazione a tenant singolo in una configurazione multi-tenant, modifica nuovamente l'istanza CDB e scegli	Opzione CLI: `--engine oracle-ee-cdb` (multitenant Oracle) `--engine oracle-se2-cdb` (multitenant Oracle) Parametro API: `Engine`	Se si sceglie di applicare la modifica immediatamente, questa si verifica immediatamente. Se non si sceglie di applicare la modifica immediatamente, questa si verifica durante la finestra di manutenzione successiva.	Durante questa modifica si verifica un'interruzione.	Oracle

Impostazione e descrizione della console	Opzione CLI e parametro API di RDS	Quando avvengono le modifiche	Note sui tempi di inattività	Motori di database supportati
Configurazione multi-tenant per Configurazione dell'architettura. Per ulteriori informazioni, consulta Configurazione a tenant singolo dell'architettura CDB.				

Impostazione e descrizione della console	Opzione CLI e parametro API di RDS	Quando avvengono le modifiche	Note sui tempi di inattività	Motori di database supportati
Auto minor version upgrade (Aggiornamento automatico della versione secondaria) Scegli Abilita aggiornamento automatico della versione secondaria per consentire all'istanza DB di ricevere automaticamente gli aggiornamenti delle versioni secondarie preferite del motore DB non appena diventano disponibili. Questo è il comportamento che segue di default. Amazon RDS esegue aggiornamenti automatici di versioni secondarie nella finestra di manutenzione. Se non scegli Abilita l'aggiornamento automatico della versione secondaria, l'istanza DB non viene aggiornata automaticamente quando diventano disponibili nuove versioni secondarie. Per ulteriori informazioni, consulta Aggiornamento automatico della versione secondaria del motore.	Opzione CLI: `--auto-minor-version-upgrade | --no-auto-minor-version-upgrade` Parametro API RDS: `AutoMinorVersionUpgrade`	La modifica avviene immediatamente. Questa impostazione ignora l'impostazione Applica immediatamente.	Durante questa modifica non si verifica un'interruzione.	Tutti i motori di database

Impostazione e descrizione della console	Opzione CLI e parametro API di RDS	Quando avvengono le modifiche	Note sui tempi di inattività	Motori di database supportati
Replica di backup Scegli Abilita la replica in un'altra AWS regione per creare backup in un'altra regione per il disaster recovery. Seleziona quindi la regione di destinazione per i backup aggiuntivi.	Non disponibile durante la modifica di un'istanza database. Per informazioni sull'abilitazione dei backup tra regioni utilizzando l'API AWS CLI o RDS, consulta. Abilitazione dei backup automatici tra regioni	La modifica viene applicata in modo asincrono, appena possibile.	Durante questa modifica non si verifica un'interruzione.	Oracle, PostgreSQL, SQL Server

Impostazione e descrizione della console	Opzione CLI e parametro API di RDS	Quando avvengono le modifiche	Note sui tempi di inattività	Motori di database supportati
Backup retention period (Periodo di retention dei backup) Numero di giorni durante i quali vengono conservati i backup automatici. Per disabilitare i backup automatici, imposta il periodo di retention dei backup su 0. Per ulteriori informazioni, consulta Introduzione ai backup. **Note** Se lo utilizzi AWS Backup per gestire i tuoi backup, questa opzione non è applicabile. Per informazioni in merito AWS Backup, consulta la AWS Backup Developer Guide.	Opzione CLI: `--backup-retention-period` Parametro API RDS: `BackupRetentionPeriod`	Se si sceglie di applicare la modifica immediatamente, questa si verifica immediatamente. Se non si sceglie di applicare la modifica immediatamente ed è possibile modificare l'impostazione da un valore diverso da zero a un altro valore diverso da zero, la modifica viene applicata in modo asincrono, appena possibile. In caso contrario, la modifica	Si verifica un'interruzione se cambi da 0 a un valore diverso da zero o da un valore diverso da zero a 0. Questo vale sia per le istanze database a singola zona di disponibilità che Multi-AZ.	Tutti i motori di database

Impostazione e descrizione della console	Opzione CLI e parametro API di RDS	Quando avvengono le modifiche	Note sui tempi di inattività	Motori di database supportati
		avviene durante la finestra di manutenzione successiva.		
Backup window (Finestra di backup) Intervallo di tempo in cui vengono eseguiti i backup automatici del database. L'ora di inizio della finestra di backup è indicata in base al formato Universal Coordinated Time (UTC) e la sua durata è in ore. Per ulteriori informazioni, consulta Introduzione ai backup. **Note** Se lo utilizzi AWS Backup per gestire i tuoi backup, questa opzione non viene visualizzata. Per informazioni in merito AWS Backup, consulta la Guida per gli AWS Backup sviluppatori.	Opzione CLI: `--preferred-backup-window` Parametro API RDS: `PreferredBackupWindow`	La modifica viene applicata in modo asincrono, appena possibile.	Durante questa modifica non si verifica un'interruzione.	Tutti i motori di database

Impostazione e descrizione della console	Opzione CLI e parametro API di RDS	Quando avvengono le modifiche	Note sui tempi di inattività	Motori di database supportati
Autorità di certificazione L'autorità di certificazione (CA) per il certificato del server utilizzato dall'istanza database. Per ulteriori informazioni, consulta .	Opzione CLI: `--ca-certificate-identifier` Parametro API RDS: `CACertificateIdentifier`	Se si sceglie di applicare la modifica immediatamente, questa si verifica immediatamente. Se non si sceglie di applicare la modifica immediatamente, questa si verifica durante la finestra di manutenzione successiva.	I tempi di interruzione si verificano solo se il motore database non supporta la rotazione senza riavvio. È possibile utilizzare il describe-db-engine-versions AWS CLI comando per determinare se il motore DB supporta la rotazione senza riavvio.	Tutti i motori di database

Impostazione e descrizione della console	Opzione CLI e parametro API di RDS	Quando avvengono le modifiche	Note sui tempi di inattività	Motori di database supportati
Copy tags to snapshots (Copia tag in snapshot) Se sono presenti tag dell'istanza database, abilitare questa opzione per copiarli al momento della creazione di uno snapshot DB. Per ulteriori informazioni, consulta Tagging delle risorse Amazon RDS.	Opzione CLI: `--copy-tags-to-snapshot` o `--no-copy-tags-to-snapshot` Parametro API RDS: `CopyTagsToSnapshot`	La modifica avviene immediatamente. Questa impostazione ignora l'impostazione Applica immediatamente.	Durante questa modifica non si verifica un'interruzione.	Tutti i motori di database
Database port (Porta del database) La porta da utilizzare per accedere all'istanza database. Il valore della porta non deve corrispondere ad altri valori di porta specificati per le opzioni nel gruppo di opzioni associato all'istanza database. Per ulteriori informazioni, consulta Connessione a un'istanza database Amazon RDS.	Opzione CLI: `--db-port-number` Parametro API RDS: `DBPortNumber`	La modifica avviene immediatamente. Questa impostazione ignora l'impostazione Applica immediatamente.	L'istanza database viene riavviata immediatamente.	Tutti i motori di database

Impostazione e descrizione della console	Opzione CLI e parametro API di RDS	Quando avvengono le modifiche	Note sui tempi di inattività	Motori di database supportati
DB engine version (Versione motore del database La versione del motore di database da utilizzare. Prima di eseguire l'aggiornamento delle istanze database di produzione, è consigliabile testare il processo di aggiornamento in un'istanza database di test per verificarne la durata e convalidare le applicazioni. Per ulteriori informazioni, consulta Aggiornamento della versione del motore di un'istanza database.	Opzione CLI: `--engine-version` Parametro API RDS: `EngineVersion`	Se si sceglie di applicare la modifica immediatamente, questa si verifica immediatamente. Se non si sceglie di applicare la modifica immediatamente, questa si verifica durante la finestra di manutenzione successiva.	Durante questa modifica si verifica un'interruzione.	Tutti i motori di database

Impostazione e descrizione della console	Opzione CLI e parametro API di RDS	Quando avvengono le modifiche	Note sui tempi di inattività	Motori di database supportati
DB instance class (Classe istanza database) Classe dell'istanza database da utilizzare. Per ulteriori informazioni, consulta Classi di istanze database.	Opzione CLI: `--db-instance-class` Parametro API RDS: `DBInstanceClass`	Se si sceglie di applicare la modifica immediatamente, questa si verifica immediatamente. Se non si sceglie di applicare la modifica immediatamente, questa si verifica durante la finestra di manutenzione successiva.	Durante questa modifica si verifica un'interruzione.	Tutti i motori di database

Impostazione e descrizione della console	Opzione CLI e parametro API di RDS	Quando avvengono le modifiche	Note sui tempi di inattività	Motori di database supportati
DB instance identifier (Identificatore istanze DB) Nuovo identificatore istanze DB. Questo valore è archiviato come stringa in caratteri minuscoli. Per ulteriori informazioni sugli effetti della modifica del nome di un'istanza database, consulta Ridenominazione di un'istanza database.	Opzione CLI: `--new-db-instance-identifier` Parametro API RDS: `NewDBInstanceIdentifier`	Se si sceglie di applicare la modifica immediatamente, questa si verifica immediatamente. Se non si sceglie di applicare la modifica immediatamente, questa si verifica durante la finestra di manutenzione successiva.	Durante questa modifica si verificano tempi di inattività, a meno che la versione del motore database non supporti il caricamento SSL dinamico. Per determinare se la tua versione richiede il riavvio, esegui il AWS CLI comando seguente: `aws rds describe-db-engine-versions \ --default-only \ --engine your-db-engine \ --query 'DBEngineVersions[`	Tutti i motori di database

Impostazione e descrizione della console	Opzione CLI e parametro API di RDS	Quando avvengono le modifiche	Note sui tempi di inattività	Motori di database supportati
			`*].Supports CertificateRotationWithoutRestart'`	

Impostazione e descrizione della console	Opzione CLI e parametro API di RDS	Quando avvengono le modifiche	Note sui tempi di inattività	Motori di database supportati
DB parameter group (Gruppo di parametri database) Gruppo di parametri database da associare all'istanza database. Per ulteriori informazioni, consulta Utilizzo di gruppi di parametri.	Opzione CLI: `--db-parameter-group-name` Parametro API RDS: `DBParameterGroupName`	L'associazione del nuovo gruppo di parametri DB all'istanza DB avviene immediatamente.	I tempi di inattività non si verificano quando si associa un nuovo gruppo di parametri DB all'istanza DB. L'associazione di un gruppo di parametri DB è diversa dall'applicazione delle modifiche dei parametri all'interno di un gruppo di parametri. RDS applica le impostazioni modificate dei parametri statici e dinamici nel nuovo gruppo associato solo dopo aver riavviato	Tutti i motori di database

Impostazione e descrizione della console	Opzione CLI e parametro API di RDS	Quando avvengono le modifiche	Note sui tempi di inattività	Motori di database supportati
			manualmente l'istanza DB. Tuttavia, se si modificano i parametri dinamici nel gruppo di parametri DB dopo averlo associato all'istanza DB, tali impostazioni dei parametri vengono applicate immediatamente senza richiedere il riavvio. Per ulteriori informazioni, consulta Utilizzo di gruppi di parametri e Riavvio di un'istanza database.	

Impostazione e descrizione della console	Opzione CLI e parametro API di RDS	Quando avvengono le modifiche	Note sui tempi di inattività	Motori di database supportati
Volume di log dedicato Utilizza un volume di log dedicato (DLV, Dedicated Log Volume) per archiviare i log delle transazioni del database su un volume di archiviazione separato dal volume contenente le tabelle del database. Per ulteriori informazioni, consulta Utilizzo di un volume di log dedicato (DLV).	Opzione CLI: `--dedicated-log-volume` Parametro API RDS: `DedicatedLogVolume`	La modifica viene applicata al riavvio dell'istanza DB.	Si verifica inattività durante il riavvio dell'istanza database.	MariaDB, MySQL, PostgreSQL
Deletion protection (Protezione da eliminazione) L'opzione Enable deletion protection (Abilita protezione da eliminazione) permette di impedire l'eliminazione dell'istanza database. Per ulteriori informazioni, consulta Eliminazione di un'istanza database.	Opzione CLI: `--deletion-protection|--no-deletion-protection` Parametro API RDS: `DeletionProtection`	La modifica avviene immediatamente. Questa impostazione ignora l'impostazione Applica immediatamente.	Durante questa modifica non si verifica un'interruzione.	Tutti i motori di database

Impostazione e descrizione della console	Opzione CLI e parametro API di RDS	Quando avvengono le modifiche	Note sui tempi di inattività	Motori di database supportati
Enhanced Monitoring L'opzione Enable Enhanced Monitoring (Abilita monitoraggio avanzato) permette di raccogliere i parametri in tempo reale per il sistema operativo in cui viene eseguita l'istanza database. Per ulteriori informazioni, consulta Monitoraggio dei parametri del sistema operativo con il monitoraggio avanzato.	Opzione CLI: `--monitoring-interval` e `--monitoring-role-arn` Parametro API RDS: `MonitoringInterval` e `MonitoringRoleArn`	La modifica avviene immediatamente. Questa impostazione ignora l'impostazione Applica immediatamente.	Durante questa modifica non si verifica un'interruzione.	Tutti i motori di database

Impostazione e descrizione della console	Opzione CLI e parametro API di RDS	Quando avvengono le modifiche	Note sui tempi di inattività	Motori di database supportati
IAM DB authentication (Autenticazione DB IAM) Enable IAM DB authentication (Abilita autenticazione database IAM) per autenticare gli utenti del database tramite utenti e ruoli. Per ulteriori informazioni, consulta Autenticazione del database IAM per MariaDB, MySQL e PostgreSQL.	Opzione CLI: `--enable-iam-database-authentication |--no-enable-iam-database-authentication` Parametro API RDS: `EnableIAMDatabaseAuthentication`	Se si sceglie di applicare la modifica immediatamente, questa si verifica immediatamente. Se non si sceglie di applicare la modifica immediatamente, questa si verifica durante la finestra di manutenzione successiva.	Durante questa modifica non si verifica un'interruzione.	Solo MariaDB, MySQL e PostgreSQL

Impostazione e descrizione della console	Opzione CLI e parametro API di RDS	Quando avvengono le modifiche	Note sui tempi di inattività	Motori di database supportati
Autenticazione Kerberos Scegliere l'Active Directory in cui spostare l'istanza database. La directory deve esistere prima di questa operazione. Se una directory è già selezionata, è possibile specificare None (Nessuno) per rimuovere l'istanza database dalla directory corrente. Per ulteriori informazioni, consulta Autenticazione Kerberos.	Opzione CLI: `--domain` e `--domain-iam-role-name` Parametro API RDS: `Domain` e `DomainIAMRoleName`	Se si sceglie di applicare la modifica immediatamente, questa si verifica immediatamente. Se non si sceglie di applicare la modifica immediatamente, questa si verifica durante la finestra di manutenzione successiva.	Durante questa modifica si verifica un breve tempo di inattività.	Solo Microsoft SQL Server, MySQL, Oracle e PostgreSQL

Impostazione e descrizione della console	Opzione CLI e parametro API di RDS	Quando avvengono le modifiche	Note sui tempi di inattività	Motori di database supportati
License model (Modello di licenza) Scegli bring-your-own-licensedi usare la tua licenza per Db2 e Oracle. Scegliere license-included per usare il contratto di licenza generale per Microsoft SQL Server o Oracle. Per ulteriori informazioni, consulta Opzioni di licenza Amazon RDS per Db2, Licenza per Microsoft SQL Server su Amazon RDS e Opzioni di licenza per RDS per Oracle.	Opzione CLI: `--license-model` Parametro API RDS: `LicenseModel`	Se si sceglie di applicare la modifica immediata mente, questa si verifica immediata mente. Se non si sceglie di applicare la modifica immediata mente, questa si verifica durante la finestra di manutenzione successiva.	Durante questa modifica si verifica un'interruzione.	Solo Microsoft SQL Server e Oracle

Impostazione e descrizione della console	Opzione CLI e parametro API di RDS	Quando avvengono le modifiche	Note sui tempi di inattività	Motori di database supportati
Log exports (Esportazioni log) I tipi di file di log del database da pubblicare su Amazon CloudWatch Logs. Per ulteriori informazioni, consulta Pubblicazione di log di database su Amazon CloudWatch Logs.	Opzione CLI: `--cloudwatch-logs-export-configuration` Parametro API RDS: `CloudwatchLogsExportConfiguration`	La modifica avviene immediatamente. Questa impostazione ignora l'impostazione Applica immediatamente.	Durante questa modifica non si verifica un'interruzione.	Tutti i motori di database

Impostazione e descrizione della console	Opzione CLI e parametro API di RDS	Quando avvengono le modifiche	Note sui tempi di inattività	Motori di database supportati
Maintenance window (Finestra di manutenzione) Intervallo di tempo durante cui viene eseguita la manutenzione del sistema. La manutenzione del sistema include gli aggiornamenti, se applicabili. L'ora di inizio della finestra di manutenzione è indicata in base al formato Universal Coordinated Time (UTC) e la sua durata è in ore. Se imposti la finestra sull'ora corrente, tra l'ora corrente e la fine della finestra devono esserci almeno 30 minuti per garantire che tutte le modifiche in sospeso vengano applicate. Per ulteriori informazioni, consulta Finestra di manutenzione Amazon RDS.	Opzione CLI: `--preferred-maintenance-window` Parametro API RDS: `PreferredMaintenanceWindow`	La modifica avviene immediatamente. Questa impostazione ignora l'impostazione Applica immediatamente.	Se sono presenti una o più operazioni in sospeso che causano un'interruzione e la finestra di manutenzione viene modificata per includere l'ora corrente, tali operazioni in sospeso vengono applicate immediatamente e si verifica un'interruzione.	Tutti i motori di database

Impostazione e descrizione della console	Opzione CLI e parametro API di RDS	Quando avvengono le modifiche	Note sui tempi di inattività	Motori di database supportati
Gestisci le credenziali principali in AWS Secrets Manager Seleziona Manage master credentials in AWS Secrets Manager (Gestione credenziali master in AWS Secrets Manager) per gestire la password dell'utente master in un segreto di Secrets Manager. Facoltativamente, scegli la chiave KMS da utilizzare per proteggere il segreto. Scegliere tra le chiavi KMS presenti nell'account o inserire la chiave da un altro account. Se RDS sta già gestendo la password dell'utente master per l'istanza database, puoi ruotare la password dell'utente master scegliendo Rotate secret immediately (Ruota il segreto immediatamente). Per ulteriori informazioni, consulta Gestione delle password con Amazon RDS e AWS Secrets Manager.	Opzione CLI: `--manage-master-user-password | --no-manage-master-user-password` `--master-user-secret-kms-key-id` `--rotate-master-user-password | --no-rotate-master-user-password` Parametro API RDS: `ManageMasterUserPassword`	Quando attivi o disattivi la gestione automatica delle password dell'utente master, la modifica viene applicata immediatamente. Questa modifica ignora l'impostazione Apply immediately (Applica immediatamente). Quando ruoti la password dell'utente master, è necessario specificare che la modifica venga applicata immediatamente.	Durante questa modifica non si verifica un'interruzione.	Tutti i motori di database

Impostazione e descrizione della console	Opzione CLI e parametro API di RDS	Quando avvengono le modifiche	Note sui tempi di inattività	Motori di database supportati
	`MasterUserSecretKmsKeyId` `RotateMasterUserPassword`			
Multi-AZ deployment (Implementazione Multi-AZ) Yes (Sì) per distribuire l'istanza database in zone di disponibilità multiple. In caso contrario, no. Per ulteriori informazioni, consulta Configurazione e gestione di un'implementazione multi-AZ.	Opzione CLI: `--multi-az|--no-multi-az` Parametro API RDS: `MultiAZ`	Se si sceglie di applicare la modifica immediatamente, questa si verifica immediatamente. Se non si sceglie di applicare la modifica immediatamente, questa si verifica durante la finestra di manutenzione successiva.	Durante questa modifica non si verifica un'interruzione. Tuttavia, è possibile riscontrare un impatto sulle prestazioni. Per ulteriori informazioni, consulta Trasformazione di un'istanza database in implementazione d'istanza database Multi-AZ.	Tutti i motori di database

Impostazione e descrizione della console	Opzione CLI e parametro API di RDS	Quando avvengono le modifiche	Note sui tempi di inattività	Motori di database supportati
Tipo di rete I protocolli di indirizzo IP supportati dall'istanza database. IPv4 per specificare che le risorse possono comunicare con l'istanza database solo tramite il protocollo di indirizzo IP versione 4 (IPv4). Modalità dual-stack per specificare che le risorse possono comunicare con l'istanza database tramite IPv4, IPv6 o entrambi i protocolli. Utilizza la modalità dual-stack se le risorse devono comunicare con l'istanza database tramite il protocollo di indirizzo IPv6. Inoltre, assicurati di associare un blocco CIDR IPv6 a tutte le sottoreti del gruppo di sottoreti DB specificato. Per ulteriori informazioni, consulta Assegnazione di indirizzi IP in Amazon RDS.	Opzione CLI: `--network-type` Parametro API RDS: `NetworkType`	Se si sceglie di applicare la modifica immediatamente, questa si verifica immediatamente. Se non si sceglie di applicare la modifica immediatamente, questa si verifica durante la finestra di manutenzione successiva.	Durante questa modifica è possibile che si verifichino tempi di inattività.	Tutti i motori di database

Impostazione e descrizione della console	Opzione CLI e parametro API di RDS	Quando avvengono le modifiche	Note sui tempi di inattività	Motori di database supportati
New master password (Nuova password master) Password dell'utente master. La password deve contenere da 8 a 41 caratteri alfanumerici.	Opzione CLI: `--master-user-password` Parametro API RDS: `MasterUserPassword`	La modifica viene applicata in modo asincrono, appena possibile. Questa impostazione ignora l'impostazione Applica immediatamente.	Durante questa modifica non si verifica un'interruzione.	Tutti i motori di database

Impostazione e descrizione della console	Opzione CLI e parametro API di RDS	Quando avvengono le modifiche	Note sui tempi di inattività	Motori di database supportati
Option group (Gruppo di opzioni) Gruppo di opzioni da associare all'istanza database. Per ulteriori informazioni, consulta Uso di gruppi di opzioni.	Opzione CLI: `--option-group-name` Parametro API RDS: `OptionGroupName`	Se si sceglie di applicare la modifica immediatamente, questa si verifica immediatamente. Se non si sceglie di applicare la modifica immediatamente, questa si verifica durante la finestra di manutenzione successiva.	Durante questa modifica non si verifica un'interruzione. Un'eccezione è l'aggiunta del plug-in per audit MariaDB a un'istanza database RDS per MariaDB o RDS per MySQL, che potrebbe causare un'interruzione.	Tutti i motori di database

Impostazione e descrizione della console	Opzione CLI e parametro API di RDS	Quando avvengono le modifiche	Note sui tempi di inattività	Motori di database supportati
Approfondimenti sulle prestazioni Enable Performance Insights (Abilita Performance Insights) per monitorare il carico delle istanze database e consentire l'analisi e la risoluzione dei problemi di prestazioni del database. Performance Insights non è disponibile per alcune versioni del motore di database e classi di istanza database. La sezione Performance Insights non viene visualizzata nella console se non è disponibile per l'istanza database. Per ulteriori informazioni, consulta Monitoraggio del carico DB con Performance Insights su Amazon RDS e Supporto di classe di istanza, regione e motore di database Amazon RDS per Performance Insights.	Opzione CLI: `--enable-performance-insights| --no-enable-performance-insights` Parametro API RDS: `EnablePerformanceInsights`	La modifica avviene immediatamente. Questa impostazione ignora l'impostazione Applica immediatamente.	Durante questa modifica non si verifica un'interruzione.	Tutti tranne Db2

Impostazione e descrizione della console	Opzione CLI e parametro API di RDS	Quando avvengono le modifiche	Note sui tempi di inattività	Motori di database supportati
Approfondimenti sulle prestazioni AWS KMS key L'identificatore AWS KMS chiave AWS KMS key per la crittografia dei dati di Performance Insights. L'identificatore chiave è Amazon Resource Name (ARN) AWS KMS , identificatore chiave o alias chiave per la chiave KMS. Per ulteriori informazioni, consulta Attivazione e disattivazione di Performance Insights.	Opzione CLI: `--performance-insights-kms-key-id` Parametro API RDS: `PerformanceInsightsKMSKeyId`	La modifica avviene immediatamente. Questa impostazione ignora l'impostazione Applica immediatamente.	Durante questa modifica non si verifica un'interruzione.	Tutti tranne Db2

Impostazione e descrizione della console	Opzione CLI e parametro API di RDS	Quando avvengono le modifiche	Note sui tempi di inattività	Motori di database supportati
Performance Insights retention period (Periodi di retention di Performance Insights) Quantità di tempo, espressa in giorni, per cui conservare i dati di Performance Insights. L'impostazione del periodo di conservazione nel livello gratuito è Default (7 days) (Impostazione predefinita (7 giorni)). Per mantenere i dati sulle prestazioni più a lungo, specifica da 1 a 24 mesi. Per altre informazioni sui periodi di conservazione, consulta Prezzi e conservazione dei dati per Performance Insights. Per ulteriori informazioni, consulta Attivazione e disattivazione di Performance Insights.	Opzione CLI: `--performance-insights-retention-period` Parametro API RDS: `PerformanceInsightsRetentionPeriod`	La modifica avviene immediatamente. Questa impostazione ignora l'impostazione Applica immediatamente.	Durante questa modifica non si verifica un'interruzione.	Tutti tranne Db2

Impostazione e descrizione della console	Opzione CLI e parametro API di RDS	Quando avvengono le modifiche	Note sui tempi di inattività	Motori di database supportati
Caratteristiche processore Il numero di core CPU e il numero di thread per core per la classe di istanza database dell'istanza database. Per ulteriori informazioni, consulta Configurazione del processore per una classe di istanza database in RDS per Oracle.	Opzione CLI: `--processor-features` e `--use-default-processor-features` | `--no-use-default-processor-features` Parametro API RDS: `ProcessorFeatures` e `UseDefaultProcessorFeatures`	Se si sceglie di applicare la modifica immediatamente, questa si verifica immediatamente. Se non si sceglie di applicare la modifica immediatamente, questa si verifica durante la finestra di manutenzione successiva.	Durante questa modifica si verifica un'interruzione.	Solo Oracle

Impostazione e descrizione della console	Opzione CLI e parametro API di RDS	Quando avvengono le modifiche	Note sui tempi di inattività	Motori di database supportati
IOPS con provisioning Il valore di Provisioned IOPS (operazioni di I/O al secondo) per l'istanza database. Questa impostazione è disponibile solo se scegli una delle seguenti opzioni per Storage type (Tipo di archiviazione): • SSD per uso generico (gp3) • SSD per capacità di IOPS allocata (io1) • SSD IOPS fornito (io2) Per ulteriori informazioni, consulta the section called “Storage Provisioned IOPS” e the section called “archiviazione gp3 (consigliata)”.	Opzione CLI: `--iops` Parametro API RDS: `Iops`	Se si sceglie di applicare la modifica immediatamente, questa si verifica immediatamente. Se non si sceglie di applicare la modifica immediatamente, questa si verifica durante la finestra di manutenzione successiva.	Durante questa modifica non si verifica un'interruzione.	Tutti i motori di database

Impostazione e descrizione della console	Opzione CLI e parametro API di RDS	Quando avvengono le modifiche	Note sui tempi di inattività	Motori di database supportati
Accesso pubblico Publicly accessible (Accessibile pubblicamente) per assegnare all'istanza database un indirizzo IP pubblico, ovvero renderla accessibile al di fuori del VPC. Per essere accessibile pubblicamente, l'istanza database deve anche trovarsi in una sottorete pubblica nel VPC. Not publicly accessible (Non accessibile pubblicamente) per rendere l'istanza database accessibile solo dall'interno del VPC. Per ulteriori informazioni, consulta Nascondere istanze database in un VPC da Internet. Per connettersi a un'istanza database dall'esterno del proprio VPC, l'istanza database deve essere accessibile pubblicamente. Inoltre, l'accesso deve essere concesso utilizzando le regole in ingresso del gruppo di sicurezza dell'istanza database e devono essere soddisfatti altri requisiti. Per ulteriori informazioni,	Opzione CLI: `--publicly-accessible | --no-publicly-accessible` Parametro API RDS: `PubliclyAccessible`	La modifica avviene immediatamente. Questa impostazione ignora l'impostazione Applica immediatamente.	Durante questa modifica non si verifica un'interruzione.	Tutti i motori di database

Impostazione e descrizione della console	Opzione CLI e parametro API di RDS	Quando avvengono le modifiche	Note sui tempi di inattività	Motori di database supportati
consulta Impossibile connettersi all'istanza database di Amazon RDS. Se la tua istanza DB non è accessibile pubblicamente, puoi anche utilizzare una connessione AWS VPN da sito a sito o AWS Direct Connect una connessione per accedervi da una rete privata. Per ulteriori informazioni, consulta Riservatezza del traffico Internet.				
Gruppo di sicurezza Gruppo di sicurezza VPC da associare all'istanza database. Per ulteriori informazioni, consulta Controllo dell'accesso con i gruppi di sicurezza.	Opzione CLI: `--vpc-security-group-ids` Parametro API RDS: `VpcSecurityGroupIds`	La modifica viene applicata in modo asincrono, appena possibile. Questa impostazione ignora l'impostazione Applica immediatamente.	Durante questa modifica non si verifica un'interruzione.	Tutti i motori di database

Impostazione e descrizione della console	Opzione CLI e parametro API di RDS	Quando avvengono le modifiche	Note sui tempi di inattività	Motori di database supportati
Storage autoscaling (Auto Scaling dello storage) Enable storage autoscaling (Abilita Auto Scaling) affinché Amazon RDS aumenti automaticamente lo storage quando necessario, così da evitare che l'istanza database termini lo spazio di storage. Maximum storage threshold (Soglia massima di storage) per impostare il limite superiore affinché Amazon RDS aumenti automaticamente lo storage per l'istanza database. Il valore predefinito è 1.000. Per ulteriori informazioni, consulta Gestione della capacità automaticamente con Auto Scaling dello storage Amazon RDS.	Opzione CLI: `--max-allocated-storage` Parametro API RDS: `MaxAllocatedStorage`	La modifica avviene immediatamente. Questa impostazione ignora l'impostazione Applica immediatamente.	Durante questa modifica non si verifica un'interruzione.	Tutti i motori di database

Impostazione e descrizione della console	Opzione CLI e parametro API di RDS	Quando avvengono le modifiche	Note sui tempi di inattività	Motori di database supportati
Velocità di trasmissione effettiva per archiviazione Il nuovo valore della velocità di trasmissione effettiva per archiviazione dell'istanza database. Questa impostazione è disponibile solo se scegli SSD per generico (gp3) come Tipo di archiviazione. Per ulteriori informazioni, consulta the section called “archiviazione gp3 (consigliata)”.	Opzione CLI: `--storage-throughput` Parametro API RDS: `StorageThroughput`	Se si sceglie di applicare la modifica immediatamente, questa si verifica immediatamente. Se non si sceglie di applicare la modifica immediatamente, questa si verifica durante la finestra di manutenzione successiva.	Durante questa modifica non si verifica un'interruzione.	Tutti i motori di database

Impostazione e descrizione della console	Opzione CLI e parametro API di RDS	Quando avvengono le modifiche	Note sui tempi di inattività	Motori di database supportati
Storage Type (Tipo di storage) Tipo di storage da utilizzare. Se scegli SSD per uso generico (gp3), puoi aggiungere capacità di IOPS allocata e velocità di trasmissione effettiva per archiviazione in Advanced settings (Impostazioni avanzate). Se scegli Provisioned IOPS SSD (io1) o Provisioned IOPS SSD (io2), inserisci il valore Provisioned IOPS. Quando Amazon RDS inizia a modificare l'istanza database per modificare le dimensioni o il tipo dell'archiviazione, non puoi inviare un'altra richiesta per modificare le dimensioni, le prestazioni o il tipo di archiviazione prima di sei ore. Per ulteriori informazioni, consulta Tipi di storage Amazon RDS.	Opzione CLI: `--storage-type` Parametro API RDS: `StorageType`	Se si sceglie di applicare la modifica immediatamente, questa si verifica immediatamente. Se non si sceglie di applicare la modifica immediatamente, questa si verifica durante la finestra di manutenzione successiva.	Le modifiche seguenti comportano tutte una breve interruzione durante l'avvio del processo. Successivamente, è possibile utilizzare il database normalmente mentre la modifica viene applicata. • Da General Purpose (SSD) (SSD per scopi generici) o Capacità di IOPS allocata (SSD) in Magnetic. • Da Magnetic a General Purpose (SSD) (SSD	Tutti i motori di database

Impostazione e descrizione della console	Opzione CLI e parametro API di RDS	Quando avvengono le modifiche	Note sui tempi di inattività	Motori di database supportati
			per scopi generici) o Capacità di IOPS allocata (SSD).	
DB subnet group (Gruppo di sottoreti DB) Gruppo di sottoreti database per l'istanza database. Puoi utilizzare questa impostazione per spostare l'istanza database in un VPC diverso. Per ulteriori informazioni, consulta VPC di Amazon VPC e Amazon RDS.	Opzione CLI: `--db-subnet-group-name` Parametro API RDS: `DBSubnetGroupName`	Se si sceglie di applicare la modifica immediatamente, questa si verifica immediatamente. Se non si sceglie di applicare la modifica immediatamente, questa si verifica durante la finestra di manutenzione successiva.	Durante questa modifica si verifica un'interruzione.	Tutti i motori di database

Manutenzione di un'istanza database

Periodicamente, Amazon RDS esegue la manutenzione delle risorse Amazon RDS. La manutenzione spesso comporta aggiornamenti alle seguenti risorse dell'istanza database:

- Hardware sottostante
- Sistema operativo (OS) sottostante
- Versione del motore del database

Gli aggiornamenti al sistema operativo si verificano generalmente per problemi di sicurezza. È opportuno eseguirli il prima possibile.

Per alcune operazioni di manutenzione, Amazon RDS deve portare offline l'istanza database per un breve intervallo di tempo. Tra le operazioni di manutenzione che richiedono l'impostazione offline di una risorsa si annovera l'applicazione delle patch necessarie al sistema operativo o al database. L'applicazione delle patch necessarie viene pianificata automaticamente solo per le patch correlate alla sicurezza e all'affidabilità dell'istanza. Tali patch si verificano raramente, in genere una volta ogni pochi mesi. Raramente richiedono più di una frazione del periodo di manutenzione.

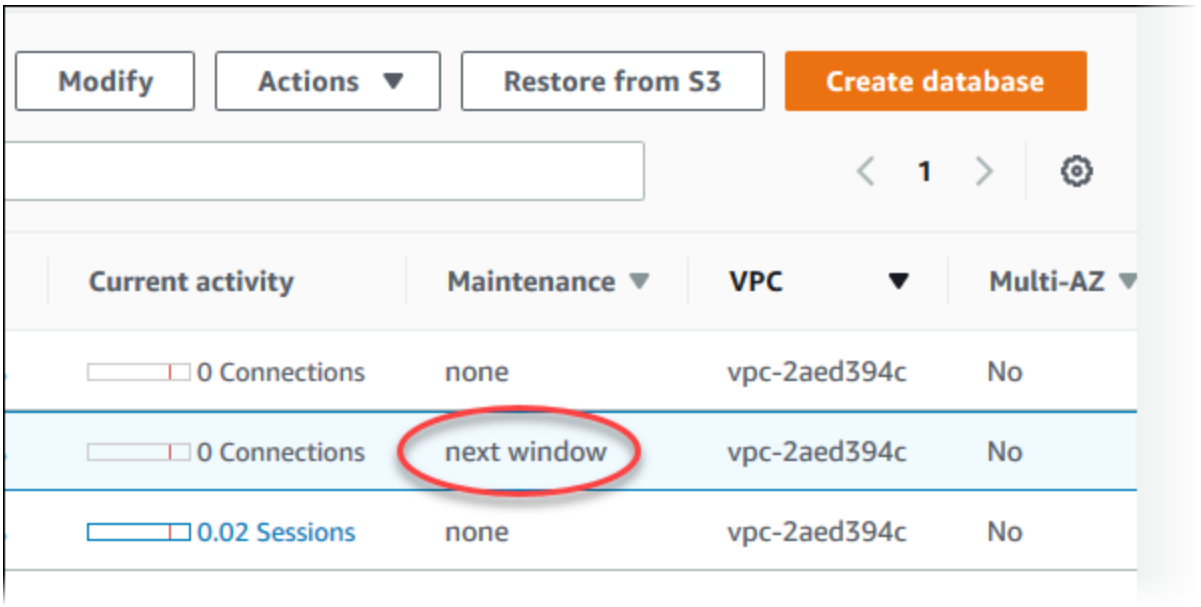
Le modifiche dell'istanza database differita che si è scelto di non applicare immediatamente vengono applicate durante la finestra di manutenzione. Ad esempio, puoi scegliere di modificare la classe di istanza database o il gruppo di parametri durante la finestra di manutenzione. Le modifiche specificate utilizzando l'impostazione di riavvio in sospeso non vengono visualizzate nell'elenco Manutenzione in sospeso . Per ulteriori informazioni sulla modifica di un'istanza database, consulta Modifica di un'istanza database Amazon RDS.

Per vedere le modifiche in sospeso per la prossima finestra di manutenzione, usa il describe-db-instances AWS CLI comando e controlla il `PendingModifiedValues` campo.

Argomenti

Visualizzazione della manutenzione in sospeso

Verifica se è disponibile un aggiornamento di manutenzione per il di istanze DB utilizzando la console RDS, l'API RDS o l' AWS CLI API RDS. Se è disponibile un aggiornamento, viene indicato nella colonna Maintenance (Manutenzione) per il dell'istanza database nella console Amazon RDS, come illustrato di seguito.



Se non è disponibile alcun aggiornamento di manutenzione per il di un'istanza database, il valore della colonna corrispondente è none (nessuno).

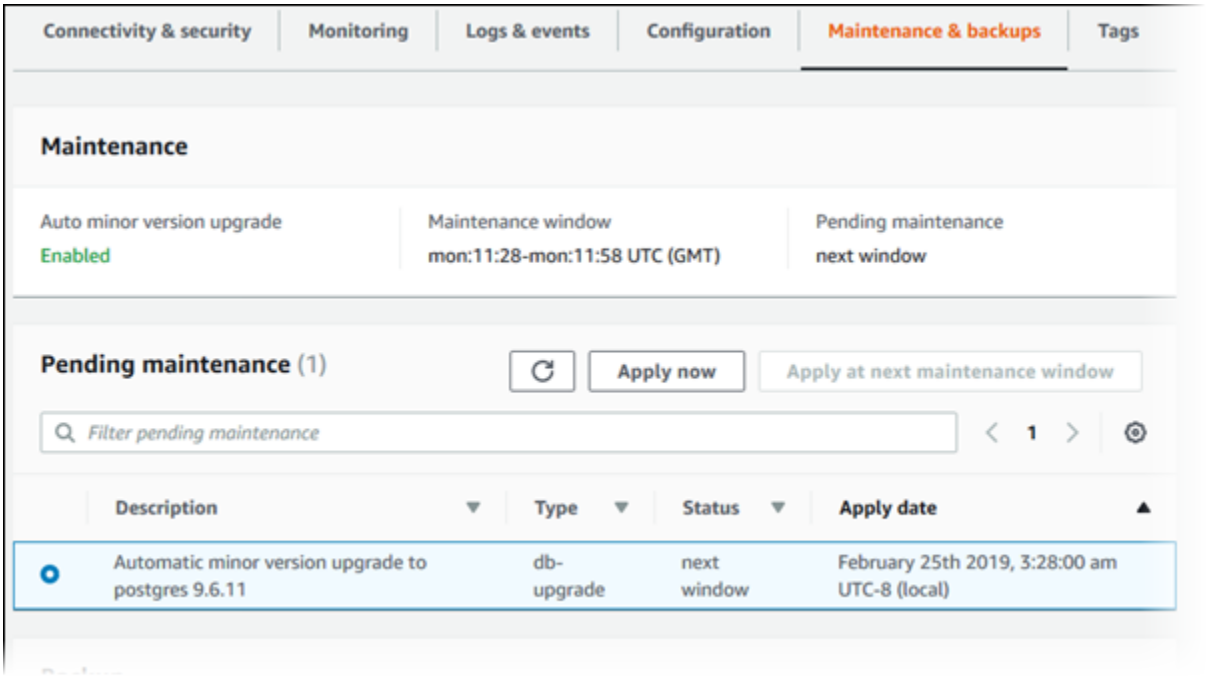
Se è disponibile un aggiornamento di manutenzione per il di un'istanza database, la colonna può avere i seguenti valori:

- richiesto – L'operazione di manutenzione sarà applicata alla risorsa e non può essere a tempo indeterminato.
- available (disponibile) – L'operazione di manutenzione è disponibile ma non sarà automaticamente applicata alla risorsa. Puoi applicarla manualmente.
- next window (finestra successiva) – L'operazione di manutenzione sarà applicata alla risorsa durante la finestra di manutenzione successiva.
- In progress (In corso) – L'operazione di manutenzione è in fase di applicazione alla risorsa.

Se è disponibile un aggiornamento, puoi scegliere tra una di queste operazioni:

- Se il valore di manutenzione è next window (finestra successiva), posticipare le operazioni di manutenzione scegliendo defer upgrade (posticipa aggiornamento) da Actions (Operazioni). Non puoi rinviare un'azione di manutenzione se è già stata avviata.
- Applicare immediatamente le operazioni di manutenzione.
- Pianificare le operazioni di manutenzione affinché vengano avviate durante la successiva finestra di manutenzione.
- Non eseguire alcuna operazione.

Per eseguire un'operazione, scegliere il dell'istanza database per mostrarne i dettagli, quindi selezionare Maintenance & backups (Manutenzione e backup). Vengono visualizzate le operazioni di manutenzione in sospeso.



La finestra di manutenzione determina l'avvio delle operazioni in sospeso, ma non limita il tempo di esecuzione totale di tali operazioni. Non è garantito che le operazioni di manutenzione terminino prima della fine della finestra di manutenzione e potrebbero continuare oltre l'ora di fine specificata. Per ulteriori informazioni, consulta Finestra di manutenzione Amazon RDS.

È inoltre possibile verificare se è disponibile un aggiornamento di manutenzione per il di istanze DB eseguendo il `describe-pending-maintenance-actions` AWS CLI comando.

Applicazione di aggiornamenti a un'istanza database

Con Amazon RDS, puoi scegliere quando eseguire le operazioni di manutenzione. Puoi decidere quando Amazon RDS applicare gli aggiornamenti utilizzando la console RDS, AWS Command Line Interface (AWS CLI) o l'API RDS.

> **Note**
>
> Per RDS per SQL Server, è possibile applicare un aggiornamento al sistema operativo sottostante arrestando e avviando l'istanza database oppure aumentando e quindi riducendo la classe dell'istanza database.

Console

Per gestire un aggiornamento per un'istanza database

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, scegliere Databases (Database).
3. Scegliere il dell'istanza database da aggiornare.
4. Per Actions (Operazioni), scegliere una delle seguenti opzioni:
   - Aggiorna ora
   - Aggiornamento alla finestra successiva

   > **Note**
   >
   > Se si sceglie Upgrade at next window (Aggiornamento alla finestra successiva) e successivamente si desidera ritardare l'aggiornamento, è possibile scegliere Defer upgrade (Rinvia aggiornamento). Non puoi rinviare un'azione di manutenzione se è già stata avviata.
   > Per annullare un'azione di manutenzione, modificare l'istanza DB e disabilitare l'aggiornamento automatico della versione minore.

AWS CLI

Per applicare un aggiornamento in sospeso a un di istanze DB, usa il apply-pending-maintenance-action AWS CLI comando.

Example

Per LinuxmacOS, oUnix:

```
aws rds apply-pending-maintenance-action \
    --resource-identifier arn:aws:rds:us-west-2:001234567890:db:mysql-db \
    --apply-action system-update \
    --opt-in-type immediate
```

Per Windows:

```
aws rds apply-pending-maintenance-action ^
    --resource-identifier arn:aws:rds:us-west-2:001234567890:db:mysql-db ^
    --apply-action system-update ^
    --opt-in-type immediate
```

> **Note**
>
> Per rinviare un'azione di manutenzione, specificare `undo-opt-in` per `--opt-in-type`. Non è possibile specificare `undo-opt-in` per `--opt-in-type` se l'azione di manutenzione è già stata avviata.
> Per annullare un'azione di manutenzione, esegui il modify-db-instance AWS CLI comando e specifica`--no-auto-minor-version-upgrade`.

Per restituire un elenco di risorse con almeno un aggiornamento in sospeso, usa il describe-pending-maintenance-actions AWS CLI comando.

Example

Per LinuxmacOS, oUnix:

```
aws rds describe-pending-maintenance-actions \
    --resource-identifier arn:aws:rds:us-west-2:001234567890:db:mysql-db
```

Per Windows:

```
aws rds describe-pending-maintenance-actions ^
    --resource-identifier arn:aws:rds:us-west-2:001234567890:db:mysql-db
```

È inoltre possibile restituire un elenco di risorse per un di istanze DB specificando il `--filters` parametro del `describe-pending-maintenance-actions` AWS CLI comando. Il formato del comando `--filters` è `Name=`*`filter-name`*`,Value=`*`resource-id`*`,...`.

Di seguito sono indicati i valori accettati per il parametro `Name` di un filtro:

- `db-instance-id` – Accetta un elenco di Amazon Resource Name (ARN) o identificatori istanze database. L'elenco restituito include solo le operazioni di manutenzione in sospeso per le istanze database specificate da questi identificatori o ARN.
- `db-cluster-id` – Accetta un elenco di identificatori di cluster database o ARN per Amazon Aurora. L'elenco restituito include solo le operazioni di manutenzione in sospeso per i cluster database specificati da questi identificatori o ARN.

L'esempio seguente restituisce le operazioni di manutenzione in sospeso per le istanze database `sample-instance1` e `sample-instance2`.

Example

Per LinuxmacOS, oUnix:

```
aws rds describe-pending-maintenance-actions \
 --filters Name=db-instance-id,Values=sample-instance1,sample-instance2
```

Per Windows:

```
aws rds describe-pending-maintenance-actions ^
 --filters Name=db-instance-id,Values=sample-instance1,sample-instance2
```

API RDS

Per applicare un aggiornamento a un'istanza database, chiamare l'operazione `ApplyPendingMaintenanceAction` dell'API Amazon RDS.

Per ottenere un elenco delle risorse con almeno un aggiornamento in sospeso, chiamare l'operazione API Amazon RDS `DescribePendingMaintenanceActions`.

Manutenzione per le implementazioni Multi-AZ

L'esecuzione di un'istanza database come implementazione multi-AZ permette di ridurre ulteriormente l'impatto di un evento di manutenzione. Questo perché Amazon RDS applica gli aggiornamenti del sistema operativo seguendo questi passaggi:

1. Esecuzione della manutenzione nell'istanza di standby.
2. Promozione dell'istanza di standby a primaria.
3. Esecuzione della manutenzione nell'istanza primaria precedente, che diventa la nuova istanza di standby.

Quando il motore di database per l'istanza database viene aggiornato in un'implementazione Multi-AZ, Amazon RDS modifica contemporaneamente l'istanza database primaria e quella secondaria. In questo caso, l'istanza database primaria e quella secondaria nell'implementazione Multi-AZ non sono disponibili durante l'aggiornamento. Finché l'aggiornamento non è completato, l'operazione causa tempi di inattività. La durata dell'interruzione varia in base alla dimensione dell'istanza database.

Se è necessario applicare le patch del sistema operativo sottostante, è necessario un breve failover Multi-AZ per applicare le patch all'istanza database principale. Questo failover dura in genere meno di un minuto.

Se la tua istanza DB esegue RDS per MySQL, RDS per PostgreSQL o RDS per MariaDB, puoi ridurre al minimo i tempi di inattività necessari per un aggiornamento utilizzando una distribuzione blu/verde. Per ulteriori informazioni, consulta Utilizzo delle implementazioni blu/verde Amazon RDS per gli aggiornamenti del database. Se esegui l'upgrade di un'istanza DB RDS per SQL Server o RDS Custom per SQL Server in una distribuzione Multi-AZ, Amazon RDS esegue gli aggiornamenti in sequenza, quindi si verifica un'interruzione solo per la durata di un failover. Per ulteriori informazioni, consulta Considerazioni su Multi-AZ e sull'ottimizzazione in memoria.

Se l'istanza database esegue RDS per SQL Server in un'implementazione multi-AZ, è possibile applicare un aggiornamento al sistema operativo sottostante utilizzando uno dei seguenti metodi:

- Modifica la classe dell'istanza database con una dimensione diversa, quindi modificala di nuovo riportandola alla dimensione originale.
- Aumenta la dimensione dell'istanza database e quindi riducila alla dimensione originale.
- Modifica l'istanza database da Multi-AZ a Single-AZ, arresta e avvia l'istanza database, quindi ripristina l'istanza in Multi-AZ.

Per ulteriori informazioni sulle implementazioni Multi-AZ, consulta Configurazione e gestione di un'implementazione multi-AZ.

Finestra di manutenzione Amazon RDS

Le finestre di manutenzione sono un intervallo di tempo settimanale durante il quale vengono applicate le modifiche al sistema. Ogni di istanze DB ha una finestra di manutenzione settimanale. La finestra di manutenzione come opportunità per controllare quando vengono apportate modifiche e patch al software.

Durante l'esecuzione delle attività di manutenzione, RDS utilizza alcune risorse dell'istanza database . Ciò potrebbe influire, in modo minimo, sulle prestazioni. Per un'istanza database, in rari casi, potrebbe essere necessario un failover Multi-AZ per il completamento di un aggiornamento di manutenzione.

Se un evento di manutenzione è pianificato per una settimana specifica, viene avviato durante la finestra di manutenzione di 30 minuti indicata. La maggior parte degli eventi di manutenzione viene completata durante la finestra di manutenzione di 30 minuti, tuttavia l'esecuzione degli eventi di manutenzione di dimensioni maggiori può richiedere più di 30 minuti per il completamento. La finestra di manutenzione viene sospesa quando il di istanze DB viene arrestato.

La finestra di manutenzione di 30 minuti è selezionata a caso da un blocco di tempo di 8 ore per regione. Se non specifichi una finestra di manutenzione quando crei l'istanza database, Amazon RDS assegna una finestra di manutenzione di 30 minuti in un giorno della settimana selezionato in modo casuale.

Di seguito sono indicati, per ogni regione, gli intervalli di tempo durante cui viene assegnata la finestra di manutenzione predefinita.

Nome della regione	Regione	Periodo di tempo
US East (Ohio)	us-east-2	03:00 - 11:00 UTC
US East (N. Virginia)	us-east-1	03:00 - 11:00 UTC
US West (N. Californi a)	us-west-1	06:00 - 14:00 UTC
US West (Oregon)	us-west-2	06:00 - 14:00 UTC

Nome della regione	Regione	Periodo di tempo
Africa (Cape Town)	af-south-1	03:00 - 11:00 UTC
Asia Pacific (Hong Kong)	ap-east-1	06:00 - 14:00 UTC
Asia Pacific (Hyderabad)	ap-south-2	06:30 - 14:30 UTC
Asia Pacifico (Giacarta)	ap-southeast-3	08:00–16:00 UTC
Asia Pacifico (Melbourne)	ap-southeast-4	11:00 - 19:00 UTC
Asia Pacific (Mumbai)	ap-south-1	06:00 - 14:00 UTC
Asia Pacific (Osaka)	ap-northeast-3	22:00 - 23:59 UTC
Asia Pacific (Seoul)	ap-northeast-2	13:00 - 21:00 UTC
Asia Pacific (Singapore)	ap-southeast-1	14:00 - 22:00 UTC
Asia Pacific (Sydney)	ap-southeast-2	12:00 - 20:00 UTC
Asia Pacific (Tokyo)	ap-northeast-1	13:00 - 21:00 UTC
Canada (Central)	ca-central-1	03:00 - 11:00 UTC
Canada occidentale (Calgary)	ca-west-1	18:00 - 02:00 UTC
China (Beijing)	cn-north-1	06:00 - 14:00 UTC
China (Ningxia)	cn-northwest-1	06:00 - 14:00 UTC
Europe (Frankfurt)	eu-central-1	21:00 - 05:00 UTC
Europe (Ireland)	eu-west-1	22:00 - 06:00 UTC

Nome della regione	Regione	Periodo di tempo
Europe (London)	eu-west-2	22:00 - 06:00 UTC
Europa (Milano)	eu-south-1	02:00 - 10:00 UTC
Europe (Paris)	eu-west-3	23:59 - 07:29 UTC
Europa (Spagna)	eu-south-2	02:00 - 10:00 UTC
Europe (Stockholm)	eu-north-1	23:00 - 07:00 UTC
Europa (Zurigo)	eu-central-2	02:00 - 10:00 UTC
Israele (Tel Aviv)	il-central-1	03:00 - 11:00 UTC
Medio Oriente (Bahrein)	me-south-1	06:00 - 14:00 UTC
Medio Oriente (Emirati Arabi Uniti)	me-central-1	05:00–13:00 UTC
Sud America (São Paulo)	sa-east-1	00:00 - 08:00 UTC
AWS GovCloud (Stati Uniti orientali)	us-gov-east-1	17:00 - 01:00 UTC
AWS GovCloud (Stati Uniti occidentali)	us-gov-west-1	06:00 - 14:00 UTC

Impostazione della finestra di manutenzione preferita dell'istanza database

La finestra di manutenzione deve avvenire nel momento dell'utilizzo più basso e pertanto potrebbe essere necessario apportare modifiche di tanto in tanto. In questo lasso di tempo, l'istanza database non è disponibile solo se le modifiche al sistema, ad esempio una modifica nella classe di istanza database, vengono applicate e richiedono un'interruzione. L'indisponibilità dell'istanza database dura solo per il tempo strettamente necessario all'esecuzione delle modifiche richieste.

Nell'esempio seguente viene impostata la finestra di manutenzione preferita per un'istanza database.

Per questo esempio supponi che sia presente un'istanza database denominata mydbinstance con una finestra di manutenzione preferita corrispondente a "Sun:05:00-Sun:06:00" UTC.

Console

Per impostare la finestra di manutenzione preferita

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, scegliere Databases (Database), quindi selezionare l'istanza database che si desidera modificare.
3. Scegliere Modify (Modifica). Viene visualizzata la pagina Modify DB Instance (Modifica istanza database).
4. Nella sezione Maintenance (Manutenzione) aggiornare a finestra di manutenzione.

   > **Note**
   >
   > La finestra di manutenzione e la finestra di backup per l'istanza database non possono sovrapporsi. Se si immette un valore per la finestra di manutenzione che si sovrappone alla finestra di backup, viene visualizzato un messaggio di errore.

5. Scegli Continue (Continua).

   Nella pagina di conferma esaminare le modifiche.
6. Per applicare immediatamente le modifiche alla finestra di manutenzione, selezionare Apply immediately (Applica immediatamente).
7. Scegliere Modifica istanza database per salvare le modifiche.

   In alternativa, scegliere Back (Indietro) per cambiare le modifiche o Cancel (Annulla) per annullare le modifiche.

AWS CLI

Per regolare la finestra di manutenzione preferita, usa il AWS CLI `modify-db-instance`comando con i seguenti parametri:

- `--db-instance-identifier`
- `--preferred-maintenance-window`

Example

L'esempio di codice seguente imposta la finestra di manutenzione su martedì dalle 4.00 alle 4.30 UTC.

Per LinuxmacOS, oUnix:

```
aws rds modify-db-instance \
--db-instance-identifier mydbinstance \
--preferred-maintenance-window Tue:04:00-Tue:04:30
```

Per Windows:

```
aws rds modify-db-instance ^
--db-instance-identifier mydbinstance ^
--preferred-maintenance-window Tue:04:00-Tue:04:30
```

API RDS

Per impostare la finestra di manutenzione preferita, utilizzare l'operazione API Amazon RDS `ModifyDBInstance` con i parametri seguenti:

- `DBInstanceIdentifier`
- `PreferredMaintenanceWindow`

Utilizzo degli aggiornamenti del sistema operativo

Le istanze RDS per Db2, RDS per MariaDB, RDS per MySQL, RDS per PostgreSQL e RDS per Oracle DB richiedono occasionalmente aggiornamenti del sistema operativo. Amazon RDS aggiorna il sistema operativo a una versione più recente per migliorare le prestazioni del database e la posizione di sicurezza generale dei clienti. In genere, gli aggiornamenti richiedono circa 10 minuti. Gli aggiornamenti del sistema operativo non modificano la versione del motore database o la classe istanza database di un'istanza database.

Gli aggiornamenti del sistema operativo possono essere facoltativi o obbligatori:

- Un aggiornamento facoltativo può essere applicato in qualsiasi momento. Sebbene questi aggiornamenti siano facoltativi, ti consigliamo di applicarli periodicamente per mantenere aggiornato il parco istanze RDS. RDS non applica automaticamente questi aggiornamenti.

Per ricevere una notifica quando diventa disponibile una nuova patch facoltativa del sistema operativo, è possibile iscriversi a RDS-EVENT-0230 nella categoria degli eventi di applicazione delle patch di sicurezza. Per informazioni sulla sottoscrizione agli eventi RDS, consulta Sottoscrizione alle notifiche eventi di Amazon RDS.

> **Note**
>
> `RDS-EVENT-0230` non si applica agli aggiornamenti di distribuzione del sistema operativo.

> **Note**
>
> Se hai ricevuto `RDS-EVENT-0230` per un'istanza database RDS per SQL Server, l'aggiornamento del sistema operativo non può essere applicato tramite l'operazione `apply-pending-maintenance`. Per ulteriori informazioni, consulta Applicazione di aggiornamenti a un'istanza database.

- Un aggiornamento obbligatorio è richiesto e dispone di una data di applicazione. Pianificare la programmazione dell'aggiornamento prima di questa data di applicazione. Dopo la data di applicazione specificata, Amazon RDS aggiorna automaticamente il sistema operativo per l'istanza database alla versione più recente durante una delle finestre di manutenzione assegnate.

> **Note**
>
> Potrebbe essere necessario rimanere aggiornati su tutti gli aggiornamenti facoltativi e obbligatori per soddisfare vari obblighi di conformità. Si consiglia di applicare regolarmente tutti gli aggiornamenti resi disponibili da RDS durante le finestre di manutenzione.

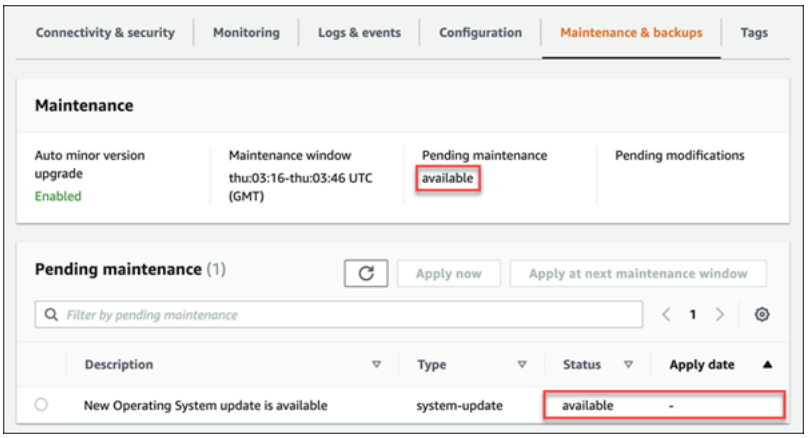
È possibile utilizzare AWS Management Console o il per ottenere informazioni sul tipo di aggiornamento del AWS CLI sistema operativo.

Console

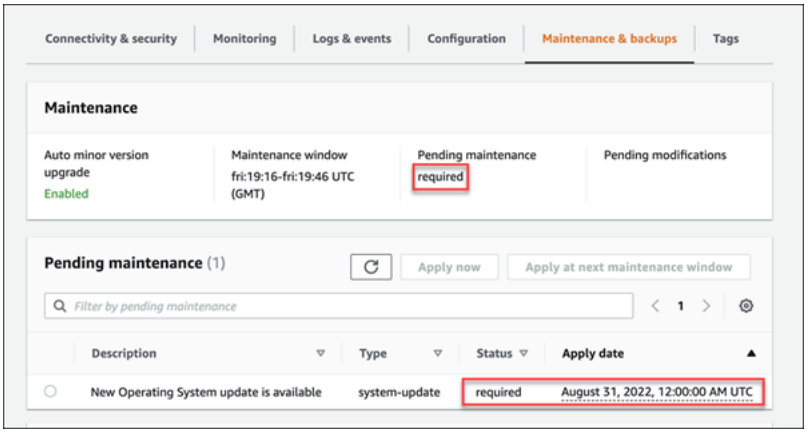
Per ottenere informazioni di aggiornamento, utilizzare AWS Management Console

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, seleziona Databases (Database), quindi scegli l'istanza database.
3. Scegli Maintenance & backups (Manutenzione e backup).
4. Nella sezione In attesa di manutenzione, cercare l'aggiornamento del sistema operativo e controllare il valore del campo Stato.

In AWS Management Console, un aggiornamento opzionale ha lo stato di manutenzione impostato su Disponibile e non ha una data di applicazione, come mostrato nell'immagine seguente.



Un aggiornamento obbligatorio ha il parametro Status (Stato) della manutenzione impostato su required (obbligatorio) e un parametro Apply date (Data di applicazione), come illustrato nell'immagine seguente.



AWS CLI

Per ottenere informazioni di aggiornamento da AWS CLI, utilizzare il describe-pending-maintenance-actionscomando.

```
aws rds describe-pending-maintenance-actions
```

Un aggiornamento obbligatorio del sistema operativo include un valore `AutoAppliedAfterDate` e un valore `CurrentApplyDate`. Un aggiornamento facoltativo del sistema operativo non include questi valori.

Il seguente output mostra un aggiornamento obbligatorio del sistema operativo.

```
{
  "ResourceIdentifier": "arn:aws:rds:us-east-1:123456789012:db:mydb1",
  "PendingMaintenanceActionDetails": [
    {
      "Action": "system-update",
      "AutoAppliedAfterDate": "2022-08-31T00:00:00+00:00",
      "CurrentApplyDate": "2022-08-31T00:00:00+00:00",
      "Description": "New Operating System update is available"
    }
  ]
}
```

Il seguente output mostra un aggiornamento facoltativo del sistema operativo.

```
{
  "ResourceIdentifier": "arn:aws:rds:us-east-1:123456789012:db:mydb2",
  "PendingMaintenanceActionDetails": [
    {
      "Action": "system-update",
      "Description": "New Operating System update is available"
    }
  ]
}
```

Disponibilità di aggiornamenti del sistema operativo

Gli aggiornamenti del sistema operativo sono specifici per la versione del motore database e la classe istanza database. Pertanto, le istanze database ricevono o richiedono aggiornamenti in

momenti diversi. Se è disponibile un aggiornamento del sistema operativo per l'istanza database basato sulla versione del motore e sulla classe di istanza, l'aggiornamento viene visualizzato nella console. Può essere visualizzato anche eseguendo AWS CLI describe-pending-maintenance-actionsil comando o chiamando l'operazione dell'DescribePendingMaintenanceActionsAPI RDS. Se è disponibile un aggiornamento per l'istanza, puoi aggiornare il sistema operativo seguendo le istruzioni in Applicazione di aggiornamenti a un'istanza database.

Aggiornamento della versione del motore di un'istanza database

Amazon RDS up-to-date Versioni più recenti possono includere correzioni di bug, miglioramenti in termini di sicurezza e altri miglioramenti per il motore di database. Quando Amazon RDS supporta una nuova versione di un motore di database, puoi scegliere come e quando aggiornare i delle istanze database.

Sono disponibili due tipi di aggiornamenti: quelli di versione principale e quelli di versione secondaria. In generale, un aggiornamento della versione principale del motore può introdurre modifiche che non sono compatibili con le applicazioni esistenti. Al contrario, un aggiornamento della versione secondaria include solo modifiche compatibili con le versioni precedenti delle applicazioni esistenti.

Per i cluster database multi-AZ, gli aggiornamenti della versione principale sono supportati solo per RDS per PostgreSQL. Gli aggiornamenti della versione secondaria sono supportati per tutti i motori che supportano i cluster database multi-AZ. Per ulteriori informazioni, consulta the section called “Aggiornamento della versione del motore di un cluster database multi-AZ”.

La sequenza di numerazione delle versioni è specifica per ogni motore di database. Ad esempio, RDS for MySQL 5.7 e 8.0 sono versioni principali del motore e l'aggiornamento da qualsiasi versione 5.7 a qualsiasi versione 8.0 è un aggiornamento della versione principale. RDS for MySQL versione 5.7.22 e 5.7.23 sono versioni secondarie e l'aggiornamento da 5.7.22 a 5.7.23 è un aggiornamento della versione secondaria.

> **Important**
>
> Non è possibile modificare un'istanza database quando viene aggiornata. Durante un aggiornamento, lo stato dell'istanza database è `upgrading`.

Per ulteriori informazioni sugli aggiornamenti delle versioni principali e secondarie per un motore di database specifico, consulta la documentazione seguente per il motore di database in uso:

- Aggiornamento del motore di database MariaDB
- Aggiornamento del motore di database Microsoft SQL Server
- Aggiornamento del motore di database MySQL
- Aggiornamento del motore di database RDS per Oracle
- Aggiornamento del motore del database PostgreSQL per Amazon RDS

Per gli aggiornamenti delle versioni principali, è necessario modificare manualmente la versione del motore DB tramite AWS Management Console AWS CLI, o l'API RDS. Per gli aggiornamenti della versione secondaria, puoi modificare manualmente la versione del motore o scegliere di abilitare l'opzione Aggiornamento automatico versione secondaria.

> **Note**
>
> Per gli aggiornamenti del motore di database si verificano tempi di inattività. È possibile ridurre al minimo i tempi di inattività necessari per l'aggiornamento dell'istanza database utilizzando un'implementazione blu/verde. Per ulteriori informazioni, consulta Utilizzo delle implementazioni blu/verde Amazon RDS per gli aggiornamenti del database.

Argomenti

Aggiornamento manuale della versione del motore

Per aggiornare manualmente la versione del motore di un'istanza DB, puoi utilizzare l' AWS Management Console, the o l' AWS CLI API RDS.

Console

Per aggiornare la versione del motore di un'istanza database tramite la console

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, scegliere Databases (Database) quindi selezionare l'istanza database da aggiornare.
3. Scegliere Modify (Modifica). Viene visualizzata la pagina Modify DB Instance (Modifica istanza database).
4. In DB engine version (Versione motore database) scegliere la nuova versione.
5. Scegliere Continue (Continua) e controllare il riepilogo delle modifiche.
6. Decidi quando pianificare l'aggiornamento. Per applicare immediatamente le modifiche, scegliere Apply immediately (Applica immediatamente). In alcuni casi, la chiusura di questa opzione può

causare un'interruzione. Per ulteriori informazioni, consulta Impostazione delle modifiche alla pianificazione.

7. Nella pagina di conferma esaminare le modifiche. Se sono corrette, seleziona Modifica istanza database per salvare le modifiche.

   In alternativa, scegliere Back (Indietro) per cambiare le modifiche o Cancel (Annulla) per annullare le modifiche.

AWS CLI

Per aggiornare la versione del motore di un'istanza DB, usa il comando CLI modify-db-instance. Specifica i seguenti parametri:

- `--db-instance-identifier` – Nome dell'istanza database.
- `--engine-version` – Numero di versione del motore di database a cui effettuare l'aggiornamento.

  Per informazioni sulle versioni valide del motore, utilizzate il AWS CLI describe-db-engine-versionscomando.
- `--allow-major-version-upgrade` – Per aggiornare la versione principale.
- `--no-apply-immediately` – Per applicare le modifiche durante la finestra di manutenzione successiva. Per applicare immediatamente le modifiche utilizzare `--apply-immediately`.

Example

Per LinuxmacOS, oUnix:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --engine-version new_version \
    --allow-major-version-upgrade \
    --no-apply-immediately
```

Per Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --engine-version new_version ^
```

```
    --allow-major-version-upgrade ^
    --no-apply-immediately
```

API RDS

Per aggiornare la versione del motore di un'istanza database, utilizza l'operazione ModifyDBInstance. Specifica i seguenti parametri:

- `DBInstanceIdentifier` – Nome dell'istanza database, ad esempio *`mydbinstance`*.
- `EngineVersion` – Numero di versione del motore di database a cui effettuare l'aggiornamento. Per informazioni sulle versioni valide del motore, utilizzate l'operazione DescribeDB EngineVersions.
- `AllowMajorVersionUpgrade` – Se consentire un aggiornamento della versione principale. A questo scopo, imposta il valore su `true`.
- `ApplyImmediately` – Indica se applicare le modifiche immediatamente o durante la finestra di manutenzione successiva. Per applicare le modifiche immediatamente, imposta il valore su `true`. Per applicare le modifiche durante la finestra di manutenzione successiva imposta il valore su `false`.

Aggiornamento automatico della versione secondaria del motore

Una versione secondaria del motore è un aggiornamento a una versione del motore di database all'interno di una versione principale del motore. Ad esempio, una versione principale del motore potrebbe essere 9.6 con al suo interno le versioni secondarie 9.6.11 e 9.6.12.

Se desideri che Amazon RDS esegua automaticamente l'aggiornamento della versione del motore di database, puoi abilitare gli aggiornamenti automatici della versione secondaria per il database.

RDS per SQL Server attualmente non supporta gli aggiornamenti automatici delle versioni secondarie.

Argomenti

- Funzionamento degli aggiornamenti automatici di versioni secondarie
- Attivazione degli aggiornamenti a versioni secondarie automatiche
- Determinazione della disponibilità degli aggiornamenti di manutenzione
- Individuazione delle destinazioni degli aggiornamenti automatici delle versioni secondarie

Funzionamento degli aggiornamenti automatici di versioni secondarie

Amazon RDS designa una versione secondaria del motore come preferita quando vengono soddisfatte le condizioni seguenti:

- Il database esegue una versione secondaria del motore di database che è inferiore alla versione secondaria del motore preferita.

  È possibile trovare la versione attuale del motore per la tua istanza DB cercando nelConfigurazionescheda della pagina dei dettagli del database o esecuzione del comando CLI`describe-db-instances`.
- Nel database è abilitato l'aggiornamento automatico della versione secondaria

RDS pianifica gli aggiornamenti in modo che vengano eseguiti automaticamente nella finestra di manutenzione. Durante l'aggiornamento, RDS esegue le seguenti operazioni di base:

1. Esegue un controllo preliminare per verificare che il database sia integro e pronto per essere aggiornato
2. Aggiorna il motore DB
3. Esegue controlli successivi all'aggiornamento
4. Contrassegna l'aggiornamento del database come completo

Gli aggiornamenti automatici comportano tempi di inattività. La durata del tempo di inattività dipende da vari fattori, tra cui il tipo di motore DB e le dimensioni del database.

Attivazione degli aggiornamenti a versioni secondarie automatiche

Puoi verificare se l'aggiornamento automatico della versione secondaria è abilitato per un'istanza database quando esegui le seguenti attività:

- Creazione di un'istanza database
- Modifica di un'istanza database
- Creazione di una replica di lettura
- Ripristino di un'istanza database da uno snapshot
- Ripristino di un'istanza database a un orario specifico
- Importazione di un'istanza database da Amazon S3 (per un backup MySQL su Amazon S3)

Quando eseguiti queste attività, puoi verificare se l'aggiornamento automatico della versione secondaria è abilitato per l'istanza database nei modi seguenti:

- Tramite la console, imposta l'opzione Auto minor version upgrade (Aggiornamento automatico della versione secondaria).
- Utilizzando AWS CLI, imposta l'`--auto-minor-version-upgrade|--no-auto-minor-version-upgrade`opzione.
- Tramite l'API RDS, imposta il parametro `AutoMinorVersionUpgrade`.

Determinazione della disponibilità degli aggiornamenti di manutenzione

Per determinare se un aggiornamento di manutenzione, ad esempio un aggiornamento della versione del motore di database, è disponibile per il di istanze DB, è possibile utilizzare la console o l'API RDS. AWS CLI Puoi anche aggiornare la versione del motore di database manualmente e regolare la finestra di manutenzione. Per ulteriori informazioni, consulta Manutenzione di un'istanza database.

Individuazione delle destinazioni degli aggiornamenti automatici delle versioni secondarie

È possibile utilizzare il AWS CLI comando seguente per determinare la versione di destinazione dell'aggiornamento secondario automatico corrente per una versione secondaria del motore DB specificata in una specifica Regione AWS. Puoi trovare i possibili valori `--engine` per questo comando nella descrizione del parametro `Engine` in CreateDBInstance.

Per LinuxmacOS, oUnix:

```
aws rds describe-db-engine-versions \
--engine engine \
--engine-version minor-version \
--region region \
--query "DBEngineVersions[*].ValidUpgradeTarget[*].
{AutoUpgrade:AutoUpgrade,EngineVersion:EngineVersion}" \
--output text
```

Per Windows:

```
aws rds describe-db-engine-versions ^
--engine engine ^
--engine-version minor-version ^
```

```
--region region ^
--query "DBEngineVersions[*].ValidUpgradeTarget[*].
{AutoUpgrade:AutoUpgrade,EngineVersion:EngineVersion}" ^
--output text
```

Ad esempio, il AWS CLI comando seguente determina l'obiettivo di aggiornamento secondario automatico per la versione secondaria di MySQL 8.0.11 nella regione Stati Uniti orientali (Ohio) (us-east-2). AWS

PermacOS, oUnix: Linux

```
aws rds describe-db-engine-versions \
--engine mysql \
--engine-version 8.0.11 \
--region us-east-2 \
--query "DBEngineVersions[*].ValidUpgradeTarget[*].
{AutoUpgrade:AutoUpgrade,EngineVersion:EngineVersion}" \
--output table
```

Per Windows:

```
aws rds describe-db-engine-versions ^
--engine mysql ^
--engine-version 8.0.11 ^
--region us-east-2 ^
--query "DBEngineVersions[*].ValidUpgradeTarget[*].
{AutoUpgrade:AutoUpgrade,EngineVersion:EngineVersion}" ^
--output table
```

L'output è simile a quello riportato di seguito.

```
---------------------------------
|    DescribeDBEngineVersions    |
+--------------+-----------------+
|  AutoUpgrade |  EngineVersion  |
+--------------+-----------------+
|  False       |  8.0.15         |
|  False       |  8.0.16         |
|  False       |  8.0.17         |
|  False       |  8.0.19         |
|  False       |  8.0.20         |
|  False       |  8.0.21         |
```

```
|  True         |  8.0.23         |
|  False        |  8.0.25         |
+---------------+-----------------+
```

In questo esempio, il valore `AutoUpgrade` è `True` per MySQL versione 8.0.23. Quindi, il target di aggiornamento secondario automatico è MySQL versione 8.0.23, che è evidenziato nell'output.

> **Important**
>
> Se intendi eseguire la migrazione di un'istanza database RDS per PostgreSQL a un cluster database Aurora PostgreSQL in futuro, ti consigliamo di disattivare gli aggiornamenti automatici delle versioni secondarie per l'istanza database all'inizio della pianificazione. La migrazione a Aurora PostgreSQL potrebbe subire un ritardo se la versione RDS for PostgreSQL non è ancora supportata da Aurora PostgreSQL. Per informazioni sulle versioni di Aurora PostgreSQL, consulta Versioni del motore per Amazon Aurora PostgreSQL.

Ridenominazione di un'istanza database

Puoi rinominare un'istanza database usando la AWS Management Console, il comando `modify-db-instance` della AWS CLI o l'operazione `ModifyDBInstance` dell'API Amazon RDS. La ridenominazione di un'istanza database può avere ripercussioni significative. Di seguito è riportato un elenco di elementi da tenere in considerazione prima di rinominare un'istanza database.

- Quando si rinomina un'istanza database, il relativo endpoint cambia, in quanto l'URL include il nome assegnato all'istanza. È sempre consigliabile reindirizzare il traffico dall'URL precedente a quello nuovo.
- Quando si rinomina un'istanza database, il nome DNS usato in precedenza dall'istanza database viene eliminato immediatamente, ma può rimanere memorizzato nella cache per alcuni minuti. Il nuovo nome DNS per l'istanza database rinominata diventa effettivo dopo circa 10 minuti. L'istanza database ridenominata non è disponibile fino a quando il nuovo nome non diventa effettivo.
- Quando si rinomina un'istanza, non è possibile usare un nome di istanza database esistente.
- Dopo che un'istanza database viene rinominata, tutte le repliche di lettura a essa associate rimangono associate. Supponi, ad esempio, di avere un'istanza database utilizzata dal database di produzione e che all'istanza siano associate diverse repliche di lettura. Se rinomini l'istanza database e quindi la sostituisci nell'ambiente di produzione con uno snapshot DB, all'istanza database rinominata restano comunque associate le repliche di lettura.
- Se riutilizzi un nome di istanza database, i parametri e gli eventi associati a tale nome vengono mantenuti. Ad esempio, se promuovi una replica di lettura e la rinomini assegnandole il nome dell'istanza database primaria precedente, gli eventi e i parametri associati all'istanza database primaria vengono associati all'istanza rinominata.
- I tag dell'istanza database rimangono con l'istanza, a prescindere dalla ridenominazione.
- Per un'istanza database ridenominata vengono mantenute le snapshot.

> **Note**
>
> Un'istanza database è un ambiente di database isolato in esecuzione nel cloud. Un'istanza database può ospitare più database o un singolo database Oracle con più schemi. Per informazioni sulla modifica del nome di un database, consulta la documentazione relativa al motore database.

Ridenominazione per la sostituzione di un'istanza database esistente

I motivi più comuni per rinominare un'istanza DB sono la promozione di una replica di lettura o il ripristino dei dati da un'istantanea o point-in-time da un ripristino del database (PITR). Rinominando il database, puoi sostituire l'istanza database senza dover modificare il codice dell'applicazione che fa riferimento all'istanza database. In questi casi, sono necessarie le operazioni seguenti:

1. Arrestare tutto il traffico diretto all'istanza database primaria. A tale scopo, può essere necessario reindirizzare il traffico che accede ai database nell'istanza database o intervenire in un altro modo, per impedire che il traffico acceda ai database nell'istanza database.
2. Rinominare l'istanza database primaria con un nome che indica che non è più l'istanza database primaria, come descritto più avanti in questo argomento.
3. Creare una nuova istanza database primaria eseguendo il ripristino da uno snapshot DB oppure promuovendo una replica di lettura e quindi assegnando alla nuova istanza il nome dell'istanza database primaria precedente.
4. Associare le repliche di lettura alla nuova istanza database primaria.

Eliminando l'istanza database primaria precedente, vengono eliminati tutti gli snapshot DB indesiderati dell'istanza database primaria precedente.

Per informazioni sulla promozione di una replica di lettura, consulta Promozione di una replica di lettura a istanza database standalone.

> **Important**
>
> L'istanza database viene riavviata quando viene rinominata.

Console

Per ridenominare un'istanza database

1. Accedi alla AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, scegliere Databases (Database).
3. Scegliere l'istanza database che si desidera rinominare.
4. Scegliere Modify (Modifica).

5. In Settings (Impostazioni) immettere un nuovo nome per DB instance identifier (Identificatore istanze DB).
6. Scegli Continue (Continua).
7. Per applicare immediatamente le modifiche, scegliere Apply immediately (Applica immediatamente). In alcuni casi, la chiusura di questa opzione può causare un'interruzione. Per ulteriori informazioni, consulta Modifica di un'istanza database Amazon RDS.
8. Nella pagina di conferma esaminare le modifiche. Se sono corrette, seleziona Modifica istanza database per salvare le modifiche.

   In alternativa, scegliere Back (Indietro) per cambiare le modifiche o Cancel (Annulla) per annullare le modifiche.

AWS CLI

Per rinominare un'istanza, utilizzare il comando AWS CLI `modify-db-instance`. Fornisci il valore corrente di `--db-instance-identifier` e il parametro `--new-db-instance-identifier` con il nuovo nome dell'istanza database.

Example

Per, o: Linux macOS Unix

```
aws rds modify-db-instance \
    --db-instance-identifier DBInstanceIdentifier \
    --new-db-instance-identifier NewDBInstanceIdentifier
```

Per Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier DBInstanceIdentifier ^
    --new-db-instance-identifier NewDBInstanceIdentifier
```

API RDS

Per rinominare un'istanza database, richiamare l'operazione dell'API Amazon RDS `ModifyDBInstance` con i seguenti parametri:

- `DBInstanceIdentifier` — nome esistente per l'istanza

- `NewDBInstanceIdentifier` — nuovo nome per l'istanza

Riavvio di un'istanza database

Puoi interrompere e avviare il servizio di database sulla tua istanza DB RDS con un'unica operazione, chiamata riavvio.

Argomenti

In genere, si riavvia l'istanza DB per motivi di manutenzione in modo che le modifiche abbiano effetto. I seguenti casi d'uso sono comuni:

- Associazione di un nuovo gruppo di parametri DB: quando si associa un nuovo gruppo di parametri DB a un'istanza DB, RDS applica i parametri statici e dinamici modificati solo dopo il riavvio dell'istanza DB. Tuttavia, se si modificano i parametri dinamici nel gruppo di parametri DB dopo averlo associato all'istanza DB, queste modifiche vengono applicate immediatamente senza riavvio. Per ulteriori informazioni, consulta Utilizzo di gruppi di parametri.
- Applicazione di una modifica a un parametro statico in un gruppo di parametri DB esistente: quando si modifica un parametro statico e si salva il gruppo di parametri DB, lo stato delle istanze DB associate a questo gruppo di parametri nella console diventa in sospeso di riavvio. La modifica dei parametri ha effetto solo dopo il riavvio delle istanze DB associate. Quando si modifica un parametro dinamico in un gruppo di parametri esistente, la modifica ha effetto immediato per impostazione predefinita, senza richiedere il riavvio.

> **Note**
>
> Lo stato di riavvio in sospeso non comporta un riavvio automatico durante la finestra di manutenzione successiva. Per applicare le ultime modifiche ai parametri all'istanza DB, riavvia l'istanza database manualmente. Per ulteriori informazioni sui gruppi di parametri, consultare Utilizzo di gruppi di parametri.

- Test del failover Multi-AZ: la strategia di test per un cluster DB Multi-AZ potrebbe comportare il riavvio dell'istanza DB principale per avviare un failover su un'altra AZ.
- Risoluzione dei problemi: potrebbero verificarsi problemi di prestazioni o altri problemi operativi che richiedono un riavvio. Ad esempio, l'istanza DB potrebbe non rispondere.

Come funziona il riavvio di un'istanza DB un cluster DB

Quando Amazon RDS riavvia l'istanza DB, esegue le seguenti attività sequenziali:

1. Interrompe il servizio di database sull'istanza DB
2. Avvia il servizio di database sull'istanza DB

Il processo di riavvio comporta una breve interruzione. Durante questa interruzione, lo stato dell'istanza DB viene riavviato. Si verifica un'interruzione per un'implementazione single-AZ e per un'implementazione di istanza database multi-AZ, anche quando si riavvia con un failover.

Come funziona il riavvio di un'istanza DB in una distribuzione Multi-AZ

Se l'istanza DB di Amazon RDS è in una distribuzione Multi-AZ, puoi riavviarla con un failover. Questa operazione è utile per simulare un guasto di un'istanza DB o ripristinare le operazioni nella zona di disponibilità originale dopo un failover.

Durante il riavvio con failover, Amazon RDS esegue le seguenti operazioni

- Interrompe bruscamente il database. L'istanza database e le sue sessioni client potrebbero non avere il tempo di chiudersi normalmente.

  > **Warning**
  >
  > Per evitare la possibilità di perdita di dati, ti consigliamo di interrompere le transazioni sull'istanza database prima di eseguire il riavvio con un failover.

- Passa automaticamente a una replica in standby in un'altra zona. La modifica AZ potrebbe non riflettersi nelle AWS Management Console chiamate e nelle chiamate all'API AWS CLI e RDS per diversi minuti.
- Aggiorna il record DNS dell'istanza DB in modo che punti all'istanza DB in standby. Di conseguenza, è necessario eliminare e ristabilire le connessioni esistenti all'istanza database. Per ulteriori informazioni, consulta Configurazione e gestione di un'implementazione multi-AZ.

- Crea un evento Amazon RDS dopo il riavvio.

In RDS per Microsoft SQL Server, il failover riavvia solo l'istanza database principale. Dopo aver eseguito il failover, l'istanza database primaria diventa la nuova istanza database secondaria. I parametri potrebbero non essere aggiornati per istanze Multi-AZ. Per il riavvio senza failover, le istanze database primarie e secondarie vengono riavviate e i parametri vengono aggiornati dopo il riavvio. Se l'istanza database non risponde, si consiglia di riavviare senza failover.

Prima di riavviare l'istanza, considera quanto segue:

- Per un'istanza database con repliche di lettura puoi riavviare l'istanza database di origine e le relative repliche di lettura in modo indipendente. Al termine del riavvio, la replica riprende automaticamente.
- Il tempo di riavvio dipende dal processo di ripristino in caso di arresto anomalo, dall'attività del database al momento del riavvio e dal comportamento del motore DB specifico. Per migliorare il tempo di riavvio, si consiglia di ridurre il più possibile l'attività del database durante il riavvio. Questa tecnica riduce l'attività di rollback per le transazioni in transito.

Assicurati di soddisfare i seguenti prerequisiti:

- L'istanza database deve essere nello stato `available`. Il database può non essere disponibile per diversi motivi, ad esempio un backup in corso, una modifica richiesta in precedenza o un'operazione durante una finestra di manutenzione.
- Se si impone un failover su un'altra AZ, l'istanza DB deve essere configurata per Multi-AZ.
- Se si impone un failover verso un'altra AZ, si consiglia innanzitutto di interrompere le transazioni sull'istanza DB per evitare possibili perdite di dati.

Riavvio di un'istanza DB

È possibile riavviare l'istanza DB utilizzando l'API AWS Management Console AWS CLI, o RDS.

Console

Per riavviare un'istanza database

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, scegliere Databases (Database) e l'istanza database da riavviare.
3. In Actions (Operazioni), scegliere Reboot (Riavvia).

   Viene visualizzata la pagina Riavvia l'istanza DB.
4. (Facoltativo) Scegliere Reboot with failover? (Riavvia con failover?) per forzare un failover da una zona di disponibilità a un'altra.
5. Scegliere Reboot (Riavvia) per riavviare l'istanza database.

   In alternativa, scegliere Cancel (Annulla).

AWS CLI

Per riavviare un'istanza DB utilizzando il AWS CLI, chiamate il `reboot-db-instance`comando.

Example Riavvio semplice

Per LinuxmacOS, oUnix:

```
aws rds reboot-db-instance \
    --db-instance-identifier mydbinstance
```

Per Windows:

```
aws rds reboot-db-instance ^
    --db-instance-identifier mydbinstance
```

Example Riavvio con failover

Per forzare un failover da una AZ all'altra in un cluster DB Multi-AZ, utilizzate il `--force-failover` parametro.

PerLinux, omacOS: Unix

```
aws rds reboot-db-instance \
```

```
    --db-instance-identifier mydbinstance \
    --force-failover
```

Per Windows:

```
aws rds reboot-db-instance ^
    --db-instance-identifier mydbinstance ^
    --force-failover
```

API RDS

Per riavviare un'istanza database tramite l'API Amazon RDS, chiamare l'operazione `RebootDBInstance`.

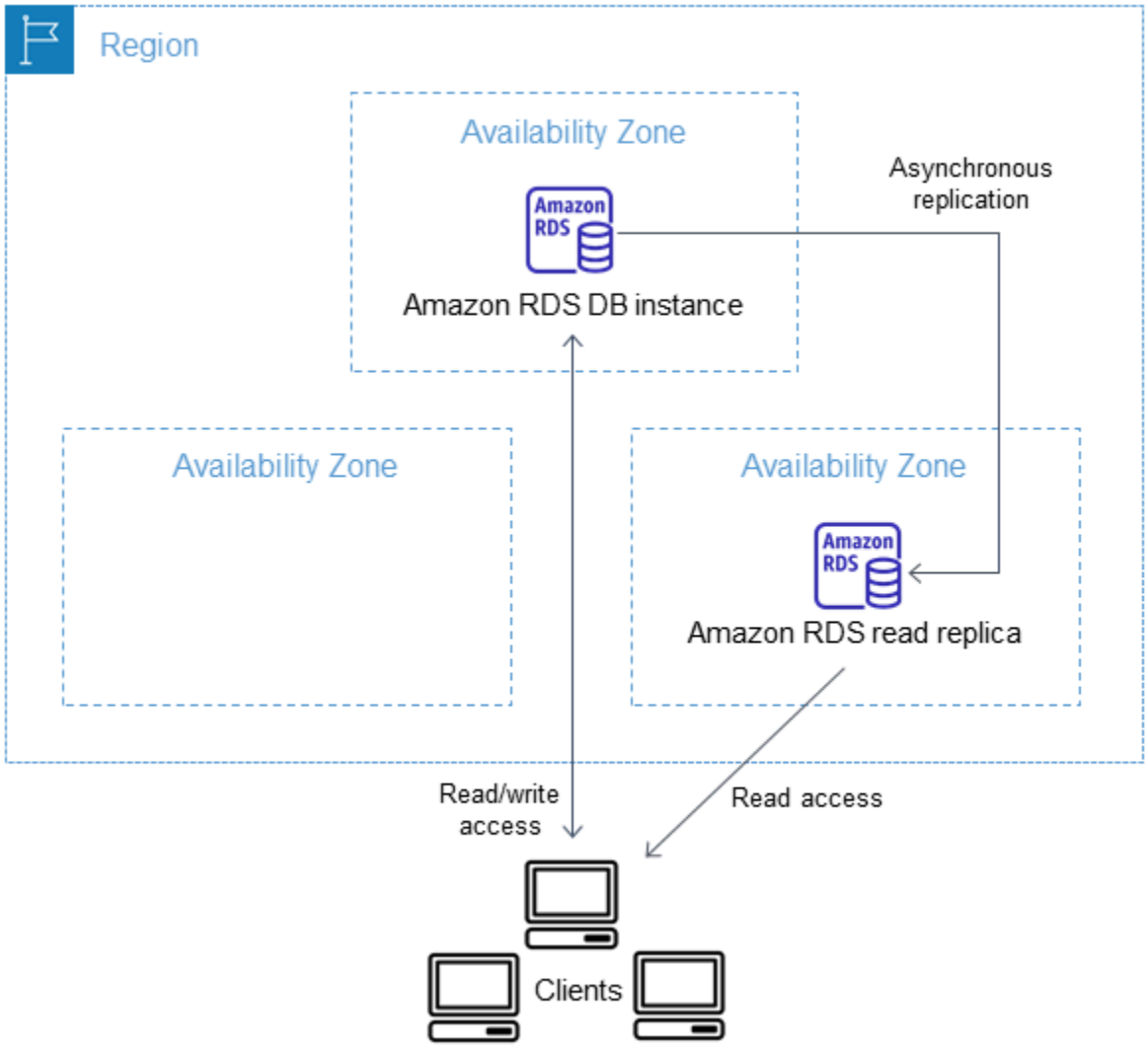
Uso delle repliche di lettura dell'istanza database

Una replica di lettura è una copia di sola lettura di un'istanza database. È possibile ridurre il carico sull'istanza database primaria instradando le query dalle applicazioni alla replica di lettura. In questo modo, è possibile impiegare la scalabilità orizzontale in modo elastico oltre i vincoli di capacità di una singola istanza database per carichi di lavoro di database particolarmente gravosi in lettura.

Per creare una replica di lettura da un'istanza database di origine, Amazon RDS utilizza le funzionalità di replica integrata del motore database. Per informazioni sull'uso di repliche di lettura con un motore specifico, consulta le sezioni seguenti:

- Uso di repliche di lettura MariaDB
- Utilizzo di repliche di lettura per Microsoft SQL Server in Amazon RDS
- Uso delle repliche di lettura MySQL
- Utilizzo di repliche di lettura per Amazon RDS per Oracle
- Utilizzo delle repliche di lettura per Amazon RDS per PostgreSQL

Dopo aver creato una replica di lettura da un'istanza database di origine, l'origine diventa l'istanza database primaria. Quando si aggiorna l'istanza database primaria, Amazon RDS copia l'aggiornamento in modo asincrono nella replica di lettura. Il diagramma seguente mostra un'istanza database di origine replicata su una replica di lettura in una zona di disponibilità (AZ) diversa. I client hanno accesso in lettura/scrittura all'istanza DB principale e accesso in sola lettura alla replica.



Argomenti

- Panoramica delle repliche di lettura di Amazon RDS
- Creazione di una replica di lettura
- Promozione di una replica di lettura a istanza database standalone
- Monitoraggio della replica di lettura
- Creazione di una replica di lettura in un altro Regione AWS

Panoramica delle repliche di lettura di Amazon RDS

Le seguenti sezioni trattano le repliche di lettura delle istanze database. Per informazioni sull'aggiunta di repliche di lettura a un cluster Multi-AZ, consulta the section called “Utilizzo delle repliche di lettura del cluster di database multi-AZ”.

Argomenti

Casi d'uso per le repliche di lettura

La distribuzione di una o più repliche di lettura per un'istanza database di origine specifica può essere una scelta logica in svariati scenari, inclusi i seguenti:

- Dimensionamento oltre la capacità di calcolo o di I/O di una singola istanza database per carichi di lavoro di database gravosi in lettura. Puoi indirizzare questo traffico in lettura in eccesso a una o più repliche di lettura.
- Assegnazione di traffico in lettura mentre l'istanza DB di origine non è disponibile. In alcuni casi, l'istanza database di origine potrebbe non riuscire a ricevere richieste di I/O, ad esempio a causa della sospensione delle operazioni di I/O in occasione dell'esecuzione di backup o della manutenzione pianificata. In questi casi puoi indirizzare il traffico in lettura verso le repliche di lettura. Per questo caso d'uso, tieni presente che i dati nella replica di lettura potrebbero restare non aggiornati, perché l'istanza database di origine non è disponibile.
- Scenari di creazione di report o di data warehousing in cui potrebbe essere necessario eseguire query per la creazione di report aziendali su una replica di lettura invece che sull'istanza DB di produzione principale.
- Implementazione del disaster recovery. Puoi promuovere una replica di lettura a un'istanza standalone come soluzione di disaster recovery in caso di errore dell'istanza database primaria.

Funzionamento delle repliche di lettura

Quando crei una replica di lettura, devi prima di tutto specificare un'istanza database esistente come origine. Amazon RDS acquisisce quindi uno snapshot dell'istanza di origine e crea un'istanza di

sola lettura dallo snapshot. Amazon RDS usa quindi il metodo di replica asincrona per il motore del database per aggiornare la replica di lettura ogni volta che viene apportata una modifica all'istanza database primaria.

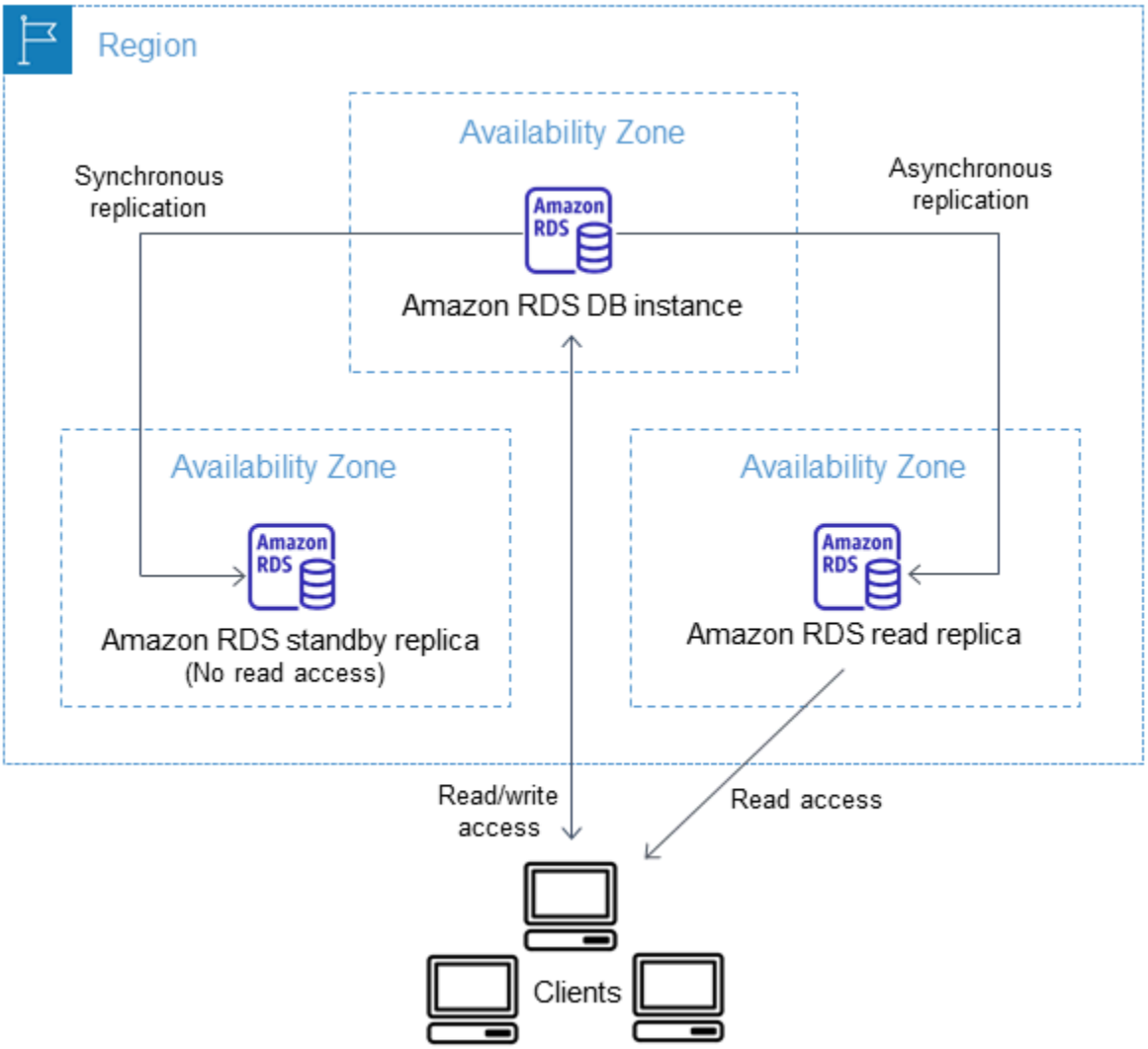
La replica di lettura opera come istanza database che permette solo connessioni di sola lettura. Un'eccezione è il motore database RDS per Oracle, che supporta i database di replica in modalità montata. Una replica montata non accetta connessioni utente e quindi non può gestire un carico di lavoro di sola lettura. L'uso principale per le repliche montate è il disaster recovery tra regioni. Per ulteriori informazioni, consulta Utilizzo di repliche di lettura per Amazon RDS per Oracle.

Le applicazioni si connettono a una replica di lettura allo stesso modo in cui si connettono a qualsiasi istanza database. Amazon RDS replica tutti i database dall'istanza database di origine.

Repliche di lettura in una implementazione multi-AZ

È possibile configurare una replica di lettura per un'istanza database che dispone anche di una replica in standby configurata per l'elevata disponibilità in un'implementazione Multi-AZ. La replica con la replica in standby è sincrona. A differenza di una replica di lettura, una replica in standby non può gestire il traffico di lettura.

Nel seguente scenario, i client hanno accesso in lettura/scrittura a un'istanza database primaria in una zona di disponibilità (AZ). L'istanza primaria copia gli aggiornamenti in modo asincrono su una replica di lettura in una seconda zona di disponibilità e li copia anche in modo sincrono su una replica in standby in una terza zona di disponibilità. I client hanno accesso in lettura solo alla replica di lettura.



Per ulteriori informazioni sull'elevata disponibilità e sulle repliche in standby, consulta Configurazione e gestione di un'implementazione multi-AZ.

Repliche di lettura tra regioni diverse

In alcuni casi, una replica di lettura risiede in un'istanza DB diversa da quella principale. Regione AWS In questi casi, Amazon RDS configura un canale di comunicazione sicuro tra l'istanza database primaria e la replica di lettura. Amazon RDS stabilisce tutte le configurazioni AWS di sicurezza necessarie per abilitare il canale sicuro, ad esempio l'aggiunta di voci ai gruppi di sicurezza. Per informazioni sulle repliche di lettura tra regioni, consulta Creazione di una replica di lettura in un altro Regione AWS.

Le informazioni contenute in questo capitolo si applicano alla creazione di repliche di lettura di Amazon RDS nella Regione AWS stessa istanza DB di origine o in un'altra istanza. Regione AWS

Le informazioni seguenti non si applicano alla configurazione della replica con un'istanza eseguita in un'istanza Amazon EC2 o on-premise.

Differenze tra repliche di lettura per i motori DB

Poiché i motori DB di Amazon RDS implementano la replica in modo diverso, ci sono molte differenze significative delle quali dovresti essere consapevole, come mostrato nella tabella seguente.

Caratteristica o comportamento	MySQL e MariaDB	Oracle	PostgreSQL	SQL Server
Qual è il metodo di replica?	Replica logica.	Replica fisica.	Replica fisica.	Replica fisica.
Come vengono rimossi i log delle transazioni?	RDS for MySQL e RDS for MariaDB conservano tutti i log binari che non sono stati applicati.	Se un'istanza database di origine non dispone di repliche di lettura tra regioni, Amazon RDS for Oracle mantiene un minimo di due ore di log delle transazioni sull'istanza database di origine. I log vengono eliminati dall'istante database sorgente dopo due ore o dopo il periodo di tempo impostato con l'opzione, a seconda di quale risulta maggiore.	PostgreSQL include il parametro `wal_keep_segments` , che indica quanti file WAL (Write Ahead Log) vengono mantenuti per fornire dati alle repliche di lettura. Il valore del parametro specifica il numero di log da conservare.	Il file di log virtuale (VLF) del file di registro delle transazioni nella replica primaria può essere troncato quando non è più richiesto per le repliche secondarie. Il VLF può essere contrassegnato come

Caratteristica o comportamento	MySQL e MariaDB	Oracle	PostgreSQL	SQL Server
		I log vengono eliminati dalla replica di lettura dopo il periodo di tempo impostato con l'opzione solo se questi sono stati applicati correttamente al database. In alcuni casi, un'istanza database primaria potrebbe avere una o più repliche di lettura tra regioni. In questa evenienza, Amazon RDS for Oracle mantiene i log delle transazioni sull'istanza database di origine finché non vengono trasmessi e applicati a tutte le repliche di lettura tra regioni. Per informazioni sull'impostazione delle ore di conservazione dei log di archivio,		inattivo solo quando i record di log sono stati rafforzati nelle repliche. A prescindere dalla velocità dei sottosistemi del disco nella replica primaria, il log delle transazioni manterrà i VLF fino a quando la replica più lenta non è stata rafforzata.

Caratteristica o comportamento	MySQL e MariaDB	Oracle	PostgreSQL	SQL Server
		consulta Conservazione dei log redo archiviati.		
Una replica può essere resa scrivibile?	Sì. Puoi rendere scrivibile una replica di lettura MySQL o MariaDB.	No. Una replica di lettura di Oracle è una copia fisica e Oracle non consente di scrivere su una replica di lettura. Puoi promuovere la replica di lettura per renderla scrivibile. La replica di lettura promossa contiene i dati replicati fino al momento in cui è stata effettuata la richiesta di promozione.	No. Una replica di lettura di PostgreSQL è una copia fisica e PostgreSQL non permette di rendere scrivibile una replica di lettura.	No. Una replica di lettura di SQL Server è una copia fisica e inoltre non consente scritture. Puoi promuovere la replica di lettura per renderla scrivibile. La replica di lettura promossa contiene i dati replicati fino al momento in cui è stata effettuata la richiesta di promozione.

Caratteristica o comportamento	MySQL e MariaDB	Oracle	PostgreSQL	SQL Server
Possono essere eseguiti backup sulla replica?	Sì. I backup automatici e gli snapshot manuali sono supportati nelle repliche di lettura RDS per MySQL o RDS per MariaDB.	Sì. I backup automatici e gli snapshot manuali sono supportati nelle repliche di lettura RDS per Oracle.	Sì, è possibile creare uno snapshot manuale delle repliche di lettura RDS per PostgreSQL. I backup automatici per le repliche di lettura sono supportati solo per RDS per PostgreSQL 14.1 e versioni successive. Non è possibile attivare i backup automatici per le repliche di lettura PostgreSQL per le versioni precedenti alla 14.1 di RDS per PostgreSQL. Per RDS per PostgreSQL 13 e versioni precedenti, crea uno snapshot da una replica di lettura se si desidera creare un backup da tale snapshot.	No. I backup automatici e gli snapshot manuali non sono supportati nelle repliche di lettura RDS per SQL Server.

Caratteristica o comportamento	MySQL e MariaDB	Oracle	PostgreSQL	SQL Server
È possibile usare la replica parallela?	Sì. Tutte le versioni supportate di MariaDB e MySQL supportano i thread di replica parallela.	Sì. I dati dei log di Redo vengono sempre trasmessi in parallelo dal database primario a tutte le sue repliche di lettura.	No. PostgreSQL usa un unico processo per la gestione della replica.	Sì. I dati dei log di Redo vengono sempre trasmessi in parallelo dal database primario a tutte le sue repliche di lettura.
È possibile mantenere una replica in uno stato montato piuttosto che in uno stato di sola lettura?	No.	Sì. L'uso principale per le repliche montate è il disaster recovery tra regioni. Non è richiesta una licenza Active Data Guard per le repliche montate. Per ulteriori informazioni, consulta Utilizzo di repliche di lettura per Amazon RDS per Oracle.	No.	No.

Tipi di archiviazione della replica di lettura

Per impostazione predefinita, la replica di lettura viene creata con lo stesso tipo di storage dell'istanza database di origine. Tuttavia, puoi creare una replica di lettura con un tipo di storage diverso dall'istanza database di origine in base alle opzioni elencate nella tabella seguente.

Tipo di storage dell'istanza database di origine	Storage allocato all'istanza database di origine	Opzioni per il tipo di storage della replica di lettura
IOPS con provisioning	100 GiB–64 TiB	Capacità di IOPS allocata, per uso generico, magnetico
Uso generico	100 GiB–64 TiB	Capacità di IOPS allocata, per uso generico, magnetico
Uso generico	<100 GiB	Per uso generico, magnetico
Magnetico	Da 100 GiB a 6 TiB	Capacità di IOPS allocata, per uso generico, magnetico
Magnetico	<100 GiB	Per uso generico, magnetico

> **Note**
>
> Quando si aumenta lo storage allocato di una replica di lettura, deve essere di almeno il 10%. Se si prova ad aumentarlo di un valore inferiore al 10%, verrà visualizzato un errore.

Restrizioni per la creazione di una replica da una replica

Amazon RDS non supporta la replica circolare. Non puoi configurare un'istanza database perché funga da origine della replica per un'istanza database esistente. Puoi creare una nuova replica di lettura solo a partire da un'istanza database esistente. Ad esempio, se **`MySourceDBInstance`** si replica su **`ReadReplica1`**, non puoi configurare **`ReadReplica1`** affinché si replichi a sua volta su **`MySourceDBInstance`**.

Per RDS per MariaDB e RDS per MySQL e per alcune versioni di RDS per PostgreSQL, è possibile creare una replica di lettura a partire da una replica di lettura esistente. Ad esempio, puoi creare una

nuova replica di lettura **`ReadReplica2`** dalla replica esistente **`ReadReplica1`**. Nel caso di RDS per Oracle e RDS per SQL Server, non è possibile creare una replica di lettura a partire da una replica di lettura esistente.

Considerazioni su quando eliminare le repliche

Se non sono più necessarie repliche di lettura, è possibile eliminarle in modo esplicito utilizzando gli stessi meccanismi per l'eliminazione di un'istanza DB. Se elimini un'istanza DB di origine senza eliminarne le repliche di lettura nella stessa Regione AWS, ogni replica di lettura viene promossa a istanza DB autonoma. Per informazioni sulla creazione di un'istanza database, consulta Eliminazione di un'istanza database. Per informazioni sulla promozione della replica in lettura, vedere Promozione di una replica di lettura a istanza database standalone.

Se si dispone di repliche di lettura tra regioni, consulta Considerazioni relative alla replica tra regioni per informazioni correlate all'eliminazione dell'istanza database di origine per una replica di lettura tra regioni.

Creazione di una replica di lettura

È possibile creare una replica di lettura da un'istanza DB esistente utilizzando AWS Management Console, AWS CLI o l'API RDS. Per creare una replica di lettura, devi specificare `SourceDBInstanceIdentifier`, che è l'identificatore istanze DB di origine da cui desideri eseguire la replica.

Quando crei una replica di lettura, Amazon RDS acquisisce uno snapshot DB dell'istanza database di origine e avvia la replica. L'istanza DB di origine subisce una sospensione di I/O molto breve all'inizio dell'operazione di snapshot DB. La sospensione I/O dura in genere circa un secondo. Puoi evitare l'interruzione delle operazioni di I/O se l'istanza database di origine è un'implementazione Multi-AZ, perché in questo caso lo snapshot viene acquisito dall'istanza database secondaria.

Una transazione attiva a esecuzione prolungata può rallentare il processo di creazione della replica di lettura. Ti consigliamo di attendere il completamento delle transazioni a esecuzione prolungata prima di creare una replica di lettura. Se crei più repliche di lettura in parallelo dalla stessa istanza database di origine, Amazon RDS acquisisce un solo snapshot all'inizio della prima operazione di creazione.

Quando crei una replica di lettura, devi tenere presenti alcune considerazioni. Prima di tutto, devi abilitare i backup automatici nell'istanza database di origine impostando il periodo di retention dei backup su un valore diverso da zero. Questo requisito si applica anche a una replica di lettura che rappresenta l'istanza database di origine per un'altra replica di lettura. Per abilitare i backup

automatici in una replica di lettura per RDS per MySQL, crea prima di tutto la replica di lettura, quindi modificala in modo da abilitare i backup automatici.

> **Note**
>
> All'interno di un Regione AWS, consigliamo vivamente di creare tutte le repliche di lettura nello stesso cloud privato virtuale (VPC) basato su Amazon VPC come istanza DB di origine. Se crei una replica di lettura in un VPC diverso da quello dell'istanza database di origine, gli intervalli classless inter-domain routing (CIDR) possono sovrapporsi tra la replica e il sistema RDS. La sovrapposizione CIDR rende la replica instabile, influendo negativamente sulle applicazioni che si connettono. Se viene visualizzato un errore durante la creazione della replica di lettura, scegli un gruppo di sottoreti DB di destinazione diverso. Per ulteriori informazioni, consulta Uso di un'istanza database in un VPC.
> Non esiste un modo diretto per creare una replica di lettura in un'altra Account AWS utilizzando la console o. AWS CLI

Console

Per creare una replica di lettura da un'istanza database di origine

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, scegliere Databases (Database).
3. Scegli l'istanza database da usare come origine per la replica di lettura.
4. Per Actions (Operazioni), scegliere Create read replica (Crea replica di lettura).
5. Per DB instance identifier (Identificatore istanze DB) inserire un nome per la replica di lettura.
6. Scegli la configurazione per la tua istanza. Consigliamo di usare la stessa classe di istanza database o più grande e lo stesso tipo di archiviazione dell'istanza database di origine per la replica di lettura.
7. Per la Regione AWS, specifica la regione di destinazione per la replica di lettura.
8. In Archiviazione, specifica la dimensione di archiviazione allocata e l'eventuale uso della funzione di dimensionamento automatico dell'archiviazione.

   Se l'istanza database di origine non utilizza la configurazione dell'archiviazione più recente, è disponibile l'opzione Aggiorna la configurazione del file system di archiviazione. È possibile abilitare questa impostazione per aggiornare il file system di archiviazione della replica di

lettura alla configurazione preferita. Per ulteriori informazioni, consulta the section called “Aggiornamento del file system di archiviazione”.

9. In Disponibilità, scegli se creare una versione in standby della replica in un'altra zona di disponibilità per il supporto del failover per la replica.

> **Note**
>
> La creazione della replica di lettura come un'istanza database Multi-AZ non dipende dal fatto che il database di origine sia un'istanza database Multi-AZ.

10. Specifica le altre impostazioni dell'istanza database. Per informazioni su ciascuna impostazione disponibile, consulta Impostazioni per istanze database.
11. Per creare una replica di lettura crittografata, espandi Configurazione aggiuntiva e specifica le seguenti impostazioni:

    a. Scegliere Enable encryption (Abilita crittografia).

    b. Per AWS KMS key, scegli l' AWS KMS key identificatore della chiave KMS.

> **Note**
>
> L'istanza DB di origine deve essere crittografata. Per ulteriori informazioni sulla crittografia dell'istanza database di origine, consultare Crittografia delle risorse Amazon RDS.

12. Scegliere Create read replica (Crea replica di lettura).

Dopo aver creato la replica di lettura, è possibile visualizzarla nella pagina Databases (Database) della console RDS. Mostra Replica nella colonna Role (Ruolo).

AWS CLI

Per creare una replica di lettura da un'istanza DB di origine, usa il AWS CLI comando -replica. create-db-instance-read Questo esempio inoltre imposta la dimensione dell'archiviazione allocata, abilita il dimensionamento automatico dell'archiviazione e aggiorna il file system alla configurazione preferita.

È possibile specificare altre impostazioni. Per informazioni su ciascuna impostazione, consulta Impostazioni per istanze database.

Example

PerLinux, o: macOS Unix

```
aws rds create-db-instance-read-replica \
    --db-instance-identifier myreadreplica \
    --source-db-instance-identifier mydbinstance \
    --allocated-storage 100 \
    --max-allocated-storage 1000 \
    --upgrade-storage-config
```

Per Windows:

```
aws rds create-db-instance-read-replica ^
    --db-instance-identifier myreadreplica ^
    --source-db-instance-identifier mydbinstance ^
    --allocated-storage 100 ^
    --max-allocated-storage 1000 ^
    --upgrade-storage-config
```

API RDS

Per creare una replica di lettura da un'istanza database di MySQL, MariaDB, Oracle, PostgreSQL o SQL Server di origine, richiama l'operazione Amazon RDS `CreateDBInstanceReadReplica` dell'API con i seguenti parametri richiesti:

- `DBInstanceIdentifier`
- `SourceDBInstanceIdentifier`

Promozione di una replica di lettura a istanza database standalone

È possibile promuovere una replica di lettura in un'istanza database autonoma. Se un'istanza database di origine ha diverse repliche di lettura, la promozione di una delle repliche di lettura a istanza database non ha alcun effetto sulle altre repliche.

Quando promuovi una replica di lettura, RDS riavvia l'istanza DB prima di renderla disponibile. Il processo di promozione può richiedere alcuni minuti per il completamento, che possono aumentare a seconda delle dimensioni della replica di lettura.

Casi d'uso per promuovere una replica di lettura

Potresti voler promuovere una replica di lettura a un'istanza DB autonoma per uno dei seguenti motivi:

- Implementazione del ripristino dagli errori – Puoi usare la promozione delle repliche di lettura come schema di ripristino dei dati in caso di errore dell'istanza database primaria. Questo approccio si integra con la replica sincrona, il rilevamento automatico degli errori e il failover.

Se sei al corrente delle ramificazioni e delle limitazioni della replica asincrona, ma vuoi comunque usare la promozione delle repliche di lettura per il ripristino dei dati, puoi farlo. A questo scopo, crea prima di tutto una replica di lettura e quindi monitora l'istanza database primaria per individuare eventuali errori. In caso di errore, sono necessarie le operazioni seguenti:

1. Promuovi la replica di lettura.
2. Indirizza il traffico di database all'istanza database promossa.
3. Crea una replica di lettura sostitutiva con l'istanza database promossa come origine.

- Aggiornamento della configurazione di archiviazione: se l'istanza database di origine non si trova nella configurazione dell'archiviazione preferita, è possibile creare una replica di lettura dell'istanza e aggiornare la configurazione del file system di archiviazione. Questa opzione esegue la migrazione del file system della replica di lettura alla configurazione preferita. È possibile promuovere la replica di lettura a istanza autonoma.

  È possibile utilizzare questa opzione per superare le limitazioni di dimensionamento relative all'archiviazione e alle dimensioni dei file per i file system a 32 bit precedenti. Per ulteriori informazioni, consulta the section called “Aggiornamento del file system di archiviazione”.

  Questa opzione è disponibile solo se l'istanza database di origine non utilizza la configurazione di storage più recente o se stai modificando la classe dell'istanza database nell'ambito della stessa richiesta.

- Partizionamento – Il partizionamento include l'architettura a "zero condivisione" ed essenzialmente comporta la suddivisione di database di grandi dimensioni in diversi database più piccoli. Un metodo comune per suddividere un database consiste nel dividere le tabelle che non sono unite nella stessa query in host diversi. Un altro metodo consiste nel duplicare una tabella tra più host e quindi usare un algoritmo di hashing per determinare quale host riceverà un determinato aggiornamento. Puoi creare repliche di lettura corrispondenti a ognuno degli shard (database più piccoli) e promuoverle quando decidi di convertirle in shard standalone. Puoi quindi separare lo spazio delle chiavi (se stai suddividendo le righe) o la distribuzione delle tabelle per ognuno degli shard, a seconda dei requisiti.
- Esecuzione di operazioni DDL (solo MySQL e MariaDB) – Le operazioni DDL, come la creazione o la ricompilazione di indici, possono richiedere tempo e causano un notevole rallentamento delle prestazioni nell'istanza database. Puoi eseguire queste operazioni su una replica di lettura MySQL o MariaDB quando la replica di lettura è sincronizzata con l'istanza database primaria corrispondente. Puoi quindi promuovere la replica di lettura e indicare alle applicazioni di usare l'istanza promossa.

> **Note**
>
> Se la replica di lettura è un'istanza DB RDS per Oracle, puoi eseguire uno switchover anziché una promozione. In uno switchover, l'istanza DB di origine diventa la nuova replica e la replica diventa la nuova istanza DB di origine. Per ulteriori informazioni, consulta Esecuzione di uno switchover Oracle Data Guard.

Caratteristiche di una replica di lettura promossa

Dopo aver promosso la replica di lettura, questa cessa di funzionare come replica di lettura e diventa un'istanza DB autonoma. La nuova istanza DB autonoma presenta le seguenti caratteristiche:

- L'istanza DB autonoma mantiene il gruppo di opzioni e il gruppo di parametri della replica di lettura precedente alla promozione.
- È possibile creare repliche di lettura dall'istanza DB autonoma ed eseguire operazioni di ripristino. point-in-time
- Non è possibile utilizzare l'istanza DB come destinazione di replica perché non è più una replica di lettura.

Prerequisiti per promuovere una replica di lettura

Prima di promuovere una replica di lettura, procedi come segue:

- Rivedi la tua strategia di backup:
  - Ti consigliamo di abilitare i backup e completare almeno un backup. La durata del backup è una funzione del numero di modifiche apportate al database dal backup precedente.
  - Se hai abilitato i backup nella replica di lettura, configura la finestra dei backup automatici in modo che i backup giornalieri non interferiscano con la promozione della replica di lettura.
  - Assicurati che la replica di lettura non abbia lo `backing-up` stato. Non è possibile promuovere una replica di lettura quando si trova in questo stato.
- Impedisci la scrittura di qualsiasi transazione sull'istanza DB principale, quindi attendi che RDS applichi tutti gli aggiornamenti alla replica di lettura.

  Gli aggiornamenti del database vengono eseguiti nella replica di lettura dopo essere stati completati nell'istanza database primaria. Il ritardo di replica può variare in modo significativo.

Utilizzare il parametro `Replica Lag` per determinare quando sono stati applicati tutti gli aggiornamenti alla replica di lettura.

- (Solo MySQL e MariaDB) Per apportare modifiche a una replica di lettura MySQL o MariaDB prima di promuoverla, imposta il parametro su nel gruppo di parametri DB per la replica di lettura. `read_only` `0` È quindi possibile eseguire tutte le operazioni DDL necessarie, come la creazione di indici, nella replica di lettura. Le operazioni eseguite nella replica di lettura non influiscono sulle prestazioni dell'istanza database primaria.

Promuovere una replica di lettura: passaggi di base

Le fasi seguenti descrivono il processo generale per la promozione di una replica di lettura a istanza database:

1. Promuovi la replica di lettura utilizzando l'opzione Promote sulla console Amazon RDS, il AWS CLI comando o l'`promote-read-replica`operazione dell'API `PromoteReadReplica`Amazon RDS.

   > **Note**
   >
   > Per il completamento del processo di promozione sono necessari alcuni minuti. Quando promuovi una replica di lettura, RDS interrompe la replica e riavvia la replica di lettura. Al termine del riavvio, la replica di lettura è disponibile come nuova istanza database.

2. (Facoltativo) Modificare la nuova istanza database in modo da impostarla come implementazione Multi-AZ. Per ulteriori informazioni, consulta Modifica di un'istanza database Amazon RDS e Configurazione e gestione di un'implementazione multi-AZ.

Console

Per promuovere una replica di lettura in un'istanza database autonoma

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nella console Amazon RDS scegliere Databases (Database).

   Verrà visualizzato il riquadro Databases (Database). Ogni replica di lettura mostra la Replica nella colonna Role (Ruolo).
3. Scegliere la replica di lettura che si desidera promuovere.

4. In Actions (Operazioni), selezionare Promote (Promuovi).
5. Nella pagina Promuovi replica di lettura immettere il periodo di retention dei backup e la finestra di backup per la nuova istanza database promossa.
6. Dopo aver selezionato tutte le impostazioni desiderate, scegliere Continue (Continua).
7. Nella pagina di conferma scegliere Promote Read Replica (Promuovi replica di lettura).

AWS CLI

Per promuovere una replica di lettura a un'istanza DB autonoma, usa il comando. AWS CLI `promote-read-replica`

Example

PerLinux, omacOS: Unix

```
aws rds promote-read-replica \
    --db-instance-identifier myreadreplica
```

Per Windows:

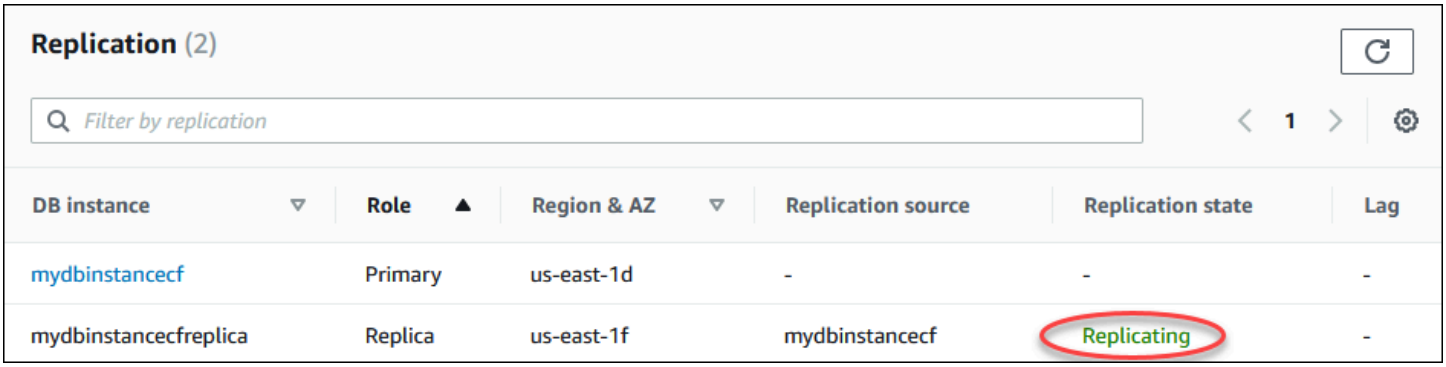
```
aws rds promote-read-replica ^
    --db-instance-identifier myreadreplica
```

API RDS

Per promuovere una replica di lettura a istanza database autonoma, richiama l'operazione `PromoteReadReplica` dell'API Amazon RDS con il parametro `DBInstanceIdentifier` richiesto.

Monitoraggio della replica di lettura

Puoi monitorare lo stato di una replica di lettura in diversi modi. La console Amazon RDS mostra lo stato di una replica di lettura nella sezione Replication (Replica) della scheda Connectivity & security (Connettività e sicurezza) nelle informazioni di dettaglio della replica di lettura. Per visualizzare i dettagli per una replica di lettura, scegli il nome della replica di lettura nell'elenco di istanze database nella console di Amazon RDS.



Puoi anche vedere lo stato di una replica di lettura utilizzando il AWS CLI `describe-db-instances` comando o l'operazione dell'API `DescribeDBInstances` Amazon RDS.

Lo stato di una replica di lettura può essere uno dei seguenti:

- replicating (replica in corso) – La replica di lettura sta eseguendo correttamente la replica.
- replica danneggiata (solo SQL Server e PostgreSQL) – Le repliche ricevono dati dall'istanza primaria, ma uno o più database potrebbe non ricevere aggiornamenti. Ciò può verificarsi, ad esempio, quando una replica sta completando l'impostazione dei database appena creati. Può verificarsi anche quando vengono apportate modifiche a oggetti di grandi dimensioni o DDL non supportate nell'ambiente blu di un'implementazione blu/verde.

  Lo stato non passa da `replication degraded` a `error`, a meno che non si verifichi un errore durante lo stato danneggiato.
- error (errore) – Si è verificato un errore di replica. Controlla il campo Replication Error (Errore di replica) nella console Amazon RDS o il log degli eventi per determinare esattamente l'errore. Per ulteriori informazioni sulla risoluzione dei problemi causati da un errore di replica, consulta Risoluzione dei problemi relativi a una replica di lettura MySQL.
- terminated (terminata) (solo MariaDB, MySQL o PostgreSQL) – La replica è terminata. Questa situazione si verifica se la replica viene arrestata per più di 30 giorni consecutivi, manualmente o a causa di un errore di replica. In questo caso, Amazon RDS termina la replica tra l'istanza database primaria e tutte le repliche di lettura. Amazon RDS si comporta così per evitare requisiti di archiviazione maggiori sull'istanza database di origine e tempi di failover prolungati.

  L'interruzione della replica può influire sullo storage, perché può causare l'aumento delle dimensioni e del numero dei log a causa del volume elevato di messaggi di errore scritti nel log. L'interruzione della replica può anche influire sul ripristino dagli errori a causa del tempo necessario ad Amazon RDS per mantenere ed elaborare il numero elevato di log durante il ripristino.

- terminated (terminata) (solo Oracle) – La replica è terminata. Questa situazione si verifica se la replica viene arrestata per più di 8 ore a causa della mancanza di spazio di archiviazione nella replica di lettura. In questo caso, Amazon RDS termina la replica tra l'istanza database primaria e tutte le repliche di lettura. Questo è uno stato terminale e la replica di lettura deve essere ricreata.
- stopped (arrestata) (solo MySQL o MariaDB) – La replica è stata interrotta a causa di una richiesta avviata dal cliente.
- replication stop point set (punto di arresto replica impostato) (solo MySQL) – È stato impostato un punto di arresto avviato dal cliente tramite la stored procedure mysql.rds_start_replication_until e la replica è in corso.
- replication stop point reached (punto di arresto replica raggiunto) (solo MySQL) – È stato impostato un punto di arresto avviato dal cliente tramite la stored procedure mysql.rds_start_replication_until e la replica è stata arrestata perché è stato raggiunto il punto di arresto.

È possibile visualizzare dove viene replicata un'istanza database e, in caso affermativo, verificarne lo stato di replica. Nella pagina Databases (Database) della console RDS viene visualizzato Primary (Primario) nella colonna Role (Ruolo). Scegliere il nome dell'istanza database. Nella relativa pagina dei dettagli, nella scheda Connectivity & security (Connettività e sicurezza), lo stato di replica si trova in Replica.

Monitoraggio del ritardo di replica

Puoi monitorare il ritardo di replica in Amazon CloudWatch visualizzando la metrica Amazon RDS `ReplicaLag`.

Per MariaDB e MySQL, il parametro `ReplicaLag` indica il valore del campo `Seconds_Behind_Master` del comando `SHOW REPLICA STATUS`. Le cause comuni del ritardo di replica per MySQL e MariaDB sono le seguenti:

- Interruzione della connessione di rete.
- Scrittura in tabelle con indici in una replica di lettura. Se il parametro `read_only` non è impostato su 0 nella replica di lettura, la replica può essere interrotta.
- Uso di un motore di storage non transazionale come MyISAM. La replica è supportata solo per il motore di storage InnoDB in MySQL e per il motore di storage XtraDB in MariaDB.

> **Note**
>
> Le versioni precedenti di MariaDB e MySQL utilizzavano `SHOW SLAVE STATUS` anziché `SHOW REPLICA STATUS`. Se si utilizza una versione di MariaDB precedente alla 10.5 o una versione di MySQL precedente alla 8.0.23, utilizzare `SHOW SLAVE STATUS`.

Quando il parametro `ReplicaLag` è 0, la replica ha raggiunto l'istanza del database primaria. Se il parametro `ReplicaLag` restituisce `-1`, la replica non è attualmente attiva. `ReplicaLag = -1` equivale a `Seconds_Behind_Master = NULL`.

Per Oracle, il parametro `ReplicaLag` è la somma del valore `Apply Lag` e della differenza tra il tempo corrente e il valore `DATUM_TIME` di Apply Lag. Il valore `DATUM_TIME` indica il tempo in cui la replica di lettura ha ricevuto per l'ultima volta i dati dall'istanza database di origine. Per ulteriori informazioni, consultare V$DATAGUARD_STATS nella documentazione di Oracle.

Per SQL Server, il parametro `ReplicaLag` è il ritardo massimo dei database che sono rimasti indietro, in secondi. Ad esempio, se sono disponibili due database con, rispettivamente, un ritardo di 5 secondi e 10 secondi, `ReplicaLag` è di 10 secondi. Il parametro `ReplicaLag` restituisce il valore della query seguente.

```
SELECT MAX(secondary_lag_seconds) max_lag FROM sys.dm_hadr_database_replica_states;
```

Per ulteriori informazioni, consulta secondary_lag_seconds nella documentazione di Microsoft.

`ReplicaLag` restituisce `-1` se RDS non è in grado di determinare il ritardo, ad esempio durante la configurazione della replica o quando lo stato della replica di lettura è `error`.

> **Note**
>
> I nuovi database non vengono inclusi nel calcolo del ritardo finché non sono accessibili nella replica di lettura.

Per PostgreSQL, il parametro `ReplicaLag` restituisce il valore della query seguente.

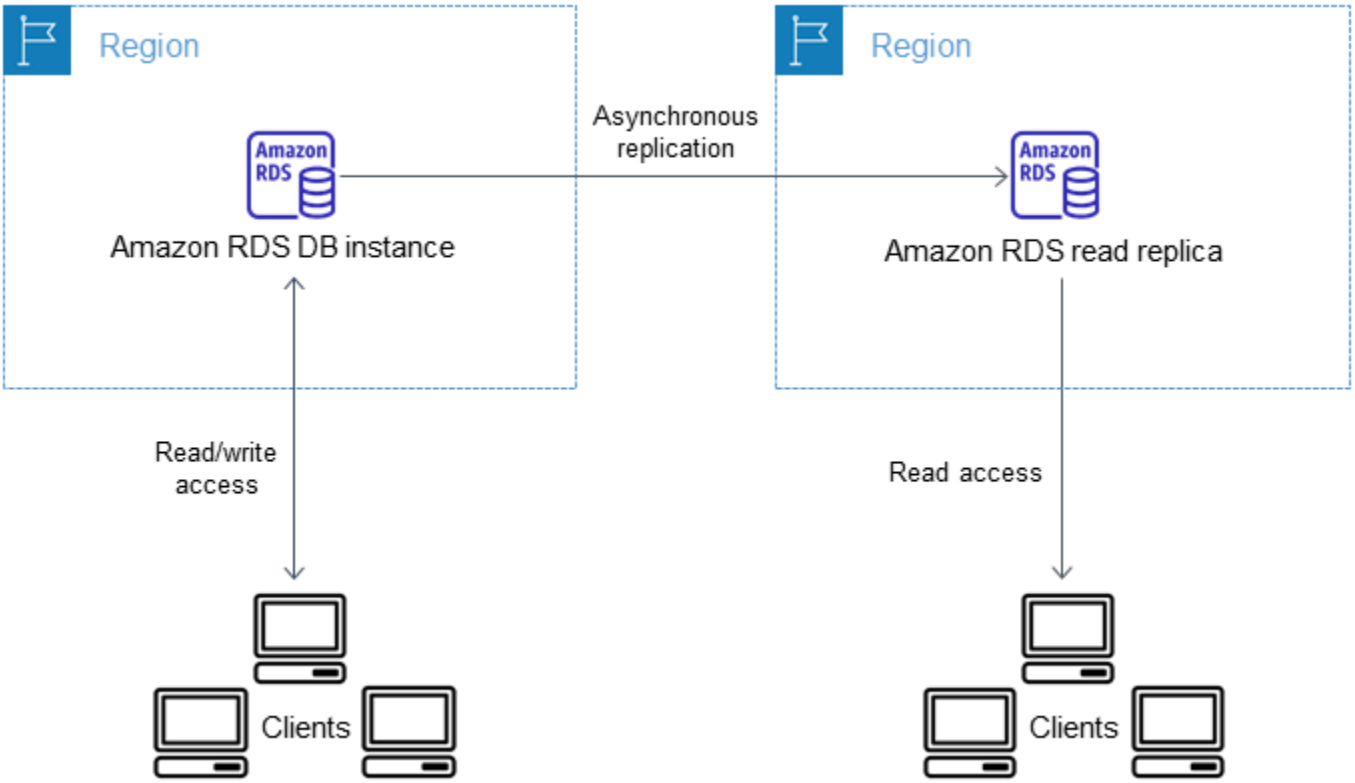
```
SELECT extract(epoch from now() - pg_last_xact_replay_timestamp()) AS reader_lag
```

PostgreSQL versioni 9.5.2 e successive usano slot di replica fisici per gestire la conservazione dei dati Write Ahead Log (WAL) nell'istanza di origine. Per ogni istanza di replica di lettura tra regioni, Amazon RDS crea uno slot di replica fisica e lo associa all'istanza. Due CloudWatch metriche di Amazon mostrano quanto sia indietro rispetto alla replica con maggior ritardo in termini di dati WAL ricevuti e di quanto spazio di archiviazione viene utilizzato per i dati WAL. `Oldest Replication Slot Lag` `Transaction Logs Disk Usage` Il valore `Transaction Logs Disk Usage` può aumentare in modo considerevole quando una replica di lettura tra regioni è in notevole ritardo.

Per ulteriori informazioni sul monitoraggio di un'istanza DB con, consulta. CloudWatch Monitoraggio dei parametri di Amazon RDS con Amazon CloudWatch

Creazione di una replica di lettura in un altro Regione AWS

Con Amazon RDS, puoi creare una replica di lettura in un'istanza DB diversa Regione AWS da quella di origine.



È possibile creare una replica di lettura in un altro Regione AWS modo per effettuare le seguenti operazioni:

- Migliorare le funzionalità di disaster recovery.
- Adattate le operazioni di lettura in modo che Regione AWS siano sempre più vicine ai vostri utenti.

- Semplifica la migrazione da un data center Regione AWS a un data center in un altro Regione AWS.

La creazione di una replica di lettura in un'istanza diversa Regione AWS da quella di origine è simile alla creazione di una replica nella stessa. Regione AWSÈ possibile utilizzare AWS Management Console, eseguire il `create-db-instance-read-replica`comando o chiamare l'operazione `CreateDBInstanceReadReplica`API.

> **Note**
>
> Per creare una replica di lettura crittografata in un'istanza DB diversa Regione AWS da quella di origine, l'istanza DB di origine deve essere crittografata.

Disponibilità di regioni e versioni

Il supporto varia a seconda delle versioni specifiche di ciascun motore di database e a seconda delle Regioni AWS. Per ulteriori informazioni sulla disponibilità di versioni e regioni con la replica tra regioni, consulta Regioni e motori DB supportati per repliche di lettura interregionali in Amazon RDS.

Creazione di una replica di lettura tra regioni

Nelle procedure seguenti viene mostrato come creare una replica di lettura da un'istanza database MariaDB, Microsoft SQL Server, MySQL, Oracle o PostgreSQL di origine in una Regione AWS diversa.

Console

È possibile creare una replica di lettura Regioni AWS utilizzando. AWS Management Console

Per creare una replica di lettura tramite Regioni AWS la console

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, scegli Databases (Database).
3. Scegli l'istanza database MariaDB, Microsoft SQL Server, MySQL, Oracle o PostgreSQL da usare come origine la replica di lettura.
4. Per Actions (Operazioni), scegliere Create read replica (Crea replica di lettura).
5. Per DB instance identifier (Identificatore istanze DB) inserire un nome per la replica di lettura.

6. Seleziona la regione di destinazione.
7. Scegliere le specifiche dell'istanza che si vuole usare. Consigliamo di usare almeno la stessa classe di istanza database e lo stesso tipo di archiviazione per la replica di lettura.
8. Per creare una replica di lettura crittografata in un'altra: Regione AWS

   a. Scegliere Enable encryption (Abilita crittografia).

   b. Per AWS KMS key, scegli l' AWS KMS key identificatore della chiave KMS nella destinazione. Regione AWS

   > **Note**
   >
   > Per creare una replica di lettura crittografata, l'istanza database di origine deve essere crittografata. Per ulteriori informazioni sulla crittografia dell'istanza database di origine, consultare Crittografia delle risorse Amazon RDS.

9. Scegli altre opzioni, ad esempio il dimensionamento automatico dello storage.
10. Scegliere Create read replica (Crea replica di lettura).

AWS CLI

Per creare una replica di lettura da un'istanza database MySQL, Microsoft SQL Server, MariaDB, Oracle o PostgreSQL di origine in una Regione AWS diversa, puoi utilizzare il comando `create-db-instance-read-replica`. In questo caso, si utilizza `create-db-instance-read-replica`dalla posizione in Regione AWS cui si desidera la replica di lettura (regione di destinazione) e si specifica l'Amazon Resource Name (ARN) per l'istanza DB di origine. Un ARN identifica in modo univoco una risorsa creata in Amazon Web Services.

Ad esempio, se l'istanza database di origine si trova nella regione US East (N. Virginia), l'aspetto dell'ARN è simile al questo esempio:

```
arn:aws:rds:us-east-1:123456789012:db:mydbinstance
```

Per informazioni sugli ARN, consulta Utilizzo di Amazon Resource Name (ARN) in Amazon RDS.

Per creare una replica di lettura in un'istanza DB diversa Regione AWS da quella di origine, puoi utilizzare il AWS CLI `create-db-instance-read-replica`comando dalla destinazione. Regione AWS Per creare una replica di lettura in un'altra Regione AWS, i seguenti parametri sono obbligatori:

- `--region`— La destinazione Regione AWS in cui viene creata la replica di lettura.
- `--source-db-instance-identifier`– L'identificatore dell'istanza database per l'istanza database di origine. L'identificatore deve usare il formato ARN per la Regione AWS di origine.
- `--db-instance-identifier`: l'identificatore per la replica di lettura nella Regione AWS di destinazione.

Example di una replica di lettura tra regioni

Il seguente codice crea una replica di lettura nella regione Stati Uniti occidentali (Oregon) da un'istanza database di origine nella regione US East (N. Virginia).

Per LinuxmacOS, oUnix:

```
aws rds create-db-instance-read-replica \
    --db-instance-identifier myreadreplica \
    --region us-west-2 \
    --source-db-instance-identifier arn:aws:rds:us-east-1:123456789012:db:mydbinstance
```

Per Windows:

```
aws rds create-db-instance-read-replica ^
    --db-instance-identifier myreadreplica ^
    --region us-west-2 ^
    --source-db-instance-identifier arn:aws:rds:us-east-1:123456789012:db:mydbinstance
```

I seguenti parametri sono inoltre obbligatori per creare una replica di lettura crittografata in un'altra Regione AWS:

- `--kms-key-id`— L' AWS KMS key identificatore della chiave KMS da utilizzare per crittografare la replica letta nella destinazione. Regione AWS

Example di una replica di lettura tra regioni crittografate

Il seguente codice crea una replica di lettura crittografata nella regione Stati Uniti occidentali (Oregon) da un'istanza database di origine nella regione US East (N. Virginia).

PerLinux, o: macOS Unix

```
aws rds create-db-instance-read-replica \
    --db-instance-identifier myreadreplica \
```

```
    --region us-west-2 \
    --source-db-instance-identifier arn:aws:rds:us-east-1:123456789012:db:mydbinstance
 \
    --kms-key-id my-us-west-2-key
```

Per Windows:

```
aws rds create-db-instance-read-replica ^
    --db-instance-identifier myreadreplica ^
    --region us-west-2 ^
    --source-db-instance-identifier arn:aws:rds:us-east-1:123456789012:db:mydbinstance
 ^
    --kms-key-id my-us-west-2-key
```

L'`--source-region`opzione è obbligatoria quando si crea una replica di lettura crittografata tra le regioni AWS GovCloud (Stati Uniti orientali) e AWS GovCloud (Stati Uniti occidentali). Per `--source-region`, specificare la Regione AWS dell'istanza database di origine.

Se non si specifica `--source-region`, è necessario specificare un valore per `--pre-signed-url`. Un URL prefirmato è un URL che contiene una richiesta firmata Signature Versione 4 per il comando `create-db-instance-read-replica` chiamato nella Regione AWS di origine. Per ulteriori informazioni sull'*`pre-signed-url`*opzione, consulta create-db-instance-read-replica nel Command Reference.AWS CLI

API RDS

Per creare una replica di lettura da un'istanza DB di origine MySQL, Microsoft SQL Server, MariaDB, Oracle o PostgreSQL in un'altra istanza, puoi chiamare l'operazione API Amazon RDS CreateDB. Regione AWS InstanceReadReplica In questo caso, chiami CreateDB InstanceReadReplica dal punto in Regione AWS cui desideri la replica di lettura (regione di destinazione) e specifichi l'Amazon Resource Name (ARN) per l'istanza DB di origine. Un ARN identifica in modo univoco una risorsa creata in Amazon Web Services.

Per creare una replica di lettura crittografata in un'istanza DB diversa Regione AWS da quella di origine, puoi utilizzare l'`CreateDBInstanceReadReplica`operazione dell'API Amazon RDS dalla destinazione. Regione AWS Per creare una replica di lettura crittografata in un'altra Regione AWS, devi specificare un valore per. `PreSignedURL` `PreSignedURL`deve contenere una richiesta per l'`CreateDBInstanceReadReplica`operazione di richiamo all'origine in Regione

AWS cui viene creata la replica di lettura. Per ulteriori informazioni su `PreSignedUrl`, consulta `CreateDBInstanceReadReplica`.

Ad esempio, se l'istanza database di origine si trova nella regione US East (N. Virginia), l'ARN è simile al seguente.

```
arn:aws:rds:us-east-1:123456789012:db:mydbinstance
```

Per informazioni sugli ARN, consulta Utilizzo di Amazon Resource Name (ARN) in Amazon RDS.

Example

```
https://us-west-2.rds.amazonaws.com/
    ?Action=CreateDBInstanceReadReplica
    &KmsKeyId=my-us-east-1-key
    &PreSignedUrl=https%253A%252F%252Frds.us-west-2.amazonaws.com%252F
         %253FAction%253DCreateDBInstanceReadReplica
         %2526DestinationRegion%253Dus-east-1
         %2526KmsKeyId%253Dmy-us-east-1-key
         %2526SourceDBInstanceIdentifier%253Darn%25253Aaws%25253Ards%25253Aus-
west-2%123456789012%25253Adb%25253Amydbinstance
         %2526SignatureMethod%253DHmacSHA256
         %2526SignatureVersion%253D4%2526SourceDBInstanceIdentifier%253Darn%25253Aaws
%25253Ards%25253Aus-west-2%25253A123456789012%25253Ainstance%25253Amydbinstance
         %2526Version%253D2014-10-31
         %2526X-Amz-Algorithm%253DAWS4-HMAC-SHA256
         %2526X-Amz-Credential%253DAKIADQKE4SARGYLE%252F20161117%252Fus-west-2%252Frds
%252Faws4_request
         %2526X-Amz-Date%253D20161117T215409Z
         %2526X-Amz-Expires%253D3600
         %2526X-Amz-SignedHeaders%253Dcontent-type%253Bhost%253Buser-agent%253Bx-amz-
content-sha256%253Bx-amz-date
         %2526X-Amz-Signature
%253D255a0f17b4e717d3b67fad163c3ec26573b882c03a65523522cf890a67fca613
    &DBInstanceIdentifier=myreadreplica
    &SourceDBInstanceIdentifier=&region-arn;rds:us-east-1:123456789012:db:mydbinstance
    &Version=2012-01-15
    &SignatureVersion=2
    &SignatureMethod=HmacSHA256
    &Timestamp=2012-01-20T22%3A06%3A23.624Z
    &AWSAccessKeyId=<&AWS; Access Key ID>
    &Signature=<Signature>
```

Processo usato da Amazon RDS per eseguire la replica tra regioni

Amazon RDS usa il processo seguente per creare una replica di lettura tra regioni. A seconda del soggetto Regioni AWS coinvolto e della quantità di dati presenti nei database, il completamento di questo processo può richiedere ore. Puoi usare queste informazioni per determinare lo stato di avanzamento del processo quando crei una replica di lettura tra regioni:

1. Amazon RDS avvia la configurazione dell'istanza database di origine come origine della replica e imposta lo stato su modifying (modifica in corso).
2. Amazon RDS inizia a configurare la replica di lettura specificata nella destinazione Regione AWS e imposta lo stato su creazione.
3. Amazon RDS crea uno snapshot DB automatico dell'istanza database di origine nella Regione AWS di origine. Il formato del nome dello snapshot DB è `rds:<InstanceID>-<timestamp>`, dove `<InstanceID>` è l'identificatore dell'istanza di origine e `<timestamp>` corrisponde alla data e all'ora di avvio della copia. Ad esempio, `rds:mysourceinstance-2013-11-14-09-24` è stato creato dall'istanza `mysourceinstance` il `2013-11-14-09-24`. Durante la creazione di uno snapshot DB automatico, lo stato dell'istanza database di origine resta modifying (modifica in corso), lo stato della replica di lettura resta creating (creazione in corso) e lo stato dello snapshot DB è creating (creazione in corso). La colonna dello stato di avanzamento della pagina dello snapshot DB nella console indica l'avanzamento della creazione dello snapshot DB. Al termine della creazione dello snapshot DB, lo stato dello snapshot DB e quello dell'istanza database di origine sono entrambi impostati su available (disponibile).
4. Amazon RDS avvia una copia dello snapshot tra regioni per il trasferimento iniziale dei dati. La copia dello snapshot viene elencata come istantanea automatica nella destinazione Regione AWS con lo stato di creazione. La copia ha lo stesso nome dello snapshot DB di origine. La colonna dello stato di avanzamento della visualizzazione dello snapshot DB indica l'avanzamento della copia. Al termine della copia, lo stato della copia dello snapshot DB è impostato su available (disponibile).
5. Amazon RDS usa quindi lo snapshot DB copiato per il caricamento dei dati iniziale nella replica di lettura. Durante questa fase, la replica di lettura è inclusa nell'elenco delle istanze database nella destinazione, con stato creating (creazione in corso). Al termine del caricamento, lo stato della replica di lettura è impostato su available (disponibile) e la copia dello snapshot DB viene eliminata.
6. Quando una replica di lettura raggiunge lo stato disponibile, Amazon RDS avvia la replica delle modifiche apportate all'istanza di origine dall'avvio dell'operazione di creazione della replica di lettura. Durante questa fase, il ritardo di replica per la replica di lettura è maggiore di 0.

Per ulteriori informazioni sui ritardi della replica, consultare Monitoraggio della replica di lettura.

Considerazioni relative alla replica tra regioni

Tutte le considerazioni relative all'esecuzione della replica all'interno di un Regione AWS si applicano alla replica tra regioni. Alla replica tra Regioni AWS si applicano anche le considerazioni aggiuntive seguenti:

- Un'istanza database di origine può avere repliche di lettura tra regioni in più Regioni AWS.
- È possibile eseguire la replica tra le regioni GovCloud (Stati Uniti orientali) e GovCloud (Stati Uniti occidentali), ma non all'interno o all'esterno (Stati Uniti). GovCloud
- Per le istanze database Microsoft SQL Server, Oracle e PostgreSQL puoi creare una replica di lettura Amazon RDS tra regioni solo da un'istanza database Amazon RDS di origine che non sia una replica di lettura di un'altra istanza database Amazon RDS. Queste limitazioni non si applicano alle istanze database MariaDB e MySQL.
- Puoi aspettarti un livello di ritardo più elevato per qualsiasi replica di lettura che si trova in un'istanza diversa Regione AWS da quella di origine. Questo ritardo è dovuto alla maggiore lunghezza dei percorsi di rete che collegano i data center regionali.
- Per le repliche di lettura tra regioni, qualsiasi comando di creazione di repliche di lettura specificato dal parametro `--db-subnet-group-name` deve specificare un gruppo di sottoreti database dello stesso VPC.
- A causa dei limiti del numero di voci delle liste di controllo degli accessi (ACL) per il VPC di origine, non è possibile garantire più di cinque istanze di replica di lettura tra regioni.
- Nella maggior parte dei casi, la replica di lettura utilizza il gruppo di parametri del database di default per il motore di database specificato.

  Per i motori MySQL e Oracle DB, è possibile specificare un gruppo di parametri personalizzato per la replica di lettura nell'opzione `--db-parameter-group-name` del comando. AWS CLI `create-db-instance-read-replica` Non è possibile specificare un gruppo di parametri personalizzato quando si utilizza la AWS Management Console.
- La replica di lettura utilizza il gruppo di sicurezza predefinito.
- Per le istanze database MariaDB, Microsoft SQL Server, MySQL e Oracle, quando l'origine per una replica di lettura tra regioni viene eliminata, la replica di lettura viene promossa.

- Per le istanze database PostgreSQL, quando l'istanza database di origine di una replica di lettura tra regioni viene eliminata, la replica di lettura viene impostata su `terminated`. La replica di lettura non viene promossa.

  Sarà necessario promuovere manualmente la replica di lettura o eliminarla.

Richiesta di una replica di lettura tra regioni

Per comunicare con la regione di origine per richiedere la creazione di una replica di lettura tra regioni, il richiedente (ruolo IAM o utente IAM) deve avere accesso all'istanza database di origine e alla regione di origine.

Alcune condizioni nella policy IAM del richiedente possono causare l'esito negativo della richiesta. Gli esempi seguenti presuppongono che l'istanza database di origine sia in Stati Uniti orientali (Ohio) e la replica di lettura sia creata in US East (N. Virginia). Questi esempi mostrano le condizioni nella policy IAM del richiedente che causano l'esito negativo della richiesta:

- La policy del richiedente ha una condizione per `aws:RequestedRegion`.

```
...
"Effect": "Allow",
"Action": "rds:CreateDBInstanceReadReplica",
"Resource": "*",
"Condition": {
    "StringEquals": {
        "aws:RequestedRegion": "us-east-1"
    }
}
```

  La richiesta ha esito negativo perché la policy non consente l'accesso alla regione di origine. Perché una richiesta sia completata correttamente, specifica sia le regioni di origine che quelle di destinazione.

```
...
"Effect": "Allow",
"Action": "rds:CreateDBInstanceReadReplica",
"Resource": "*",
"Condition": {
    "StringEquals": {
        "aws:RequestedRegion": [
```

```
            "us-east-1",
            "us-east-2"
        ]
    }
}
```

- La policy del richiedente non consente l'accesso all'istanza database di origine.

```
...
"Effect": "Allow",
"Action": "rds:CreateDBInstanceReadReplica",
"Resource": "arn:aws:rds:us-east-1:123456789012:db:myreadreplica"
...
```

  Per una richiesta riuscita, specificare sia l'istanza di origine che la replica.

```
...
"Effect": "Allow",
"Action": "rds:CreateDBInstanceReadReplica",
"Resource": [
    "arn:aws:rds:us-east-1:123456789012:db:myreadreplica",
    "arn:aws:rds:us-east-2:123456789012:db:mydbinstance"
]
...
```

- La policy del richiedente rifiuta `aws:ViaAWSService`.

```
...
"Effect": "Allow",
"Action": "rds:CreateDBInstanceReadReplica",
"Resource": "*",
"Condition": {
    "Bool": {"aws:ViaAWSService": "false"}
}
```

  La comunicazione con la regione di origine viene effettuata da RDS per conto del richiedente. Per una richiesta andata a buon fine, non rifiutate le chiamate effettuate dai servizi. AWS

- La policy del richiedente ha una condizione per `aws:SourceVpc` o `aws:SourceVpce`.

  Queste richieste potrebbero non riuscire perché quando RDS effettua la chiamata alla regione remota, non proviene dall'endpoint VPC o dal VPC specificato.

Se è necessario utilizzare una delle condizioni precedenti che causerebbero un errore di una richiesta, è possibile includere una seconda istruzione con `aws:CalledVia` nella policy in modo che la richiesta abbia esito positivo. Ad esempio, è possibile utilizzare `aws:CalledVia` con `aws:SourceVpce` come riportato di seguito:

```
...
"Effect": "Allow",
"Action": "rds:CreateDBInstanceReadReplica",
"Resource": "*",
"Condition": {
    "Condition" : {
        "ForAnyValue:StringEquals" : {
          "aws:SourceVpce": "vpce-1a2b3c4d"
        }
     }
},
{
    "Effect": "Allow",
    "Action": [
        "rds:CreateDBInstanceReadReplica"
    ],
    "Resource": "*",
    "Condition": {
        "ForAnyValue:StringEquals": {
            "aws:CalledVia": [
                "rds.amazonaws.com"
            ]
        }
    }
}
```

Per ulteriori informazioni, consulta Policy e autorizzazioni in IAM nella Guida per l'utente di IAM.

Autorizzazione della replica di lettura

Dopo che una richiesta di creazione di replica di lettura database tra regioni restituisce `success`, RDS avvia la creazione della replica in background. Viene creata un'autorizzazione per RDS per accedere all'istanza database di origine. Questa autorizzazione collega l'istanza del database di origine alla replica di lettura e consente a RDS di copiare solo la replica di lettura specificata.

L'autorizzazione è verificata da RDS utilizzando l'autorizzazione `rds:CrossRegionCommunication` nel ruolo IAM collegato al servizio. Se la replica è autorizzata, RDS comunica con la regione di origine e completa la creazione della replica.

RDS non ha accesso alle istanze database non autorizzate in precedenza da una richiesta `CreateDBInstanceReadReplica`. L'autorizzazione viene revocata al termine della creazione della replica di lettura.

RDS utilizza il ruolo collegato al servizio per verificare l'autorizzazione nella regione di origine. Se si elimina il ruolo collegato al servizio durante il processo di creazione della replica, la creazione non riesce.

Per ulteriori informazioni, consulta Utilizzo di ruoli collegati ai servizi nella Guida per l'utente di IAM.

Utilizzo delle credenziali AWS Security Token Service

I token di sessione dell'endpoint global AWS Security Token Service (AWS STS) sono validi solo se abilitati per impostazione predefinita (regioni commerciali). Regioni AWS Se utilizzi le credenziali dell'operazione `assumeRole` API in AWS STS, utilizza l'endpoint regionale se la regione di origine è una regione che richiede l'attivazione. In caso contrario, la richiesta ha esito negativo. Ciò accade perché le credenziali devono essere valide in entrambe le regioni, il che vale per le regioni che hanno aderito solo quando viene utilizzato l'endpoint regionale. AWS STS

Per utilizzare l'endpoint globale, assicurarsi che sia abilitato per entrambe le regioni nelle operazioni. Imposta l'endpoint globale su `Valid in all Regioni AWS` nelle impostazioni dell'account. AWS STS

La stessa regola si applica alle credenziali nel parametro URL prefirmato.

Per ulteriori informazioni, consulta Managing AWS STS in an Regione AWS in the IAM User Guide.

Costi della replica tra regioni

Ai dati trasferiti per la replica tra regioni si applicano i costi di trasferimento dei dati di Amazon RDS. Queste operazioni di replica tra regioni generano costi per i dati trasferiti al di fuori della Regione AWS di origine:

- Quando crei una replica di lettura, Amazon RDS acquisisce uno snapshot dell'istanza di origine e lo trasferisce nella Regione AWS della replica di lettura.
- Per ogni modifica dei dati effettuata nei database di origine, Amazon RDS trasferisce i dati dalla replica di origine Regione AWS alla replica Regione AWS di lettura.

Per ulteriori informazioni sui prezzi del trasferimento dati, consulta Prezzi di Amazon RDS.

Per istanze MySQL e MariaDB, puoi ridurre i costi di trasferimento dei dati diminuendo il numero di repliche di lettura tra regioni create. Ad esempio, supponiamo di avere un'istanza DB di origine in un'istanza Regione AWS e di voler avere tre repliche di lettura in un'altra. Regione AWS In questo caso, puoi creare solo una delle repliche di lettura dall'istanza database di origine. Le altre due repliche vengono create dalla prima replica di lettura anziché dall'istanza database di origine.

Ad esempio, se ne hai `source-instance-1` una Regione AWS, puoi fare quanto segue:

- Crea `read-replica-1` nel nuovo Regione AWS, specificando `source-instance-1` come fonte.
- Crea `read-replica-2` da `read-replica-1`.
- Crea `read-replica-3` da `read-replica-1`.

In questo esempio ti verranno addebitati solo i dati trasferiti da `source-instance-1` a `read-replica-1`. I costi dei dati trasferiti da `read-replica-1` alle altre due repliche non ti verranno addebitati perché si trovano tutti nella stessa Regione AWS. Se crei tutte e tre le repliche direttamente da `source-instance-1`, ti verrà addebitato il trasferimento dei dati in tutte le repliche.

Tagging delle risorse Amazon RDS

È possibile usare i tag Amazon RDS per aggiungere metadati alle risorse Amazon RDS. Puoi utilizzare i tag per aggiungere notazioni personalizzate su istanze di database, snapshot, cluster Aurora e così via. In questo modo puoi documentare le risorse Amazon RDS. I tag possono inoltre essere utilizzati con procedure di manutenzione automatizzate.

In particolare, puoi usare questi tag con le policy IAM per gestire l'accesso alle risorse RDS e per controllare le operazioni che è possibile applicare alle risorse RDS. Puoi utilizzare questi tag anche per tenere traccia dei costi raggruppando le spese per risorse con tag simili.

Puoi contrassegnare con i tag le seguenti risorse Amazon RDS:

- Istanze DB
- Cluster database
- Endpoint del cluster DB
- Repliche di lettura
- Snapshot DB
- Snapshot cluster database
- Istanze database riservate
- Abbonamenti a eventi
- Gruppi di opzioni database
- Gruppi di parametri database
- Gruppi di parametri di cluster database
- Gruppi di sottoreti database
- Proxy RDS
- Endpoint RDS Proxy
- Distribuzioni blu/verde
- Integrazioni Zero-ETL (anteprima)

> **Note**
>
> Attualmente, non è possibile etichettare i proxy RDS e gli endpoint proxy RDS utilizzando il. AWS Management Console

Argomenti

Panoramica sui tag delle risorse di Amazon RDS

Un tag Amazon RDS è una coppia nome-valore definita e associata a una risorsa Amazon RDS. Il nome viene definito chiave. L'indicazione di un valore per la chiave è un'operazione facoltativa. È possibile usare i tag per assegnare informazioni arbitrarie a una risorsa Amazon RDS. Una chiave tag potrebbe essere impiegata, ad esempio, per definire una categoria e il valore di tag potrebbe essere un elemento di tale categoria. Ad esempio, puoi definire una chiave tag "progetto" e un valore di tag "Salix". In questo caso, indichi che la risorsa Amazon RDS è assegnata al progetto Salix. È anche possibile usare i tag per indicare le risorse di Amazon RDS usate a scopo di test o produzione tramite una chiave, ad esempio `environment=test` o `environment=production`. È consigliabile utilizzare un set coerente di chiavi di tag per agevolare il monitoraggio dei metadati associati alle risorse Amazon RDS.

Inoltre, puoi utilizzare le condizioni nelle tue policy IAM per controllare l'accesso alle AWS risorse in base ai tag presenti su quella risorsa. Puoi farlo utilizzando la chiave di condizione `aws:ResourceTag/`*`tag-key`* globale. Per ulteriori informazioni, vedere Controlling access to AWS resources nella AWS Identity and Access Management User Guide.

Ogni risorsa Amazon RDS dispone di un set di tag contenente tutti i tag assegnati a tale risorsa Amazon RDS. Un set di tag può contenere fino a 50 tag o può essere vuoto. Se aggiungi un tag a una risorsa RDS con la stessa chiave di un tag esistente per la risorsa, il nuovo valore sovrascrive quello precedente.

AWS non applica alcun significato semantico ai tag; i tag vengono interpretati rigorosamente come stringhe di caratteri. RDS può impostare i tag in un'istanza database o altre risorse RDS. L'impostazione dei tag dipende dalle opzioni utilizzate al momento della creazione della risorsa.

Ad esempio, Amazon RDS potrebbe aggiungere un tag che indica che un'istanza database viene utilizzata solo a scopo di test o produzione.

- La chiave di tag corrisponde al nome obbligatorio del tag. Il valore della stringa può essere composto da 1 a 128 caratteri Unicode e non può avere il prefisso `aws:` o `rds:`. La stringa può contenere solo il set di lettere, cifre, spazi vuoti Unicode, '_', '.', ':', '/', '=', '+', '-', '@' (Java regex: "^([\\p{L}\\p{Z}\\p{N}_.:/=+\\-@]*)$").

- Il valore di tag è un valore di stringa opzionale del tag. La lunghezza del valore della stringa può essere composta da 1 a 256 caratteri Unicode. La stringa può contenere solo il set di lettere, cifre, spazi vuoti Unicode, '_', '.', ':', '/', '=', '+', '-', '@' (Java regex: "^([\\p{L}\\p{Z}\\p{N}_.:/=+\\-@]*)$").

  I valori non devono essere necessariamente univoci in un set di tag e possono essere Null. Ad esempio, puoi avere una coppia chiave-valore in un set di tag `project=Trinity` e `cost-center=Trinity`.

Puoi utilizzare l' AWS Management ConsoleAPI Amazon RDS o Amazon RDS per aggiungere, elencare ed eliminare tag sulle risorse Amazon RDS. AWS CLI Quando usi l'interfaccia della linea di comando o l'API, devi fornire il nome della risorsa Amazon (ARN) della risorsa RDS che vuoi utilizzare. Per ulteriori informazioni sulla creazione di un ARN, consultare Costruzione di un ARN per Amazon RDS.

I tag sono memorizzati nella cache a fini di autorizzazione. Questo è il motivo per il quale le aggiunte e gli aggiornamenti dei tag delle risorse di Amazon RDS potrebbero richiedere diversi minuti prima di diventare disponibili.

Utilizzo di tag per il controllo degli accessi con IAM

È possibile utilizzare i tag con le policy IAM per gestire l'accesso alle risorse Amazon RDS. Inoltre, puoi utilizzare i tag per controllare le operazioni che è possibile applicare alle risorse Amazon RDS.

Per informazioni sulla gestione dell'accesso alle risorse con tag tramite le policy IAM, consulta Gestione accessi e identità per Amazon RDS.

Utilizzo dei tag per produrre report di fatturazione dettagliati

Puoi utilizzare questi tag anche per tenere traccia dei costi raggruppando le spese per risorse con tag simili.

Utilizza i tag per organizzare la AWS fattura in modo da rispecchiare la tua struttura dei costi. A tale scopo, registrati per ricevere la Account AWS fattura con i valori chiave dell'etichetta inclusi. Per visualizzare il costo delle risorse combinate, puoi organizzare le informazioni di fatturazione in base alle risorse con gli stessi valori di chiave di tag. Puoi ad esempio applicare tag a numerose risorse con un nome di applicazione specifico, quindi organizzare le informazioni di fatturazione per visualizzare il costo totale dell'applicazione in più servizi. Per ulteriori informazioni, consulta la pagina sull'utilizzo dei tag per l'allocazione dei costi nella Guida per l'utente di AWS Billing .

> **Note**
>
> Puoi aggiungere un tag a uno snapshot del DB; tuttavia, la fattura non rifletterà questo raggruppamento.
> I costi degli snapshot orfani vengono aggregati in un unico elemento senza tag.

Aggiunta, pubblicazione e rimozione di tag

Le procedure seguenti illustrano come eseguire operazioni di assegnazione di tag tipiche sulle risorse correlate alle istanze database .

Console

Il processo di applicazione dei tag a una risorsa di Amazon RDS è simile per tutte le risorse. Di seguito viene mostrato come applicare i tag a un'istanza database Amazon RDS.

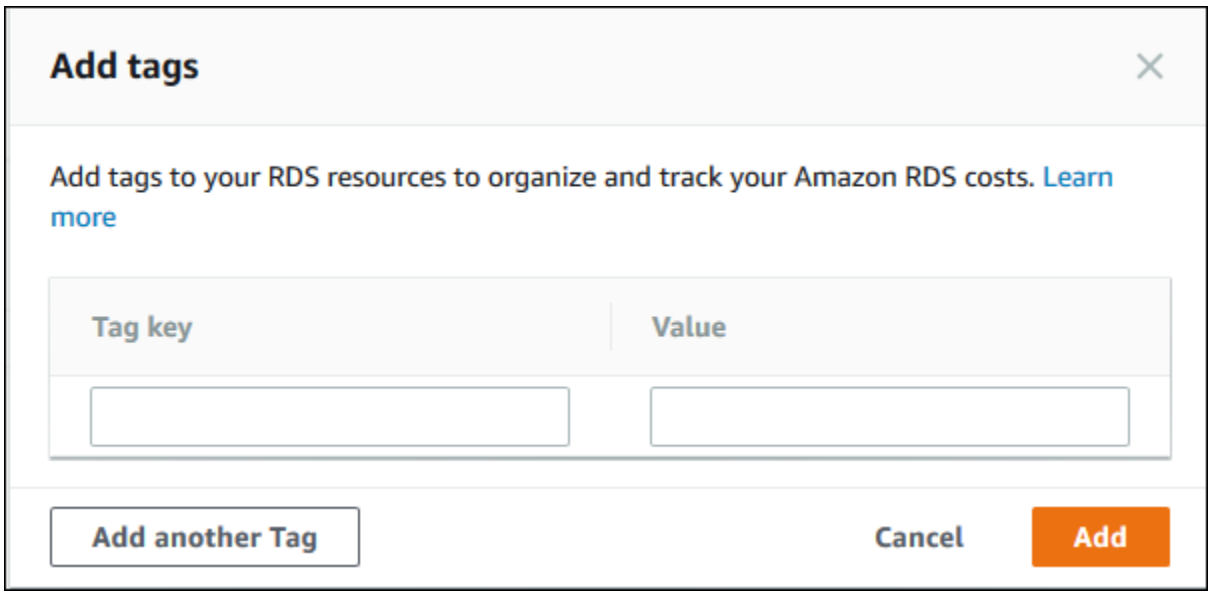
Aggiunta di un tag a un'istanza database

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, scegliere Databases (Database).

   > **Note**
   >
   > Per filtrare l'elenco di istanze database nel riquadro Databases (Database), inserire una stringa di testo per Filter instances (Filtra istanze). Vengono visualizzate solo le istanze database che contengono la stringa.

3. Scegliere il nome dell'istanza database a cui si desidera aggiungere tag per visualizzarne i dettagli.

4. Nella sezione dei dettagli, scorrere verso il basso fino alla sezione Tags (Tag).
5. Scegliere Aggiungi. Viene visualizzata la finestra Add tags (Aggiungi tag).



6. Inserire un valore per Tag key (Chiave tag) e Value (Valore).
7. Per aggiungere un altro tag, scegliere Add another Tag (Aggiungi un altro tag) e inserire un valore per Tag key (Chiave tag) e Value (Valore).

   Ripetere questa operazione tutte le volte necessarie.
8. Scegliere Aggiungi.

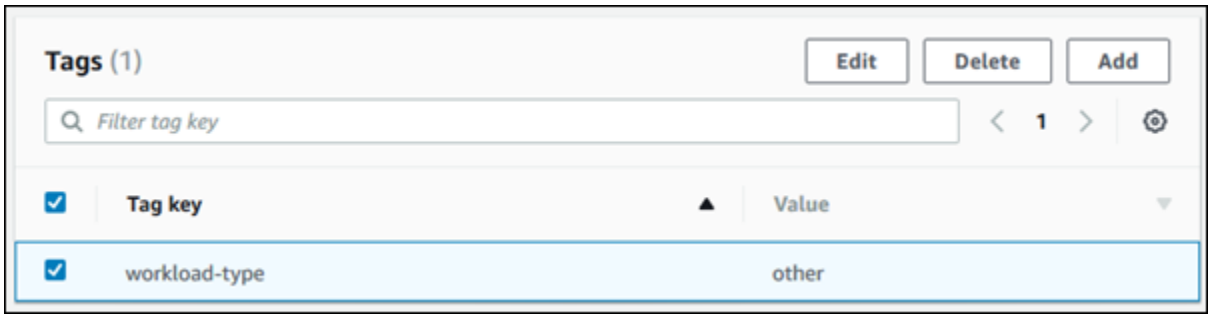
Eliminazione di un tag da un'istanza database

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, scegliere Databases (Database).

> **Note**
>
> Per filtrare l'elenco di istanze database nel riquadro Databases (Database), inserire una stringa di testo nella casella Filter instances (Filtra istanze). Vengono visualizzate solo le istanze database che contengono la stringa.

3. Scegliere il nome dell'istanza database per visualizzarne i dettagli.
4. Nella sezione dei dettagli, scorrere verso il basso fino alla sezione Tags (Tag).

5. Scegliere il tag da eliminare.



6. Scegliere Delete (Elimina) e quindi scegliere Delete (Elimina) nella finestra Delete tags (Elimina tag).

AWS CLI

È possibile aggiungere, elencare o rimuovere i tag per un'istanza database utilizzando AWS CLI.

- Per aggiungere uno o più tag a una risorsa Amazon RDS, usa il AWS CLI comando `add-tags-to-resource`.
- Per elencare i tag su una risorsa Amazon RDS, usa il AWS CLI comando `list-tags-for-resource`.
- Per rimuovere uno o più tag da una risorsa Amazon RDS, usa il AWS CLI comando `remove-tags-from-resource`.

Per ulteriori informazioni su come creare l'ARN necessario, consultare Costruzione di un ARN per Amazon RDS.

API RDS

È possibile aggiungere, elencare o rimuovere i tag per un'istanza database utilizzando l'API di Amazon RDS.

- Per aggiungere un tag a una risorsa Amazon RDS, utilizza l'operazione `AddTagsToResource`.
- Per elencare i tag assegnati a una risorsa Amazon RDS, utilizza `ListTagsForResource`.
- Per rimuovere i tag da una risorsa Amazon RDS, utilizza l'operazione `RemoveTagsFromResource`.

Per ulteriori informazioni su come creare l'ARN necessario, consultare Costruzione di un ARN per Amazon RDS.

Quando utilizzi XML con l'API di Amazon RDS i tag seguono questo schema:

```
<Tagging>
    <TagSet>
        <Tag>
            <Key>Project</Key>
            <Value>Trinity</Value>
        </Tag>
        <Tag>
            <Key>User</Key>
            <Value>Jones</Value>
        </Tag>
    </TagSet>
</Tagging>
```

La tabella riportata di seguito fornisce un elenco dei tag XML consentiti e le relative caratteristiche. I valori relativi a chiave e valore fanno distinzione tra maiuscole e minuscole. Ad esempio, project=Trinity e PROJECT=Trinity sono due tag distinti.

Elemento del tagging	Descrizione
TagSet	Un set di tag è un contenitore di tutti i tag assegnati a una risorsa Amazon RDS. Ogni risorsa può disporre di un solo set di tag. Puoi lavorare con un TagSet solo tramite l'API Amazon RDS.
Tag	Un tag è una coppia chiave-valore definita dall'utente. Un set di tag può contenere da 1 a 50 tag.
Chiave	Una chiave corrisponde al nome obbligatorio del tag. Il valore della stringa può essere composto da 1 a 128 caratteri Unicode e non può avere il prefisso `aws:` o `rds:`. La stringa può contenere solo il set di lettere, cifre, spazi vuoti Unicode, '_', '.', '/', '=', '+', '-' (espressioni regolari Java: "^([\\p{L}\\p{Z}\\p{N}_.:/=+\\-]*)$"). Le chiavi devono essere uniche per un set di tag. Ad esempio, non puoi avere una coppia di chiavi in un set di tag con la stessa chiave, ma con valori diversi, come project/Trinity e project/Xanadu.
Valore	Un valore è il valore opzione del tag. Il valore della stringa può essere composto da 1 a 256 caratteri Unicode e non può avere il prefisso `aws:`

Elemento del tagging	Descrizione
	o `rds:`. La stringa può contenere solo il set di lettere, cifre, spazi vuoti Unicode, '_', '.', '/', '=', '+', '-' (espressioni regolari Java: "^([\\p{L}\\p{Z}\\p{N}_.:/=+\\-]*)$"). I valori non devono essere necessariamente univoci in un set di tag e possono essere Null. Ad esempio, può esserci una coppia chiave-valore in un set di tag project/Trinity e in cost-center/Trinity.

Utilizzo del AWS Tag Editor

Puoi sfogliare e modificare i tag sulle tue risorse RDS AWS Management Console utilizzando l'editor di AWS tag. Per ulteriori informazioni, consulta Tag Editor nella Guida per l'utente di AWS Resource Groups.

Copia di tag in snapshot di istanze database

Quando si crea o si ripristina un'istanza database, è possibile specificare che i tag dell'istanza database vengano copiati nelle snapshot dell'istanza database. La copia dei tag garantisce che i metadati degli snapshot DB corrispondano a quelli dell'istanza database di origine. Garantisce anche che eventuali policy di accesso degli snapshot DB corrispondano a quelle dell'istanza database di origine.

È possibile specificare che i tag vengano copiati nelle snapshot DB per le seguenti azioni:

- Creazione di un'istanza database.
- Ripristino di un'istanza database.
- Creazione di una replica di lettura.
- Copia di una snapshot DB.

Nella maggior parte dei casi, i tag non vengono copiati per impostazione predefinita. Tuttavia, quando ripristini un'istanza database da uno snapshot DB, RDS verifica se vengono specificati nuovi tag. In caso affermativo, i nuovi tag vengono aggiunti all'istanza database ripristinata. Se non ci sono nuovi tag, RDS aggiunge i tag dall'istanza database di origine al momento della creazione dello snapshot nell'istanza database ripristinata.

Per evitare che i tag delle istanze database di origine vengano aggiunti alle istanze database ripristinate, si consiglia di specificare nuovi tag durante il ripristino di un'istanza database.

> **Note**
>
> In alcuni casi, è possibile includere un valore per il `--tags` parametro del create-db-snapshot AWS CLI comando. In alternativa puoi specificare almeno un tag per l'operazione API CreateDBSnapshot. In questi casi, RDS non copia i tag dall'istanza database di origine nel nuovo snapshot DB. Questa funzionalità si applica anche se per l'istanza database di origine è stata attivata l'opzione `--copy-tags-to-snapshot` (`CopyTagsToSnapshot`).
> Con questo approccio puoi creare la copia di un'istanza database da uno snapshot DB ed evitare di aggiungere tag che non si applicano alla nuova istanza database. È possibile creare lo snapshot DB utilizzando il AWS CLI `create-db-snapshot` comando (o l'operazione API `CreateDBSnapshot` RDS). Dopo aver creato uno snapshot DB, puoi aggiungere i tag come descritto più avanti in questo argomento.

Tutorial: Utilizzo dei tag per specificare le istanze database da interrompere

Si assuma di creare un numero di istanze database in un ambiente di sviluppo o di test. Sarà necessario mantenere tutte queste istanze database per diversi giorni. Alcune istanze database eseguono test durante gli orari notturni. Altre istanze database possono essere arrestate durante la notte e riavviate il giorno successivo. Nell'esempio seguente viene illustrato come assegnare un tag a quelle istanze database che possono essere interrotte durante la notte. L'esempio mostra come uno script può rilevare le istanze database che hanno quel tag e quindi arrestarle. In questo esempio, la parte del valore della coppia chiave-valore non ha importanza. La presenza del tag `stoppable` indica che l'istanza database ha questa proprietà definita dall'utente.

Per specificare le istanze database da interrompere

1. Innanzitutto, determina l'ARN di un'istanza database che desideri indicare come arrestabile.

   I comandi e le API per l'assegnazione dei tag funzionano con gli ARN. In questo modo, possono funzionare senza problemi tra AWS regioni, AWS account e diversi tipi di risorse che potrebbero avere nomi brevi identici. Puoi specificare l'ARN anziché l'ID istanza database nei comandi della CLI che operano su istanze database. Sostituisci il nome delle tue istanze DB con. *`dev-test-db-instance`* Nei comandi successivi che utilizzano i parametri ARN, sostituisci l'ARN della

tua istanza database. L'ARN include l'ID AWS dell'account e il nome della AWS regione in cui si trova l'istanza DB.

```
$ aws rds describe-db-instances --db-instance-identifier dev-test-db-instance \
  --query "*[].{DBInstance:DBInstanceArn}" --output text
arn:aws:rds:us-east-1:123456789102:db:dev-test-db-instance
```

2. Quindi, aggiungi il tag `stoppable` a questa istanza database.

   Scegli il nome per il tag. Con questo approccio puoi evitare di definire una convenzione di denominazione che codifichi tutte le informazioni rilevanti nei nomi. Con la convenzione puoi codificare le informazioni nel nome dell'istanza database o nei nomi di altre risorse. Poiché questo esempio considera il tag come un attributo presente o assente, viene omessa la parte `Value=` del parametro `--tags`.

   ```
   $ aws rds add-tags-to-resource \
     --resource-name arn:aws:rds:us-east-1:123456789102:db:dev-test-db-instance \
     --tags Key=stoppable
   ```

3. Verifica che il tag sia presente nell'istanza database.

   Questi comandi recuperano le informazioni sui tag per l'istanza database in formato JSON e in testo semplice separato da tabulazioni.

   ```
   $ aws rds list-tags-for-resource \
     --resource-name arn:aws:rds:us-east-1:123456789102:db:dev-test-db-instance
   {
       "TagList": [
           {
               "Key": "stoppable",
               "Value": ""

           }
       ]
   }
   aws rds list-tags-for-resource \
     --resource-name arn:aws:rds:us-east-1:123456789102:db:dev-test-db-instance --
   output text
   TAGLIST stoppable
   ```

4. Per arrestare tutte le istanze database designate come `stoppable`, prepara un elenco di tutte le istanze database. Scorri l'elenco e verifica se ogni istanza database è contrassegnata con l'attributo pertinente.

   In questo esempio di Linux viene utilizzato lo script della shell per salvare l'elenco di ARN delle istanze database in un file temporaneo e quindi eseguire i comandi dell'Interfaccia della linea di comando per ogni istanza database.

```
$ aws rds describe-db-instances --query "*[].[DBInstanceArn]" --output text >/tmp/
db_instance_arns.lst
$ for arn in $(cat /tmp/db_instance_arns.lst)
do
  match="$(aws rds list-tags-for-resource --resource-name $arn --output text | grep
 stoppable)"
  if [[ ! -z "$match" ]]
  then
      echo "DB instance $arn is tagged as stoppable. Stopping it now."
# Note that you need to get the DB instance identifier from the ARN.
      dbid=$(echo $arn | sed -e 's/.*://')
      aws rds stop-db-instance --db-instance-identifier $dbid
  fi
done

DB instance arn:arn:aws:rds:us-east-1:123456789102:db:dev-test-db-instance is
 tagged as stoppable. Stopping it now.
{
    "DBInstance": {
        "DBInstanceIdentifier": "dev-test-db-instance",
        "DBInstanceClass": "db.t3.medium",
        ...
```

Puoi eseguire uno script come questo alla fine di ogni giorno per assicurarti che le istanze database non essenziali vengano arrestate. Puoi inoltre pianificare un processo utilizzando un'utilità come `cron` per eseguire tale controllo ogni notte. Ad esempio, puoi eseguire questa operazione nel caso in cui alcune istanze database venissero lasciate in esecuzione per errore. Puoi quindi ottimizzare il comando che prepara l'elenco di istanze database da controllare.

Il comando seguente produce un elenco delle istanze database nello stato `available`. Lo script può ignorare le istanze database già arrestate, poiché avranno valori di stato diversi, ad esempio `stopped` o `stopping`.

```
$ aws rds describe-db-instances \
  --query '*[].{DBInstanceArn:DBInstanceArn,DBInstanceStatus:DBInstanceStatus}|[?
DBInstanceStatus == `available`]|[].{DBInstanceArn:DBInstanceArn}' \
  --output text
arn:aws:rds:us-east-1:123456789102:db:db-instance-2447
arn:aws:rds:us-east-1:123456789102:db:db-instance-3395
arn:aws:rds:us-east-1:123456789102:db:dev-test-db-instance
arn:aws:rds:us-east-1:123456789102:db:pg2-db-instance
```

Tip

Puoi utilizzare l'assegnazione di tag e la ricerca di istanze database con i tag per ridurre i costi in altri modi. Supponi, ad esempio, questo scenario con le istanze database utilizzate per lo sviluppo e il test. In questo caso, potresti designare alcune istanze database da eliminare alla fine di ogni giornata. In alternativa, potresti designarle per modificare le istanze database in piccole classi di istanza database durante i periodi previsti di basso utilizzo.

Utilizzo di Amazon Resource Name (ARN) in Amazon RDS

Le risorse create in Amazon Web Services sono identificate in modo univoco con un Amazon Resource Name (ARN). Per determinate operazioni Amazon RDS, è necessario identificare in modo univoco una risorsa Amazon RDS specificandone l'ARN. Quando, ad esempio, crei una replica di lettura di un'istanza database di RDS, devi fornire l'ARN dell'istanza database di origine.

Costruzione di un ARN per Amazon RDS

Le risorse create in Amazon Web Services sono identificate in modo univoco con un Amazon Resource Name (ARN). È possibile creare un ARN per una risorsa Amazon RDS utilizzando la sintassi seguente.

`arn:aws:rds:<region>:<account number>:<resourcetype>:<name>`

Nome della regione	Regione	Endpoint	Protocollo
US East (Ohio)	us-east-2	rds.us-east-2.amazonaws.com	HTTPS
		rds-fips.us-east-2.api.aws	HTTPS
		rds.us-east-2.api.aws	HTTPS
		rds-fips.us-east-2.amazonaws.com	HTTPS
US East (N. Virginia)	us-east-1	rds.us-east-1.amazonaws.com	HTTPS
		rds-fips.us-east-1.api.aws	HTTPS
		rds-fips.us-east-1.amazonaws.com	HTTPS
		rds.us-east-1.api.aws	HTTPS
Stati Uniti occidentali (California	us-west-1	rds.us-west-1.amazonaws.com	HTTPS
		rds.us-west-1.api.aws	HTTPS
		rds-fips.us-west-1.amazonaws.com	HTTPS

Nome della regione	Regione	Endpoint	Protocollo	
settentrionale)		rds-fips.us-west-1.api.aws	HTTPS	
US West (Oregon)	us-west-2	rds.us-west-2.amazonaws.com rds-fips.us-west-2.amazonaws.com rds.us-west-2.api.aws rds-fips.us-west-2.api.aws	HTTPS HTTPS HTTPS HTTPS	
Africa (Cape Town)	af-south-1	rds.af-south-1.amazonaws.com rds.af-south-1.api.aws	HTTPS HTTPS	
Asia Pacifico (Hong Kong)	ap-east-1	rds.ap-east-1.amazonaws.com rds.ap-east-1.api.aws	HTTPS HTTPS	
Asia Pacific (Hyderabad)	ap-south-2	rds.ap-south-2.amazonaws.com rds.ap-south-2.api.aws	HTTPS HTTPS	
Asia Pacifico (Giacarta)	ap-southeast-3	rds.ap-southeast-3.amazonaws.com rds.ap-southeast-3.api.aws	HTTPS HTTPS	
Asia Pacifico (Melbourne)	ap-southeast-4	rds.ap-southeast-4.amazonaws.com rds.ap-southeast-4.api.aws	HTTPS HTTPS	

Nome della regione	Regione	Endpoint	Protocollo	
Asia Pacifico (Mumbai)	ap-south-1	rds.ap-south-1.amazonaws.com rds.ap-south-1.api.aws	HTTPS HTTPS	
Asia Pacifico (Osaka-Locale)	ap-northeast-3	rds.ap-northeast-3.amazonaws.com rds.ap-northeast-3.api.aws	HTTPS HTTPS	
Asia Pacifico (Seoul)	ap-northeast-2	rds.ap-northeast-2.amazonaws.com rds.ap-northeast-2.api.aws	HTTPS HTTPS	
Asia Pacifico (Singapore)	ap-southeast-1	rds.ap-southeast-1.amazonaws.com rds.ap-southeast-1.api.aws	HTTPS HTTPS	
Asia Pacifico (Sydney)	ap-southeast-2	rds.ap-southeast-2.amazonaws.com rds.ap-southeast-2.api.aws	HTTPS HTTPS	
Asia Pacifico (Tokyo)	ap-northeast-1	rds.ap-northeast-1.amazonaws.com rds.ap-northeast-1.api.aws	HTTPS HTTPS	
Canada (Centrale)	ca-central-1	rds.ca-central-1.amazonaws.com rds.ca-central-1.api.aws rds-fips.ca-central-1.api.aws rds-fips.ca-central-1.amazonaws.com	HTTPS HTTPS HTTPS HTTPS	

Nome della regione	Regione	Endpoint	Protocollo	
Canada occidentale (Calgary)	ca-west-1	rds.ca-west-1.amazonaws.com rds-fips.ca-west-1.amazonaws.com	HTTPS HTTPS	
Europa (Francoforte)	eu-central-1	rds.eu-central-1.amazonaws.com rds.eu-central-1.api.aws	HTTPS HTTPS	
Europa (Irlanda)	eu-west-1	rds.eu-west-1.amazonaws.com rds.eu-west-1.api.aws	HTTPS HTTPS	
Europa (Londra)	eu-west-2	rds.eu-west-2.amazonaws.com rds.eu-west-2.api.aws	HTTPS HTTPS	
Europa (Milano)	eu-south-1	rds.eu-south-1.amazonaws.com rds.eu-south-1.api.aws	HTTPS HTTPS	
Europa (Parigi)	eu-west-3	rds.eu-west-3.amazonaws.com rds.eu-west-3.api.aws	HTTPS HTTPS	
Europa (Spagna)	eu-south-2	rds.eu-south-2.amazonaws.com rds.eu-south-2.api.aws	HTTPS HTTPS	
Europa (Stoccolma)	eu-north-1	rds.eu-north-1.amazonaws.com rds.eu-north-1.api.aws	HTTPS HTTPS	
Europa (Zurigo)	eu-central-2	rds.eu-central-2.amazonaws.com rds.eu-central-2.api.aws	HTTPS HTTPS	

Nome della regione	Regione	Endpoint	Protocollo
Israele (Tel Aviv)	il-central-1	rds.il-central-1.amazonaws.com rds.il-central-1.api.aws	HTTPS HTTPS
Medio Oriente (Bahrein)	me-south-1	rds.me-south-1.amazonaws.com rds.me-south-1.api.aws	HTTPS HTTPS
Medio Oriente (Emirati Arabi Uniti)	me-central-1	rds.me-central-1.amazonaws.com rds.me-central-1.api.aws	HTTPS HTTPS
Sud America (São Paulo)	sa-east-1	rds.sa-east-1.amazonaws.com rds.sa-east-1.api.aws	HTTPS HTTPS
AWS GovCloud (Stati Uniti orientali)	us-gov-east-1	rds.us-gov-east-1.amazonaws.com rds.us-gov-east-1.api.aws	HTTPS HTTPS
AWS GovCloud (Stati Uniti occidentali)	us-gov-west-1	rds.us-gov-west-1.amazonaws.com rds.us-gov-west-1.api.aws	HTTPS HTTPS

La tabella riportata di seguito mostra il formato da utilizzare quando si crea un ARN per un determinato tipo di risorsa di Amazon RDS.

Tipo di risorsa	Formato ARN
Istanza database	arn:aws:rds:*<region>*:*<account>* :db:*<name>* Ad esempio: `arn:aws:rds: us-east-2 :123456789012 :db:my-mysql-instance-1`
Cluster DB	arn:aws:rds:*<region>*:*<account>* :cluster:*<name>* Ad esempio: `arn:aws:rds: us-east-2 :123456789012 :cluster: my-aurora-cluster-1`
Sottoscrizione a eventi	arn:aws:rds:*<region>*:*<account>* :es:*<name>* Ad esempio: `arn:aws:rds: us-east-2 :123456789012 :es:my-subscription`
Gruppo di opzioni database	arn:aws:rds:*<region>*:*<account>* :og:*<name>* Ad esempio: `arn:aws:rds: us-east-2 :123456789012 :og:my-og`
DB parameter group (Gruppo di parametri database)	arn:aws:rds:*<region>*:*<account>* :pg:*<name>* Ad esempio: `arn:aws:rds: us-east-2 :123456789012 :pg:my-param-enable-logs`
Gruppo di parametri del cluster DB	arn:aws:rds:*<region>*:*<account>* :cluster-pg:*<name>* Ad esempio:

Tipo di risorsa	Formato ARN
	`arn:aws:rds: us-east-2 :123456789012 :cluster-pg: my-cluster-param-timezone`
Istanza database riservata	arn:aws:rds:*<region>*:*<account>* :ri:*<name>* Ad esempio: `arn:aws:rds: us-east-2 :123456789012 :ri:my-reserved-postgresql`
Gruppo di sicurezza DB	arn:aws:rds:*<region>*:*<account>* :secgrp:*<name>* Ad esempio: `arn:aws:rds: us-east-2 :123456789012 :secgrp:my-public`
Snapshot di database automatizzato	arn:aws:rds:*<region>*:*<account>* :snapshot:rds:*<name>* Ad esempio: `arn:aws:rds: us-east-2 :123456789012 :snapshot:rds: my-mysql-db-2019-07-22-07-23`
Snapshot di cluster database automatizzato	arn:aws:rds:*<region>*:*<account>* :cluster-snapshot:rds:*<name>* Ad esempio: `arn:aws:rds: us-east-2 :123456789012 :cluster-snapshot:rds:  my-aurora-cluster-2019-07-22-16-16`

Tipo di risorsa	Formato ARN
Snapshot di database manuale	arn:aws:rds:*<region>*:*<account>* :snapshot:*<name>* Ad esempio: `arn:aws:rds: us-east-2 :123456789012 :snapshot: my-mysql-db-snap`
Snapshot del cluster di database manuale	arn:aws:rds:*<region>*:*<account>* :cluster-snapshot:*<name>* Ad esempio: `arn:aws:rds: us-east-2 :123456789012 :cluster-snapshot: my-aurora-cluster-snap`
Gruppo di sottoreti DB	arn:aws:rds:*<region>*:*<account>* :subgrp:*<name>* Ad esempio: `arn:aws:rds: us-east-2 :123456789012 :subgrp:my-subnet-10`

Recupero di un ARN esistente

È possibile ottenere l'ARN di una risorsa RDS utilizzando l'API AWS Management Console, AWS Command Line Interface (AWS CLI) o RDS.

Console

Per ottenere un ARN da AWS Management Console, vai alla risorsa per cui desideri un ARN e visualizza i dettagli di quella risorsa.

Ad esempio, puoi ottenere l'ARN per un'istanza database dalla scheda Configurazione dei dettagli dell'istanza database.

AWS CLI

Per ottenere un ARN da AWS CLI per una particolare risorsa RDS, si utilizza il `describe` comando relativo a tale risorsa. La tabella seguente mostra ogni AWS CLI comando e la proprietà ARN utilizzata con il comando per ottenere un ARN.

AWS CLI comando	Proprietà ARN
describe-event-subscriptions	EventSubscriptionArn
describe-certificates	CertificateArn
describe-db-parameter-groups	DB ParameterGroupArn
describe-db-cluster-parameter-gruppi	DB ClusterParameterGroupArn
describe-db-instances	DB InstanceArn
describe-db-security-groups	DB SecurityGroupArn
describe-db-snapshots	DB SnapshotArn
describe-events	SourceArn
describe-reserved-db-instances	DB riservato InstanceArn
describe-db-subnet-groups	DB SubnetGroupArn
describe-option-groups	OptionGroupArn
describe-db-clusters	DB ClusterArn
describe-db-cluster-snapshots	DB ClusterSnapshotArn

Ad esempio, il AWS CLI comando seguente ottiene l'ARN per un'istanza DB.

Example

Per LinuxmacOS, oUnix:

```
aws rds describe-db-instances \
--db-instance-identifier DBInstanceIdentifier \
--region us-west-2 \
--query "*[].{DBInstanceIdentifier:DBInstanceIdentifier,DBInstanceArn:DBInstanceArn}"
```

Per Windows:

```
aws rds describe-db-instances ^
--db-instance-identifier DBInstanceIdentifier ^
--region us-west-2 ^
--query "*[].{DBInstanceIdentifier:DBInstanceIdentifier,DBInstanceArn:DBInstanceArn}"
```

L'output del comando è simile al seguente:

```
[
    {
        "DBInstanceArn": "arn:aws:rds:us-west-2:account_id:db:instance_id",
        "DBInstanceIdentifier": "instance_id"
    }
]
```

API RDS

Per ottenere un ARN per una determinata risorsa RDS, puoi chiamare le operazioni API RDS e le proprietà ARN seguenti.

Operazione API RDS	Proprietà ARN
DescribeEventSubscriptions	EventSubscriptionArn
DescribeCertificates	CertificateArn
Descritto B ParameterGroups	DB ParameterGroupArn
Descritto B ClusterParameterGroups	DB ClusterParameterGroupArn
DescribeDBInstances	DB InstanceArn

Operazione API RDS	Proprietà ARN
Descritto B SecurityGroups	DB SecurityGroupArn
DescribeDBSnapshots	DB SnapshotArn
DescribeEvents	SourceArn
DescribeReservedIstanze DB	DB riservato InstanceArn
B descritto SubnetGroups	DB SubnetGroupArn
DescribeOptionGroups	OptionGroupArn
DescribeDBClusters	DB ClusterArn
Descritto B ClusterSnapshots	DB ClusterSnapshotArn

Uso dello storage per istanze database di Amazon RDS

Per specificare il modo in cui archiviare i dati in Amazon RDS, scegliere un tipo di storage e specificare le dimensioni di storage quando crei o modifichi un'istanza database. In seguito, potrai aumentare la quantità o cambiare il tipo di storage modificando l'istanza database. Per ulteriori informazioni sul tipo di storage da usare per un carico di lavoro specifico, consulta Tipi di storage Amazon RDS.

Argomenti

Aumento della capacità di storage dell'istanza database

Se ti serve spazio per dati aggiuntivi, puoi aumentare la quantità di storage di un'istanza database esistente. A questo scopo, puoi utilizzare la console Amazon RDS, l'API Amazon RDS o la AWS Command Line Interface (AWS CLI). Per informazioni sui limiti di storage, consulta Storage delle istanze di database Amazon RDS.

> **Note**
>
> Il dimensionamento dello storage per Amazon RDS per istanze database Microsoft SQL Server è supportato solo per i tipi di storage SSD per scopi generici ed SSD Provisioned IOPS.

Per monitorare la quantità di spazio di archiviazione gratuito per la tua istanza DB in modo da poter rispondere quando necessario, ti consigliamo di creare un CloudWatch allarme Amazon. Per ulteriori informazioni sull'impostazione degli CloudWatch allarmi, consulta Uso degli CloudWatch allarmi.

Il dimensionamento dell'archiviazione di solito non causa alcuna interruzione o peggioramento delle prestazioni dell'istanza database. Dopo aver modificato le dimensioni di storage di un'istanza database, lo stato passa a storage-optimization (ottimizzazione-storage).

> **Note**
>
> L'ottimizzazione dello spazio di archiviazione può richiedere alcune ore. Non puoi apportare altre modifiche all'archiviazione prima di sei (6) ore dal completamento dell'ottimizzazione dello spazio di archiviazione nell'istanza. È possibile visualizzare l'avanzamento dell'ottimizzazione dello storage in AWS Management Console o utilizzando il describe-db-instances AWS CLI comando.

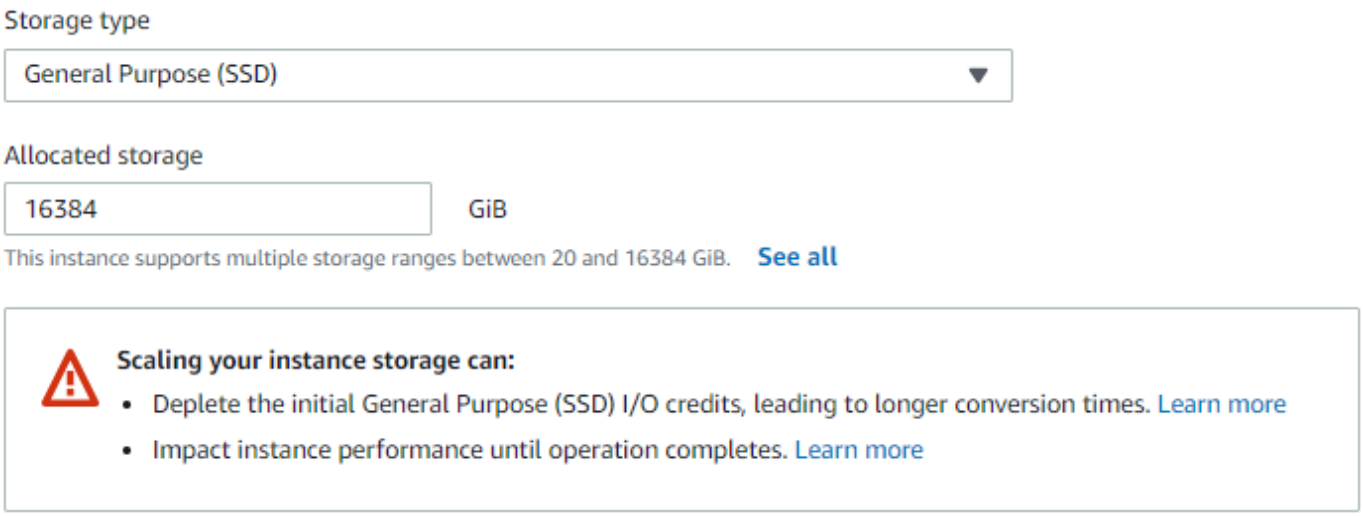
Tuttavia, un caso particolare è se hai un'istanza database SQL Server e non hai modificato la configurazione dello storage da novembre 2017. In questo caso, si potrebbe verificare un'interruzione di alcuni minuti quando modifichi l'istanza database per aumentare lo storage allocato. Dopo l'interruzione, l'istanza database è di nuovo online, ma con stato `storage-optimization`. Le prestazioni possono risultare inferiori durante l'ottimizzazione dello storage.

> **Note**
>
> Non puoi ridurre la quantità di storage per un'istanza database dopo l'allocazione. Quando si aumenta lo storage allocato, questo valore deve essere almeno del 10%. Se si prova ad aumentarlo di un valore inferiore al 10%, verrà visualizzato un errore.

Console

Per aumentare lo storage per un'istanza database

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, scegliere Databases (Database).
3. Scegliere l'istanza database che si vuole modificare.
4. Scegliere Modify (Modifica).
5. Inserire un nuovo valore per Allocated Storage (Storage allocato). Questo valore deve essere maggiore di quello corrente.



6. Scegliere Continue (Continua) per passare alla schermata successiva.
7. Scegliere Apply immediately (Applica immediatamente) nella sezione Scheduling of modifications (Pianificazione delle modifiche) per applicare immediatamente le modifiche all'istanza database.

   In alternativa, scegliere Apply during the next scheduled maintenance window (Applica durante la prossima finestra di manutenzione pianificata): per applicare le modifiche durante la prossima finestra di manutenzione.
8. Dopo aver specificato le impostazioni desiderate, scegliere Modify DB instance (Modifica istanza database).

AWS CLI

Per aumentare lo storage per un'istanza DB, usa il AWS CLI comando `modify-db-instance`. Imposta i seguenti parametri:

- `--allocated-storage`: la quantità di spazio di archiviazione, in gibibyte, da allocare per l'istanza database.
- `--apply-immediately`: utilizza `--apply-immediately` per applicare immediatamente le modifiche.

  Oppure utilizza `--no-apply-immediately` (impostazione di default) per applicare le modifiche durante la finestra di manutenzione successiva. Quando vengono applicate le modifiche, si verifica un'interruzione immediata.

Per ulteriori informazioni sullo storage, consultare Storage delle istanze di database Amazon RDS.

API RDS

Per aumentare lo storage per un'istanza database, utilizza l'operazione API Amazon RDS `ModifyDBInstance`. Imposta i seguenti parametri:

- `AllocatedStorage`: la quantità di spazio di archiviazione, in gibibyte, da allocare per l'istanza database.
- `ApplyImmediately`: imposta questa opzione su `True` per applicare immediatamente le modifiche. Imposta questa opzione su `False` (impostazione di default) per applicare le modifiche durante la finestra di manutenzione successiva. Quando vengono applicate le modifiche, si verifica un'interruzione immediata.

Per ulteriori informazioni sullo storage, consultare Storage delle istanze di database Amazon RDS.

Gestione della capacità automaticamente con Auto Scaling dello storage Amazon RDS

Se il carico di lavoro è imprevedibile, puoi abilitare l'Auto Scaling dello storage per un'istanza database Amazon RDS. A questo scopo, puoi utilizzare la console Amazon RDS, l'API Amazon RDS o la AWS CLI.

Ad esempio, potresti utilizzare questa funzionalità per una nuova applicazione di gaming mobile che gli utenti stanno adottando rapidamente. In questo caso, un carico di lavoro in rapido aumento potrebbe superare lo storage di database disponibile. Per evitare di dover dimensionare manualmente lo storage del database, puoi utilizzare l'Auto Scaling dello storage Amazon RDS.

Con la scalabilità automatica dello spazio di archiviazione, quando Amazon RDS rileva che stai per esaurire lo spazio libero sul database lo adatta automaticamente al tuo spazio. Amazon RDS avvia una modifica dello spazio di archiviazione per un'istanza database abilitata per la scalabilità automatica quando si applicano questi fattori:

- Lo spazio disponibile gratuito è minore o uguale al 10% dello spazio di archiviazione allocato.
- La condizione di storage basso dura almeno cinque minuti.
- Sono trascorse almeno sei ore dall'ultima modifica dell'archiviazione oppure l'ottimizzazione dello spazio di archiviazione nell'istanza, qualunque sia il periodo più lungo.

Lo spazio di archiviazione aggiuntivo è in incrementi di uno dei seguenti valori:

- 10 GiB
- 10% dello spazio di archiviazione attualmente allocato
- Crescita prevista dello spazio di archiviazione superiore alla dimensione attuale dello spazio di archiviazione allocato nelle prossime 7 ore in base alla metrica `FreeStorageSpace` dell'ultima ora. Per ulteriori informazioni sui parametri, consulta Monitoraggio con Amazon CloudWatch.

La soglia massima di archiviazione è il limite impostato per l'autoscaling dell'istanza database. Vengono applicati i seguenti vincoli:

- È necessario impostare la soglia massima di archiviazione a un valore superiore di almeno il 10% allo spazio di archiviazione correntemente allocato. Si consiglia di impostarlo almeno sul 26% o su un valore superiore per evitare di ricevere una notifica di evento indicante che le dimensioni dell'archiviazione si stanno avvicinando alla soglia massima definita per lo spazio di archiviazione

  Ad esempio, se si dispone di un'istanza DB con 1.000 GiB di spazio di archiviazione, impostare la soglia massima di archiviazione su almeno 1.100 GiB. In caso contrario, viene visualizzato un errore del tipo Dimensione massima di archiviazione non valida per *`engine_name`*. Si consiglia tuttavia di impostare la soglia massima dello spazio di archiviazione su un valore pari ad almeno 1260 GB per evitare di ricevere una notifica di evento.
- Per un'istanza DB che utilizza lo storage Provisioned IOPS (io1 o io2 Block Express), il rapporto tra IOPS e la soglia di archiviazione massima (in GiB) deve rientrare in un determinato intervallo. Per ulteriori informazioni, consulta Storage SSD Provisioned IOPS.
- Non puoi impostare la soglia massima dello spazio di archiviazione per le istanze con dimensionamento automatico abilitato su un valore maggiore rispetto a quello dello spazio di archiviazione massimo allocato.

  Per esempio, SQL Server Standard Edition su db.m5.xlarge ha uno storage predefinito per l'istanza di 20 GiB (il minimo) e uno storage massimo di 16.384 GiB. La soglia predefinita massima per l'Auto Scaling è 1.000 GiB. Se usi i valori predefiniti, l'istanza non scala automaticamente sopra i 1.000 GiB. Questo è vero anche se lo storage allocato massimo per istanza è 16.384 GiB.

> **Note**
>
> Si consiglia di scegliere con attenzione la soglia massima di storage in base ai modelli di utilizzo e alle esigenze dei clienti. In caso di aberrazioni nei modelli di utilizzo, la soglia massima di storage può impedire il ridimensionamento dello spazio di storage a un valore

inaspettatamente elevato quando il ridimensionamento automatico prevede una soglia molto alta. Dopo che un'istanza database è stata ridimensionata automaticamente, la memoria allocata non può essere ridotta.

Argomenti

Limitazioni

Le seguenti limitazioni si applicano all'Auto Scaling dello storage:

- La scalabilità automatica non si verifica se la soglia di archiviazione massima viene superata dall'incremento di archiviazione.
- Durante il dimensionamento automatico, RDS prevede le dimensioni dell'archiviazione per le successive operazioni di dimensionamento automatico. Se si prevede che un'operazione successiva superi la soglia massima di archiviazione, RDS viene automaticamente dimensionato alla soglia massima di archiviazione.
- La scalabilità automatica non può impedire completamente situazioni complete di archiviazione per carichi di dati di grandi dimensioni. Questo perché ulteriori modifiche di archiviazione non possono essere fatte per sei (6) ore o fino a quando l'ottimizzazione dell'archiviazione è stata completata sull'istanza, qualunque sia il periodo più lungo.

  Se esegui un carico di dati di grandi dimensioni e il dimensionamento automatico non fornisce spazio sufficiente, il database potrebbe rimanere nello stato di storage pieno per diverse ore. Questo può danneggiare il database.
- Se avvii un'operazione di Auto Scaling dello storage nello stesso momento in cui Amazon RDS avvia un'operazione di Auto Scaling, la modifica dello storage ha la precedenza. L'operazione di Auto Scaling è annullata.
- La scalabilità automatica non può ridurre lo spazio di archiviazione allocato. Non puoi ridurre la quantità di storage per un'istanza database dopo l'allocazione.
- L'operazione di Auto Scaling non può essere utilizzata con lo storage magnetico.

- L'operazione di Auto Scaling non può essere utilizzata con le seguenti classi di istanza della generazione precedente le cui dimensioni di storage ordinabile sono inferiori a 6 TiB: db.m3.large, db.m3.xlarge e db.m3.2xlarge.
- Le operazioni di scalabilità automatica non vengono registrate da. AWS CloudTrail Per ulteriori informazioni su, vedere. CloudTrail Monitoraggio delle chamate API di Amazon RDS in AWS CloudTrail

Sebbene l'Auto Scaling aiuti ad aumentare lo storage nell'istanza database Amazon RDS in maniera dinamica, devi comunque configurare lo storage iniziale per l'istanza database con dimensioni appropriate per il carico di lavoro tipico.

Abilitazione dell'Auto Scaling per una nuova istanza database

Quando crei una nuova istanza database Amazon RDS, puoi scegliere se abilitare l'Auto Scaling dello storage. Puoi anche impostare un limite massimo dello storage che Amazon RDS può allocare per l'istanza database.

> **Note**
>
> Quando cloni un'istanza database Amazon RDS con l'Auto Scaling dello storage abilitato, quell'impostazione non viene ereditata automaticamente dall'istanza clonata. La nuova istanza database ha la stessa quantità di storage allocato dell'istanza originale. Puoi attivare nuovamente l'Auto Scaling dello storage per la nuova istanza se l'istanza clonata continua ad aumentare i requisiti di storage.

Console

Come abilitare l'Auto Scaling dello storage per una nuova istanza database

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nell'angolo in alto a destra della console Amazon RDS, scegli la AWS regione in cui desideri creare l'istanza DB.
3. Nel riquadro di navigazione, scegliere Databases (Database).

4. Scegliere Create database (Crea database). Nella pagina Select engine (Seleziona motore), scegliere il motore del database e specificare le informazioni dell'istanza database come descritto in Nozioni di base su Amazon RDS.
5. Nella sezione Storage Autoscaling (Auto Scaling dello storage), impostare il valore Maximum Storage Limit (Limite di storage massimo) per l'istanza database.
6. Specificare il resto delle informazioni dell'istanza database come descritto in Nozioni di base su Amazon RDS.

AWS CLI

Per abilitare la scalabilità automatica dello storage per una nuova istanza DB, usa il comando. AWS CLI `create-db-instance` Imposta il seguente parametro:

- `--max-allocated-storage`: attiva la scalabilità automatica dello spazio di archiviazione e imposta il limite massimo delle dimensioni dell'archiviazione, in gibibyte.

Per verificare che lo storage autoscaling di Amazon RDS sia disponibile per la tua istanza DB, usa il comando. AWS CLI `describe-valid-db-instance-modifications` Per effettuare il controllo in base alla classe dell'istanza prima della creazione dell'istanza, utilizza il comando `describe-orderable-db-instance-options`. Controlla il seguente campo nel valore restituito:

- `SupportsStorageAutoscaling`: indica se l'istanza database o la classe di istanza supporta la scalabilità automatica dell'archiviazione.

Per ulteriori informazioni sullo storage, consultare Storage delle istanze di database Amazon RDS.

API RDS

Per abilitare l'Auto Scaling dello storage per una nuova istanza database, utilizza l'operazione API di Amazon RDS `CreateDBInstance`. Imposta il seguente parametro:

- `MaxAllocatedStorage`: attiva la scalabilità automatica dello spazio di archiviazione Amazon RDS e imposta il limite massimo delle dimensioni dell'archiviazione, in gibibyte.

Per verificare che l'Auto Scaling dello storage Amazon RDS sia disponibile per l'istanza database, utilizza l'operazione API di Amazon RDS `DescribeValidDbInstanceModifications` per

un'istanza esistente o l'operazione `DescribeOrderableDBInstanceOptions` prima della creazione di un'istanza. Controlla il seguente campo nel valore restituito:

- `SupportsStorageAutoscaling`: indica se l'istanza database supporta la scalabilità automatica dell'archiviazione.

Per ulteriori informazioni sullo storage, consultare Storage delle istanze di database Amazon RDS.

Modifica delle impostazioni dell'Auto Scaling dello storage per un'istanza database

Puoi attivare l'Auto Scaling dello storage per un'istanza database Amazon RDS esistente. Puoi anche impostare un limite massimo dello storage che Amazon RDS può allocare per l'istanza database.

Console

Come modificare le impostazioni dell'Auto Scaling dello storage per un'istanza database

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, scegliere Databases (Database).
3. Scegliere l'istanza database che si desidera modificare e selezionare Modify (Modifica). Viene visualizzata la pagina Modify DB Instance (Modifica istanza database).
4. Modificare il limite dello storage nella sezione Autoscaling (Auto Scaling). Per ulteriori informazioni, consulta Modifica di un'istanza database Amazon RDS.
5. Quando tutte le modifiche sono come le desideri, scegli Continue (Continua) e controllale.
6. Nella pagina di conferma esaminare le modifiche. Se sono corrette, scegliere Modify DB Instance (Modifica istanza database) per salvare le modifiche. Se non sono corrette, scegliere Back (Indietro) per cambiare le modifiche o Cancel (Annulla) per annullare le modifiche.

   Le modifiche al limite di storage dell'Auto Scaling vengono eseguite immediatamente. Questa impostazione ignora l'impostazione Apply immediately (Applica immediatamente).

AWS CLI

Per modificare le impostazioni di scalabilità automatica dello storage per un'istanza DB, usa il comando. AWS CLI `modify-db-instance` Imposta il seguente parametro:

- `--max-allocated-storage`: imposta il limite massimo delle dimensioni dell'archiviazione, in gibibyte. Se il valore è superiore al parametro `--allocated-storage`, l'Auto Scaling dello storage viene attivato. Se il valore equivale al parametro `--allocated-storage`, l'Auto Scaling dello storage viene disattivato.

Per verificare che lo storage autoscaling di Amazon RDS sia disponibile per la tua istanza DB, usa il comando. AWS CLI `describe-valid-db-instance-modifications` Per effettuare il controllo in base alla classe dell'istanza prima della creazione dell'istanza, utilizza il comando `describe-orderable-db-instance-options`. Controlla il seguente campo nel valore restituito:

- `SupportsStorageAutoscaling`: indica se l'istanza database supporta la scalabilità automatica dell'archiviazione.

Per ulteriori informazioni sullo storage, consultare Storage delle istanze di database Amazon RDS.

API RDS

Per modificare le impostazioni dell'Auto Scaling dello storage per un'istanza database, utilizza l'operazione API di Amazon RDS `ModifyDBInstance`. Imposta il seguente parametro:

- `MaxAllocatedStorage`: imposta il limite massimo delle dimensioni dell'archiviazione, in gibibyte.

Per verificare che l'Auto Scaling dello storage Amazon RDS sia disponibile per l'istanza database, utilizza l'operazione API di Amazon RDS `DescribeValidDbInstanceModifications` per un'istanza esistente o l'operazione `DescribeOrderableDBInstanceOptions` prima della creazione di un'istanza. Controlla il seguente campo nel valore restituito:

- `SupportsStorageAutoscaling`: indica se l'istanza database supporta la scalabilità automatica dell'archiviazione.

Per ulteriori informazioni sullo storage, consultare Storage delle istanze di database Amazon RDS.

Disabilitazione dell'Auto Scaling per una nuova istanza database

Se Amazon RDS non è più necessario per aumentare automaticamente lo storage per un'istanza database Amazon RDS, puoi disattivare l'Auto Scaling dello storage. Dopo aver eseguito questa operazione, puoi ancora aumentare manualmente la quantità di storage per l'istanza database.

Console

Come disattivare l'Auto Scaling dello storage per un'istanza database

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, scegliere Databases (Database).
3. Scegliere l'istanza database che si desidera modificare e scegliere Modify (Modifica). Viene visualizzata la pagina Modify DB Instance (Modifica istanza database).
4. Deselezionare la casella di controllo Enable storage autoscaling (Abilita Auto Scaling dello storage) nella sezione Storage autoscaling (Auto Scaling dello storage). Per ulteriori informazioni, consulta Modifica di un'istanza database Amazon RDS.
5. Quando tutte le modifiche sono come si desidera, scegliere Continue (Continua) e controllare il riepilogo delle modifiche.
6. Nella pagina di conferma esaminare le modifiche. Se sono corrette, scegliere Modify DB Instance (Modifica istanza database) per salvare le modifiche. Se non sono corrette, scegliere Back (Indietro) per cambiare le modifiche o Cancel (Annulla) per annullare le modifiche.

Le modifiche al limite di storage dell'Auto Scaling vengono eseguite immediatamente. Questa impostazione ignora l'impostazione Apply immediately (Applica immediatamente).

AWS CLI

Per disattivare la scalabilità automatica dello storage per un'istanza DB, usa il AWS CLI comando `modify-db-instance`e il seguente parametro:

- `--max-allocated-storage`: specifica un valore equivalente all'impostazione `--allocated-storage` per prevenire una ulteriore scalabilità automatica dello spazio di archiviazione Amazon RDS per l'istanza database specificata.

Per ulteriori informazioni sullo storage, consultare Storage delle istanze di database Amazon RDS.

API RDS

Per disattivare l'Auto Scaling dello storage per un'istanza database, utilizza l'operazione API di Amazon RDS `ModifyDBInstance`. Imposta il seguente parametro:

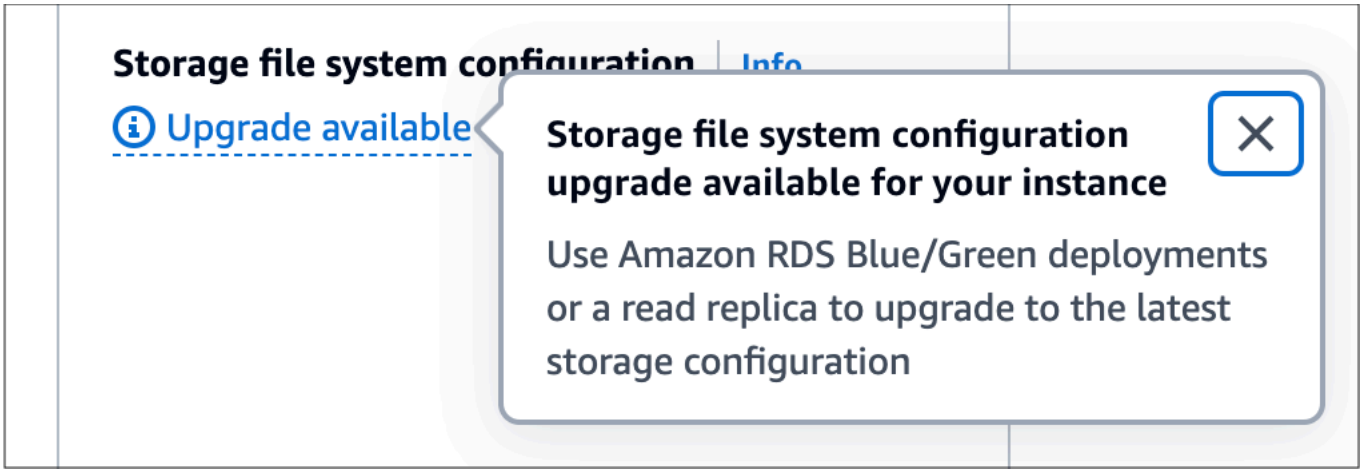
- `MaxAllocatedStorage`: specifica un valore equivalente all'impostazione `AllocatedStorage` per prevenire una ulteriore scalabilità automatica dello spazio di archiviazione Amazon RDS per l'istanza database specificata.

Per ulteriori informazioni sullo storage, consultare Storage delle istanze di database Amazon RDS.

Aggiornamento del file system di archiviazione per un'istanza database

La maggior parte delle istanze DB RDS offre una dimensione di archiviazione massima di 64 TiB per database RDS per MariaDB, MySQL e PostgreSQL. Tuttavia, alcuni file system a 32 bit meno recenti hanno capacità di archiviazione inferiori. Per determinare la capacità di archiviazione dell'istanza DB, è possibile utilizzare il comando -modifications. describe-valid-db-instance AWS CLI

Se RDS rileva che una delle istanze database esegue un file system meno recente (con una dimensione di archiviazione di 16 TiB, un limite di dimensione del file di 2 TiB o scritture non ottimizzate), la console RDS informa l'utente che la configurazione del file system è idonea per un aggiornamento. È possibile verificare l'idoneità all'aggiornamento dell'istanza database nel pannello Archiviazione della pagina dei dettagli dell'istanza database.



Se l'istanza database è idonea per un aggiornamento del file system, puoi eseguire l'aggiornamento in due modi:

- Crea un'implementazione blu/verde e specifica Aggiorna la configurazione del file system di archiviazione. Questa opzione aggiorna il file system alla configurazione preferita nell'ambiente verde. Quindi puoi eseguire lo switchover all'implementazione blu/verde, che rende l'ambiente verde il nuovo ambiente di produzione. Per istruzioni dettagliate, vedi the section called “Creazione di un'implementazione blu/verde”.

- Crea una replica di lettura dell'istanza database e specifica Aggiorna la configurazione del file system di archiviazione. Questa opzione aggiorna il file system della replica di lettura alla configurazione preferita. È possibile promuovere la replica di lettura a un'istanza autonoma. Per istruzioni dettagliate, vedi the section called “Creazione di una replica di lettura”.

L'aggiornamento della configurazione di archiviazione è un'operazione che richiede un elevato livello di I/O e comporta tempi di creazione lunghi per le repliche di lettura e le implementazioni blu/verdi. Il processo di aggiornamento dello storage è più rapido se l'istanza DB di origine utilizza lo storage Provisioned IOPS SSD (io1 o io2 Block Express) e hai effettuato il provisioning dell'ambiente verde o una replica di lettura con una dimensione dell'istanza pari o superiore a 4 volte. Gli aggiornamenti dell'archiviazione su volumi SSD per scopi generici (gp2) possono far esaurire il saldo dei crediti di I/O; se ciò si verifica, il processo di aggiornamento diventa più lungo. Per ulteriori informazioni, consulta the section called “Storage delle istanze database”.

Durante il processo di aggiornamento dell'archiviazione, il motore di database non è disponibile. Se l'utilizzo dell'archiviazione sull'istanza database di origine è maggiore o uguale al 90% della dimensione dell'archiviazione allocata, il processo di aggiornamento dell'archiviazione aumenterà la dimensione dell'archiviazione allocata del 10% per l'istanza verde o la replica di lettura.

Modifica delle impostazioni dell'archiviazione SSD con capacità di IOPS allocata

Puoi modificare le impostazioni per un'istanza database che utilizza l’archiviazione SSD IOPS con provisioning utilizzando la console Amazon RDS, AWS CLI o l'API Amazon RDS. Specifica il tipo di storage, lo storage allocato e la quantità di Provisioned IOPS necessaria. L'intervallo dipende dal motore del database e dal tipo di istanza.

Anche se puoi ridurre la quantità di capacità di IOPS allocata per l'istanza, non puoi ridurre la quantità di archiviazione.

Nella maggior parte dei casi, lo storage con dimensionamento non richiede alcuna interruzione e non influisce negativamente sulle prestazioni del server. Dopo aver modificato l'IOPS storage per un'istanza database, lo stato passa a storage-optimization (ottimizzazione-storage).

> **Note**
>
> L'ottimizzazione dello spazio di archiviazione può richiedere alcune ore. Non puoi apportare altre modifiche all'archiviazione prima di sei (6) ore dal completamento dell'ottimizzazione dello spazio di archiviazione nell'istanza.

Per informazioni sugli intervalli dell'archiviazione allocata e della capacità di IOPS allocata disponibili per ogni motore di database, consulta Storage SSD Provisioned IOPS.

Console

Per modificare le impostazioni di Provisioned IOPS per un'istanza database

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, scegliere Databases (Database).

   Per filtrare l'elenco di istanze database, in Filter databases (Filtra database) inserire una stringa di testo che verrà usata da Amazon RDS per filtrare i risultati. Verranno visualizzate solo le istanze database i cui nomi contengono la stringa.
3. Scegliere l'istanza database con Provisioned IOPS che si vuole modificare.
4. Scegliere Modify (Modifica).
5. Nella pagina Modifica dell'istanza DB, scegli Provisioned IOPS SSD (io1) o Provisioned IOPS SSD (io2) per il tipo di storage.
6. Per Provisioned IOPS (Capacità di IOPS allocata), immetti un valore.

   Se il valore specificato per Allocated storage (Storage allocato) o Provisioned IOPS (Capacità di IOPS allocata) supera i limiti supportati dall'altro parametro, viene visualizzato un messaggio di avviso. Il messaggio indica l'intervallo di valori necessario per l'altro parametro.
7. Scegli Continue (Continua).
8. Scegliere Apply immediately (Applica immediatamente) nella sezione Scheduling of modifications (Pianificazione delle modifiche) per applicare immediatamente le modifiche all'istanza database. In alternativa, scegliere Apply during the next scheduled maintenance window (Applica durante la prossima finestra di manutenzione pianificata): per applicare le modifiche durante la prossima finestra di manutenzione.

9. Esaminare i parametri da modificare e scegliere Modify DB instance (Modifica istanza database) per completare la modifica.

   Il nuovo valore per lo storage allocato o per Provisioned IOPS viene visualizzato nella colonna Status (Stato).

AWS CLI

Per modificare l'impostazione Provisioned IOPS per un'istanza DB, usa il comando. AWS CLI `modify-db-instance` Imposta i seguenti parametri:

- `--storage-type`— Impostato su `io1` o `io2` per Provisioned IOPS.
- `--allocated-storage`: la quantità di spazio di archiviazione, in gibibyte, da allocare per l'istanza database.
- `--iops`: la nuova quantità di IOPS con provisioning per l'istanza database, espressa in operazioni di I/O al secondo.
- `--apply-immediately`: utilizza `--apply-immediately` per applicare immediatamente le modifiche. Utilizza `--no-apply-immediately` (impostazione predefinita) per applicare le modifiche durante la finestra di manutenzione successiva.

API RDS

Per modificare le impostazioni di Provisioned IOPS per un'istanza database, utilizza l'operazione API Amazon RDS `ModifyDBInstance`. Imposta i seguenti parametri:

- `StorageType`— Impostato su `io1` o `io2` per Provisioned IOPS.
- `AllocatedStorage`: la quantità di spazio di archiviazione, in gibibyte, da allocare per l'istanza database.
- `Iops`: la nuova frequenza di IOPS per l'istanza database, espressa in operazioni di I/O al secondo.
- `ApplyImmediately`: imposta questa opzione su `True` per applicare immediatamente le modifiche. Imposta questa opzione su `False` (impostazione predefinita) per applicare le modifiche durante la finestra di manutenzione successiva.

Modifiche dello spazio di archiviazione con uso intensivo di I/O

Le istanze database Amazon RDS usano volumi Amazon Elastic Block Store (EBS) per l'archiviazione di registri e database. A seconda della quantità di archiviazione richiesta, RDS, ma non RDS per SQL Server, esegue automaticamente lo striping su più volumi Amazon EBS per migliorare le prestazioni. Le istanze database RDS con tipi di archiviazione SSD sono supportate da uno o quattro volumi Amazon EBS con striping in una configurazione RAID 0. In base alla progettazione, le operazioni di modifica dello spazio di archiviazione per un'istanza database RDS hanno un impatto minimo sulle operazioni correnti del database.

Nella maggior parte dei casi, le modifiche alla scalabilità dello spazio di archiviazione vengono completamente trasferite al livello Amazon EBS e sono trasparenti per il database. Questo processo in genere viene completato in pochi minuti. Tuttavia, alcuni volumi di archiviazione RDS meno recenti richiedono un processo diverso per modificare le dimensioni, la capacità di IOPS allocata o il tipo di archiviazione. Ciò comporta la creazione di una copia completa dei dati utilizzando un'operazione potenzialmente con uso intensivo di I/O.

La modifica dello spazio di archiviazione utilizza un'operazione con uso intensivo di I/O se si verifica uno dei seguenti fattori:

- Il tipo di archiviazione di origine è magnetico. L'archiviazione magnetica non supporta la modifica elastica del volume.
- L'istanza database RDS non si trova su un layout Amazon EBS a uno o quattro volumi. Puoi visualizzare il numero di volumi Amazon EBS in uso sulle tue istanze database RDS utilizzando le metriche di monitoraggio avanzato. Per ulteriori informazioni, consulta Visualizzazione dei parametri nella console RDS.
- La dimensione di destinazione della richiesta di modifica aumenta lo spazio di archiviazione allocato a 400 GB per le istanze RDS per MariaDB, MySQL e PostgreSQL e a 200 GiB per RDS per Oracle. Le operazioni di dimensionamento automatico dello spazio di archiviazione hanno lo stesso effetto quando aumentano le dimensioni dello spazio allocato dell'istanza database al di sopra di queste soglie.

Se la modifica dello spazio di archiviazione comporta un'operazione che richiede un uso intensivo di I/O, tale modifica consuma risorse di I/O e aumenta il carico sull'istanza database. Le modifiche dello spazio di archiviazione con uso intensivo di I/O che coinvolgono l'archiviazione SSD per scopi generici (gp2) possono far esaurire il saldo dei crediti di I/O, con conseguenti tempi di conversione più lunghi.

Come best practice, consigliamo di pianificare queste richieste di modifica dello spazio di archiviazione non nelle ore di punta per ridurre il tempo necessario per completare l'operazione. In alternativa, puoi creare una replica di lettura dell'istanza database ed eseguire la modifica dello spazio di archiviazione sulla replica di lettura. Quindi promuovi la replica di lettura a istanza primaria. Per ulteriori informazioni, consulta Uso delle repliche di lettura dell'istanza database.

Per ulteriori informazioni, consulta la pagina relativa al motivo del blocco di un'istanza database Amazon RDS nello stato di modifica durante il tentativo di incremento dello spazio di archiviazione allocato.

Modifica delle impostazioni dell'archiviazione SSD per uso generico (gp3)

Puoi modificare le impostazioni per un'istanza DB che utilizza lo storage General Purpose SSD (gp3) utilizzando la console Amazon RDS AWS CLI o l'API Amazon RDS. Specifica il tipo di archiviazione, lo spazio di archiviazione allocato e la quantità di capacità di IOPS allocata e la velocità di trasmissione effettiva dell'archiviazione necessari.

Sebbene sia possibile ridurre la quantità di Provisioned IOPS e il throughput di storage per l'istanza DB, non è possibile ridurre le dimensioni dello storage.

Nella maggior parte dei casi, la scalabilità dell'archiviazione non richiede alcuna interruzione. Dopo aver modificato l'IOPS storage per un'istanza database, lo stato passa a storage-optimization (ottimizzazione-storage). Puoi aspettarti latenze elevate, ma comunque nell'intervallo di millisecondi a una cifra, durante l'ottimizzazione dell'archiviazione. Dopo una modifica dello storage, l'istanza database è completamente operativa.

> **Note**
>
> Non puoi apportare altre modifiche allo storage prima di sei (6) ore dal completamento dell'ottimizzazione dello storage nell'istanza.

Per informazioni sugli intervalli dello spazio di archiviazione allocato, della capacità di IOPS allocata e della velocità di trasmissione effettiva dell'archiviazione disponibili per ogni motore di database, consulta archiviazione gp3 (consigliata).

Console

Per modificare le impostazioni delle prestazioni dell'archiviazione per un'istanza database

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, scegliere Databases (Database).

   Per filtrare l'elenco di istanze database, in Filter databases (Filtra database) inserire una stringa di testo che verrà usata da Amazon RDS per filtrare i risultati. Verranno visualizzate solo le istanze database i cui nomi contengono la stringa.
3. Scegli l'istanza database con l'archiviazione gp3 che vuoi modificare.
4. Scegli Modifica.
5. Nella pagina Modify DB Instance (Modifica istanza database), scegli General Purpose SSD (gp3) (SSD per uso generico (gp3)) come Storage type (Tipo di archiviazione), quindi procedi come segue:

   a. Per Provisioned IOPS (Capacità di IOPS allocata), scegli un valore.

      Se il valore specificato per Allocated storage (Storage allocato) o Provisioned IOPS (Capacità di IOPS allocata) supera i limiti supportati dall'altro parametro, viene visualizzato un messaggio di avviso. Il messaggio indica l'intervallo di valori necessario per l'altro parametro.

   b. Per Storage throughput (Velocità di trasmissione effettiva di archiviazione), scegli un valore.

      Se il valore specificato per Provisioned IOPS (Capacità di IOPS allocata) o Storage throughput (Velocità di trasmissione effettiva di archiviazione) supera i limiti supportati dall'altro parametro, viene visualizzato un messaggio di avviso. Il messaggio indica l'intervallo di valori necessario per l'altro parametro.
6. Scegli Continue (Continua).
7. Scegliere Apply immediately (Applica immediatamente) nella sezione Scheduling of modifications (Pianificazione delle modifiche) per applicare immediatamente le modifiche all'istanza database. In alternativa, scegliere Apply during the next scheduled maintenance window (Applica durante la prossima finestra di manutenzione pianificata): per applicare le modifiche durante la prossima finestra di manutenzione.
8. Esaminare i parametri da modificare e scegliere Modify DB instance (Modifica istanza database) per completare la modifica.

Il nuovo valore per la capacità di IOPS allocata viene visualizzato nella colonna Status (Stato).

AWS CLI

Per modificare le impostazioni delle prestazioni di archiviazione per un'istanza DB, usa il AWS CLI comando `modify-db-instance`. Imposta i seguenti parametri:

- `--storage-type`: impostato su `gp3` per SSD per uso generico (gp3).
- `--allocated-storage`: la quantità di spazio di archiviazione, in gibibyte, da allocare per l'istanza database.
- `--iops`: la nuova quantità di IOPS con provisioning per l'istanza database, espressa in operazioni di I/O al secondo.
- `--storage-throughput`— Il nuovo throughput di archiviazione per l'istanza DB, espresso in MiBps.
- `--apply-immediately`: utilizza `--apply-immediately` per applicare immediatamente le modifiche. Utilizza `--no-apply-immediately` (impostazione predefinita) per applicare le modifiche durante la finestra di manutenzione successiva.

API RDS

Per modificare le impostazioni delle prestazioni dell'archiviazione per un'istanza database, utilizza l'operazione API di Amazon RDS `ModifyDBInstance`. Imposta i seguenti parametri:

- `StorageType`: impostato su `gp3` per SSD per uso generico (gp3).
- `AllocatedStorage`: la quantità di spazio di archiviazione, in gibibyte, da allocare per l'istanza database.
- `Iops`: la nuova frequenza di IOPS per l'istanza database, espressa in operazioni di I/O al secondo.
- `StorageThroughput`— Il nuovo throughput di archiviazione per l'istanza DB, espresso in. MiBps
- `ApplyImmediately`: imposta questa opzione su `True` per applicare immediatamente le modifiche. Imposta questa opzione su `False` (impostazione predefinita) per applicare le modifiche durante la finestra di manutenzione successiva.

Utilizzo di un volume di log dedicato (DLV)

Puoi utilizzare un volume di log dedicato (DLV) per un'istanza DB che utilizza lo storage Provisioned IOPS (PIOPS). Un DLV sposta i log delle transazioni del database PostgreSQL e i redo log e i log binari di MySQL/MariaDB su un volume di archiviazione separato dal volume contenente le tabelle del database. Un DLV rende il log di scrittura delle transazioni più efficiente e coerente. I DLV sono ideali per database con archiviazione allocata di grandi dimensioni, requisiti di I/O al secondo (IOPS) elevati o carichi di lavoro sensibili alla latenza.

I DLV sono supportati per lo storage PIOPS (io1 e io2 Block Express) e vengono creati con una dimensione fissa di 1.000 GiB e 3.000 Provisioned IOPS.

Amazon RDS supporta tutti i DLV Regioni AWS per le seguenti versioni:

- MariaDB 10.6.7 e versioni successive alla 10
- MySQL 8.0.28 e versioni successive alla 8
- PostgreSQL 13.10 e versioni successive alla 13, 14.7 e versioni successive alla 14, 15.2 e versioni successive alla 15

RDS supporta i DLV con le implementazioni multi-AZ. Quando modifichi o crei un'istanza Multi-AZ, viene creato un DLV sia per l'istanza principale che per quella secondaria.

RDS supporta i DLV con le repliche di lettura. Se l'istanza database primaria ha un DLV abilitato, anche tutte le repliche di lettura create dopo aver abilitato il DLV avranno un DLV. Tutte le repliche di lettura create prima del passaggio al DLV non saranno abilitate a meno che non vengano modificate esplicitamente in tal senso. Si consiglia inoltre di modificare manualmente tutte le repliche di lettura collegate a un'istanza primaria prima dell'abilitazione del DLV per includere un DLV.

> **Note**
>
> I volumi di log dedicati sono consigliati per configurazioni di database di almeno 5 TiB.

Per informazioni sugli intervalli dello spazio di archiviazione allocato, della capacità di IOPS allocata e della velocità di trasmissione effettiva dell'archiviazione disponibili per ogni motore di database, consulta Storage SSD Provisioned IOPS.

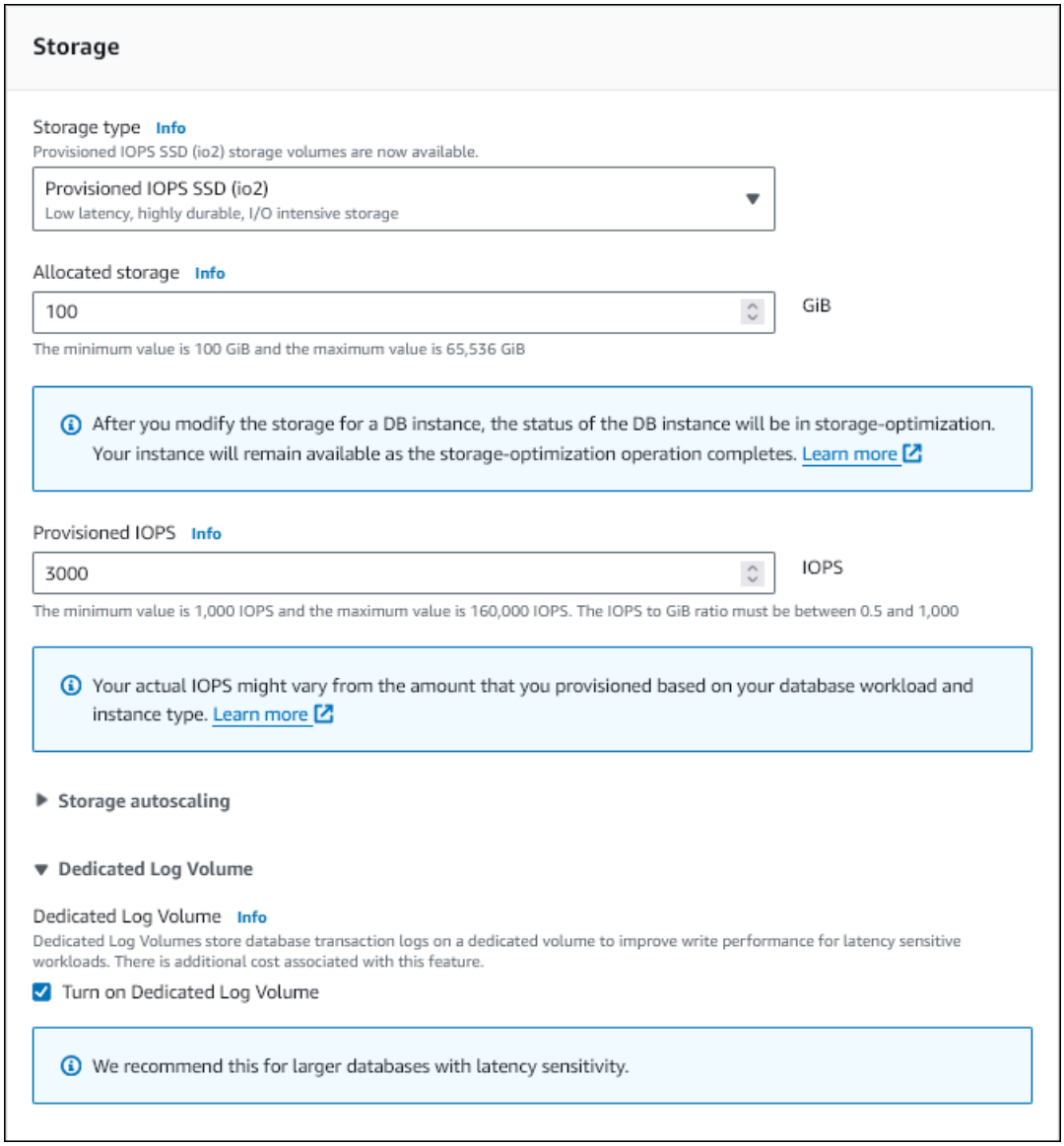
Abilitazione di DLV quando si crea un'istanza DB

È possibile utilizzare l'API AWS Management Console AWS CLI, o RDS per creare un'istanza DB con DLV abilitato.

Console

Per abilitare DLV su una nuova istanza DB

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Scegliere Crea database.
3. Nella pagina Crea un'istanza DB, scegli un motore di database che supporti DLV.
4. Per l'archiviazione:
    a. Scegli Provisioned IOPS SSD (io1) o Provisioned IOPS SSD (io2).
    b. Inserisci lo storage allocato e il Provisioned IOPS che desideri.
    c. Espandi Volume di registro dedicato, quindi seleziona Attiva volume di registro dedicato.



5. Scegliete altre impostazioni in base alle esigenze.
6. Scegliere Crea database.

Dopo la creazione del database, il valore per Dedicated Log Volume viene visualizzato nella scheda Configurazione della pagina dei dettagli del database.

CLI

Per abilitare DLV quando crei un'istanza DB utilizzando lo storage Provisioned IOPS, usa il comando. AWS CLI create-db-instance Imposta i seguenti parametri:

- `--dedicated-log-volume`— Abilita un volume di registro dedicato.
- `--storage-type`— Impostato su `io1` o `io2` per Provisioned IOPS.
- `--allocated-storage`: la quantità di spazio di archiviazione, in gibibyte, da allocare per l'istanza database.
- `--iops`— La quantità di Provisioned IOPS per l'istanza DB, espressa in operazioni di I/O al secondo.

API RDS

Per abilitare DLV quando crei un'istanza DB utilizzando lo storage Provisioned IOPS, utilizza l'operazione API Amazon RDS CreateDBInstance. Imposta i seguenti parametri:

- `DedicatedLogVolume`— Impostato per abilitare un volume di registro dedicato. `true`
- `StorageType`— Impostato su `io1` o `io2` per Provisioned IOPS.
- `AllocatedStorage`: la quantità di spazio di archiviazione, in gibibyte, da allocare per l'istanza database.
- `Iops`— La velocità di IOPS per l'istanza DB, espressa in operazioni di I/O al secondo.

Abilitazione di DLV su un'istanza DB esistente

È possibile utilizzare l'API AWS Management Console AWS CLI, o RDS per modificare un'istanza DB per abilitare DLV.

Dopo aver modificato l'impostazione DLV per un'istanza DB, è necessario riavviare l'istanza DB.

Console

Per abilitare DLV su un'istanza DB esistente

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, scegliere Databases (Database).

   Per filtrare l'elenco di istanze database, in Filter databases (Filtra database) inserire una stringa di testo che verrà usata da Amazon RDS per filtrare i risultati. Verranno visualizzate solo le istanze database i cui nomi contengono la stringa.
3. Scegli l'istanza DB con storage Provisioned IOPS che desideri modificare.

4. Scegli Modifica.
5. Nella pagina Modifica dell'istanza DB:
   - Per Storage, espandi Dedicated Log Volume, quindi seleziona Attiva volume di log dedicato.
6. Scegli Continua.
7. Scegli Applica immediatamente per applicare immediatamente le modifiche all'istanza DB. In alternativa, scegliere Apply during the next scheduled maintenance window (Applica durante la prossima finestra di manutenzione pianificata): per applicare le modifiche durante la prossima finestra di manutenzione.
8. Esaminare i parametri da modificare e scegliere Modify DB instance (Modifica istanza database) per completare la modifica.

Il nuovo valore per Dedicated Log Volume viene visualizzato nella scheda Configurazione della pagina dei dettagli del database.

CLI

Per abilitare o disabilitare DLV su un'istanza DB esistente utilizzando lo storage Provisioned IOPS, usa il comando. AWS CLI `modify-db-instance` Imposta i seguenti parametri:

- `--dedicated-log-volume`— Abilita un volume di registro dedicato.

  Utilizza `--no-dedicated-log-volume` (impostazione predefinita) per disabilitare un volume di registro dedicato.
- `--apply-immediately`: utilizza `--apply-immediately` per applicare immediatamente le modifiche.

  Utilizza `--no-apply-immediately` (impostazione predefinita) per applicare le modifiche durante la finestra di manutenzione successiva.

API RDS

Per abilitare o disabilitare il DLV su un'istanza database esistente utilizzando l'archiviazione con capacità di IOPS allocata, usa l'operazione `ModifyDBInstance` dell'API Amazon RDS. Imposta i seguenti parametri:

- `DedicatedLogVolume`— Imposta questa opzione `true` per abilitare un volume di registro dedicato.

Imposta questa opzione per `false` disabilitare un volume di registro dedicato. Si tratta del valore di default.

- `ApplyImmediately`: imposta questa opzione su `True` per applicare immediatamente le modifiche.

  Imposta questa opzione su `False` (impostazione predefinita) per applicare le modifiche durante la finestra di manutenzione successiva.

Eliminazione di un'istanza database

È possibile eliminare un'istanza DB utilizzando l' AWS Management Console API AWS CLI, the o RDS. Per eliminare un'istanza DB in un cluster DB Aurora, consulta Eliminazione di cluster e istanze database di Aurora.

Argomenti

- Prerequisiti per l'eliminazione di un'istanza database
- Considerazioni sull'eliminazione di un'istanza database
- Eliminazione di un'istanza database

Prerequisiti per l'eliminazione di un'istanza database

Prima di provare a eliminare l'istanza database, assicurati che l'opzione Protezione da eliminazione sia disattivata. Per impostazione predefinita, l'opzione Protezione da eliminazione è attivata per un'istanza database creata con la console.

Se l'opzione Protezione da eliminazione è attivata per l'istanza database, puoi disattivarla modificando le impostazioni dell'istanza. Scegli Modifica nella pagina dei dettagli del database o chiama il modify-db-instancecomando. Questa operazione non causa un'interruzione delle attività. Per ulteriori informazioni, consulta Impostazioni per istanze database.

Considerazioni sull'eliminazione di un'istanza database

L'eliminazione di un'istanza database interessa la ripristinabilità dell'istanza, la disponibilità del backup e lo stato della replica di lettura. Considera le problematiche descritte di seguito:

- È possibile decidere se creare uno snapshot DB finale. Sono disponibili le seguenti opzioni:
  - Se crei uno snapshot finale, è possibile utilizzarlo per ripristinare l'istanza database eliminata. RDS conserva sia lo snapshot finale che tutti gli snapshot manuali creati in precedenza. Non puoi creare un'istantanea database finale dell'istanza database se il relativo stato non è `Available`. Per ulteriori informazioni, consulta Visualizzazione dello stato dell'istanza database di Amazon RDS .
  - Se non scatti un'istantanea finale, l'eliminazione dell'istanza è più veloce. Lo svantaggio è che non esiste un'istantanea finale da ripristinare in un secondo momento. Se decidi di ripristinare l'istanza DB eliminata, conserva i backup automatici o utilizza un'istantanea manuale precedente per ripristinare l'istanza DB al momento in cui si trovava la copia istantanea precedente.

- Puoi decidere se mantenere i backup automatici. Sono disponibili le seguenti opzioni:
  - Se decidi di mantenere i backup automatici, RDS li conserva per il periodo di conservazione impostato sull'istanza database al momento dell'eliminazione. Puoi utilizzare i backup automatici per ripristinare l'istanza database nell'intervallo compreso nel periodo di conservazione ma successivamente a esso. Il periodo di conservazione impostato è valido indipendentemente dal fatto che si sia scelto di creare uno snapshot DB finale. Per eliminare i backup automatici mantenuti, consulta Eliminazione dei backup automatici mantenuti.
  - I backup automatici conservati e le istantanee manuali comportano costi di fatturazione fino alla loro eliminazione. Per ulteriori informazioni, consulta Costi di retention.
  - Se non conservi i backup automatici, RDS elimina i backup automatici che risiedono nella stessa istanza DB. Regione AWS Non è possibile recuperare questi backup. Se i backup automatici sono stati replicati in un'altra Regione AWS, RDS li conserva anche se decidi di non conservarli. Per ulteriori informazioni, consulta Replica dei backup automatici su un altro Regione AWS.

  > **Note**
  >
  > In genere non devi mantenere i backup automatici se crei uno snapshot DB finale.

- Quando elimini un'istanza database, RDS non elimina le istantanee database manuali. Per ulteriori informazioni, consulta Creazione di uno snapshot DB per un'istanza DB Single-AZ.
- Se desideri eliminare tutte le risorse RDS, tieni presente che le seguenti risorse comportano costi di fatturazione:
  - Istanze DB
  - Snapshot DB
  - Cluster database

  Se hai acquistato istanze riservate, queste vengono fatturate in base al contratto sottoscritto al momento dell'acquisto dell'istanza. Per ulteriori informazioni, consulta Istanze database riservate per Amazon RDS. Puoi ottenere informazioni di fatturazione per tutte le tue risorse utilizzando il. AWS AWS Cost Explorer Per ulteriori informazioni, consulta Analisi dei costi con. AWS Cost Explorer
- Se elimini un'istanza DB che contiene repliche di lettura nella stessa Regione AWS, ogni replica di lettura viene automaticamente promossa a istanza DB autonoma. Per ulteriori informazioni, consulta Promozione di una replica di lettura a istanza database standalone. Se l'istanza DB ha repliche di lettura in diverse aree geografiche Regioni AWS, consulta Considerazioni relative alla

replica tra regioni le informazioni relative all'eliminazione dell'istanza DB di origine per una replica di lettura interregionale.

- Quando lo stato di un'istanza DB è`deleting`, il relativo valore del certificato CA non viene visualizzato nella console RDS o nell'output dei AWS CLI comandi o delle operazioni dell'API RDS. Per ulteriori informazioni sui certificati CA, consulta .
- Il tempo necessario per eliminare un'istanza database varia a seconda del periodo di conservazione del backup, ovvero a seconda del numero di backup da eliminare, della quantità di dati eliminati e dell'esecuzione di uno snapshot finale.

Eliminazione di un'istanza database

È possibile eliminare un'istanza DB utilizzando AWS Management Console, the o l' AWS CLI API RDS. Completa le attività seguenti:

- Fornire il nome dell'istanza database
- Abilitare o disabilitare l'opzione per acquisire uno snapshot DB finale dell'istanza.
- Abilitare o disabilitare l'opzione per mantenere i backup automatici.

> **Note**
>
> Non puoi eliminare un'istanza database quando l'opzione Protezione da eliminazione è abilitata. Per ulteriori informazioni, consulta Prerequisiti per l'eliminazione di un'istanza database.

Console

Per eliminare un'istanza database

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, scegliere Databases (Database) e selezionare l'istanza database da eliminare.
3. In Actions (Azioni), selezionare Delete (Elimina).
4. Per creare uno snapshot DB finale per l’istanza database, abilitare Create final snapshot? (Crea snapshot finale?).

5. Se si è scelto di creare uno snapshot finale, immettere il Final snapshot name (Nome dello snapshot finale).
6. Per mantenere i backup automatici, scegliere Retain automated backups (Mantieni backup automatici).
7. Immettere **`delete me`** nella casella.
8. Scegliere Delete (Elimina).

AWS CLI

Per trovare gli ID delle istanze DB presenti nel tuo account, chiama il describe-db-instancescomando:

```
aws rds describe-db-instances --query 'DBInstances[*].[DBInstanceIdentifier]' --output
 text
```

Per eliminare un'istanza DB utilizzando il AWS CLI, chiamate il delete-db-instancecomando con le seguenti opzioni:

- `--db-instance-identifier`
- `--final-db-snapshot-identifier` o `--skip-final-snapshot`

Example Con uno snapshot finale e nessun backup automatico mantenuto

Per LinuxmacOS, oUnix:

```
aws rds delete-db-instance \
    --db-instance-identifier mydbinstance \
    --final-db-snapshot-identifier mydbinstancefinalsnapshot \
    --delete-automated-backups
```

Per Windows:

```
aws rds delete-db-instance ^
    --db-instance-identifier mydbinstance ^
    --final-db-snapshot-identifier mydbinstancefinalsnapshot ^
    --delete-automated-backups
```

Example Con backup automatici mantenuti e nessuno snapshot finale

Per LinuxmacOS, oUnix:

```
aws rds delete-db-instance \
    --db-instance-identifier mydbinstance \
    --skip-final-snapshot \
    --no-delete-automated-backups
```

Per Windows:

```
aws rds delete-db-instance ^
    --db-instance-identifier mydbinstance ^
    --skip-final-snapshot ^
    --no-delete-automated-backups
```

API RDS

Per eliminare un'istanza database tramite l'API Amazon RDS, chiamare l'operazione `DeleteDBInstance` con i parametri seguenti:

- `DBInstanceIdentifier`
- `FinalDBSnapshotIdentifier` o `SkipFinalSnapshot`

Configurazione e gestione di un'implementazione multi-AZ

Le implementazioni multi-AZ possono avere una o due istanze database in standby. Quando l'implementazione ha un'istanza DB in standby, viene chiamata una implementazione istanza database Multi-AZ. Un'implementazione di istanze database Multi-AZ ha un'istanza database in standby che fornisce supporto per il failover, ma non serve il traffico di lettura. Quando l'implementazione ha due istanze database in standby, viene chiamata una implementazione cluster di database Multi-AZ. Un'implementazione di cluster di database Multi-AZ dispone di istanze database in standby che forniscono supporto per il failover e possono anche servire il traffico di lettura.

È possibile utilizzare il plugin AWS Management Console per determinare se un'implementazione Multi-AZ è un'implementazione di istanze database Multi-AZ o un'implementazione di cluster di database Multi-AZ. Nel riquadro di navigazione, scegliere Databases (Database), quindi scegliere un DB identifier (Identificatore database).

- Un'implementazione di istanze database Multi-AZ presenta le seguenti caratteristiche:
  - Esiste una sola riga per l'istanza database.
  - Il valore di Role (Ruolo) è Instance (Istanza) o Primary (Prncipale).
  - Il valore di Multi-AZ è Yes (Sì).
- Un'implementazione di cluster di database Multi-AZ presenta le seguenti caratteristiche:
  - È presente una riga a livello di cluster con tre righe di istanza database al di sotto di essa.
  - Per la riga a livello di cluster, il valore di Role (Ruolo) è Multi-AZ DB cluster (Cluster di database Multi-AZ).
  - Per ogni riga a livello di istanza, il valore di Role (Ruolo) è Writer instance (Istanza di scrittura) o Reader instance (Istanza di lettura).
  - Per ogni riga a livello di istanza, il valore di Multi-AZ è 3 Zones (3 zone).

Argomenti

I seguenti argomenti si applicano alle istanze database e ai cluster di database multi-AZ:

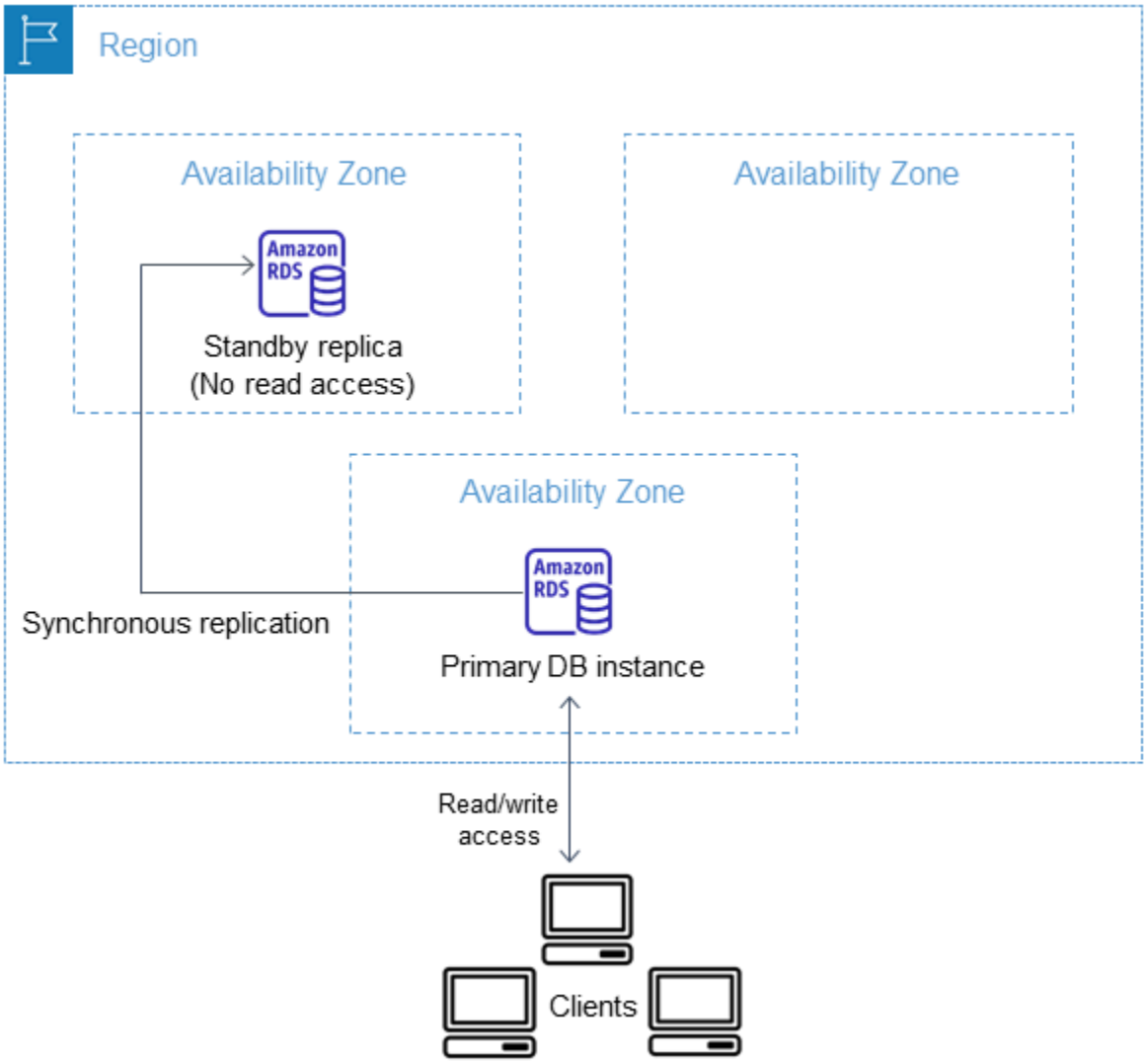
- the section called “Tagging delle risorse RDS”
- the section called “Utilizzo di ARN”
- the section called “Uso dello storage”
- the section called “Manutenzione di un'istanza database”
- the section called “Aggiornamento della versione del motore”

Implementazioni dell'istanza database Multi-AZ

Amazon RDS offre disponibilità elevata e supporto per il failover per le istanze database tramite le implementazioni Multi-AZ con una singola istanza database in standby. Questo tipo di implementazione è chiamato una implementazione di istanza database Multi-AZ. Amazon RDS utilizza varie tecnologie differenti per garantire il supporto per tale failover. Le implementazioni Multi-AZ per le istanze database MariaDB, MySQL, Oracle, PostgreSQL e RDS Custom per SQL Server utilizzano la tecnologia di failover di Amazon. Le istanze database di Microsoft SQL Server utilizzano SQL Server Database Mirroring (DBM) o Always On Availability Groups (AGs). Per informazioni sul supporto della versione di SQL Server per Multi-AZ, consulta Implementazioni Multi-AZ per Amazon RDS for Microsoft SQL Server. Per informazioni sull'utilizzo di RDS Custom per SQL Server per le implementazioni Multi-AZ, consulta Gestione di un'implementazione Multi-AZ per RDS Custom per SQL Server.

In un'implementazione istanza database Multi-AZ, Amazon RDS effettua automaticamente il provisioning e mantiene una replica in standby sincrona in un'altra zona di disponibilità. L'istanza database primaria viene replicata in modo sincrono nelle zone di disponibilità su una replica in standby per fornire ridondanza dati e ridurre al minimo i picchi di latenza durante i backup di sistema. L'esecuzione di un'istanza database con disponibilità elevata può migliorare la disponibilità durante la manutenzione pianificata del sistema. Consente inoltre di proteggere i database da errori dell'istanza database e interruzioni relative alle zone di disponibilità. Per ulteriori informazioni sulle zone di disponibilità, consulta Regioni, zone di disponibilità e Local Zones.

> **Note**
>
> L'opzione di disponibilità elevata non è una soluzione di dimensionamento per scenari di sola lettura. Non è possibile utilizzare una replica in standby per gestire il traffico di lettura. Per utilizzare il traffico di sola lettura, utilizzare invece un cluster di database Multi-AZ o una replica di lettura. Per ulteriori informazioni sui cluster di database Multi-AZ, consulta Implementazioni cluster di database multi-AZ. Per ulteriori informazioni sulle repliche di lettura, consulta Uso delle repliche di lettura dell'istanza database.



Utilizzando la console RDS, puoi creare un'implementazione istanza database Multi-AZ semplicemente specificando Multi-AZ durante la creazione di un'istanza database. Puoi utilizzare la console per convertire le istanze database esistenti in implementazioni istanza database Multi-AZ, modificando l'istanza database e specificando l'opzione Multi-AZ. Puoi anche specificare una distribuzione di istanze DB Multi-AZ con l'API AWS CLI o Amazon RDS. Utilizzate il comando create-db-instanceo modify-db-instanceCLI o l'operazione API CreateDBInstance o ModifyDBInstance.

La console RDS mostra la zona di disponibilità della replica di standby, (denominata zona di disponibilità secondaria). Puoi anche utilizzare il comando describe-db-instancesCLI o l'operazione API DescribedBInstances per trovare la AZ secondaria.

Le istanze database che utilizzano implementazioni Multi-AZ possono avere una latenza di scrittura e di commit maggiore rispetto a un'implementazione Single-AZ. Ciò può accadere a causa della replica sincrona dei dati che si verifica. È possibile che si verifichi una modifica della latenza se la

distribuzione esegue il failover sulla replica di standby, sebbene AWS sia progettata con connettività di rete a bassa latenza tra le zone di disponibilità. Per carichi di lavoro di produzione, è consigliabile utilizzare IOPS con provisioning (input/output operations per second, operazioni di input/output al secondo) per prestazioni veloci e coerenti. Per altre informazioni sulle classi di istanza database, consulta Classi di istanze database.

Trasformazione di un'istanza database in implementazione d'istanza database Multi-AZ

Se si dispone di un'istanza database in un'implementazione Single-AZ e la si modifica in un'implementazione Multi-AZ (per motori diversi da Amazon Aurora), Amazon RDS esegue le operazioni descritte di seguito:

1. Creazione di uno snapshot dei volumi Amazon Elastic Block Store (EBS) dell'istanza database primaria.
2. Creazione di nuovi volumi per la replica in standby basati sullo snapshot. Questi volumi vengono inizializzati in background e le massime prestazioni del volume vengono raggiunte dopo la completa inizializzazione dei dati.
3. Attivazione della replica sincrona a livello di blocco tra i volumi delle repliche primarie e in standby.

> **Important**
>
> L'utilizzo di uno snapshot per la creazione dell'istanza in standby evita tempi di inattività durante la conversione da Single-AZ ad Multi-AZ. Tuttavia si può verificare una riduzione delle prestazioni durante e dopo la conversione in Multi-AZ. Questo impatto può essere significativo per carichi di lavoro sensibili alla latenza di scrittura.
>
> Sebbene consenta di ripristinare rapidamente grandi volumi di dati da snapshot, questa funzionalità può causare un aumento significativo della latenza delle operazioni I/O a causa della replica sincrona. Questa latenza può compromettere le prestazioni del database. Come best practice, si consiglia vivamente di non eseguire la conversione Multi-AZ su un'istanza database di produzione.
>
> Per evitare ripercussioni sulle prestazioni dell'istanza database che gestisce il carico di lavoro sensibile, crea una replica di lettura e abilita i backup su tale replica. Converti la replica di lettura in Multi-AZ ed esegui query che caricano i dati nei volumi della replica di lettura (su entrambe le zone di disponibilità). Quindi promuovi la replica di lettura a istanza primaria. Per ulteriori informazioni, consulta Uso delle repliche di lettura dell'istanza database.

Sono disponibili due modi per modificare un'istanza database in un'implementazione di istanza database multi-AZ:

Argomenti

- Conversione in implementazione di istanza database multi-AZ mediante la console RDS
- Trasformazione di un'istanza database in implementazione d’istanza database Multi-AZ

Conversione in implementazione di istanza database multi-AZ mediante la console RDS

Puoi utilizzare la console RDS per convertire un'istanza database in un'implementazione di istanza database multi-AZ.

Per completare la conversione puoi usare solo la console. Per utilizzare la nostra API RDS, AWS CLI segui le istruzioni riportate in. Trasformazione di un'istanza database in implementazione d’istanza database Multi-AZ

Per convertire in implementazione di istanza database multi-AZ mediante la console RDS

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, scegliere Databases (Database) e selezionare l'istanza database da modificare.
3. In Actions (Operazioni) scegli Convert to Multi-AZ deployment (Converti in implementazione multi-AZ).
4. Per applicare le modifiche immediatamente, scegli l'opzione Apply immediately (Applica immediatamente) nella pagina di conferma. La scelta di questa opzione non causa tempi di inattività, ma è possibile riscontrare un impatto sulle prestazioni. In alternativa, puoi scegliere di applicare l'aggiornamento durante la successiva finestra di manutenzione. Per ulteriori informazioni, consulta Impostazione delle modifiche alla pianificazione.
5. Scegli Convert to Multi-AZ (Converti in multi-AZ).

Trasformazione di un'istanza database in implementazione d’istanza database Multi-AZ

Puoi convertire un'istanza database in un'implementazione di un'istanza database multi-AZ nei seguenti modi:

- Utilizzando la console RDS, modifica l'istanza database e imposta Multi-AZ deployment (Implementazione multi-AZ su Yes (Sì).
- Utilizzando AWS CLI, richiama il modify-db-instancecomando e imposta l'`--multi-az`opzione.
- Utilizzando l'API RDS, richiama l'operazione ModifyDBInstance e imposta il parametro `MultiAZ` su `true`.

Per ulteriori informazioni sulla modifica di un'istanza database, consulta Modifica di un'istanza database Amazon RDS. Una volta completata la modifica, Amazon RDS attiva un evento (RDS-EVENT-0025) che indica che il processo è completo. Puoi monitorare gli eventi Amazon RDS. Per ulteriori informazioni sugli eventi di , consulta Utilizzo della notifica degli eventi di Amazon RDS.

Processo di failover per Amazon RDS

Se un’interruzione pianificata o non pianificata dell'istanza database comporta un defect dell’infrastruttura, Amazon RDS passa automaticamente a una replica in standby in un'altra zona di disponibilità, se hai abilitato l'implementazione Multi-AZ. Il tempo necessario per il completamento del failover varia in base all'attività del database e ad altre condizioni presenti quando l'istanza database principale diventa non disponibile. Il failover richiede in genere da 60 a 120 secondi, tempo che può tuttavia aumentare in caso di transazioni di grandi dimensioni o di un processo di ripristino di lunga durata. Al termine del failover, la modifica della console RDS in base alla nuova zona di disponibilità può richiedere ulteriore tempo.

> **Note**
>
> Puoi forzare un failover manualmente quando riavvii un'istanza database. Per ulteriori informazioni, consulta Riavvio di un'istanza database.

Amazon RDS gestisce i failover automaticamente, in modo da consentirti di riprendere le operazioni database il più rapidamente possibile, senza alcun intervento amministrativo. L'istanza database

principale passa automaticamente alla replica di standby qualora si verifichi una delle condizioni riportate nella seguente tabella. Puoi visualizzare questi motivi di failover nel log di eventi.

Motivo del failover	Descrizione
Al sistema operativo sottostante l'istanza del database RDS viene aggiunta una patch in un'operazione offline.	È stato attivato un failover durante la finestra di manutenzione per una patch del sistema operativo o un aggiornamento di sicurezza. Per ulteriori informazioni, consulta Manutenzione di un'istanza database.
L' host primario dell'istanza Multi-AZ di RDS non è integro.	L'implementazione istanza database Multi-AZ ha rilevato un'istanza database primaria compromessa e ha attivato il failover.
L' host principale dell'istanza Multi-AZ RDS non è raggiungibile a causa della perdita di connettività di rete.	Il monitoraggio RDS ha rilevato un errore di raggiungibilità della rete per l'istanza database primaria e ha attivato un failover.
L' istanza RDS è stata modificata dal cliente.	Una modifica dell'istanza database RDS ha attivato un failover. Per ulteriori informazioni, consulta Modifica di un'istanza database Amazon RDS.
L' istanza principale Multi-AZ RDS è occupata e non risponde.	L'istanza database primaria non risponde. Ti consigliamo anche di completare le seguenti operazioni: • Esamina l'evento e CloudWatch i log per verificare l'utilizzo eccessivo di CPU, memoria o spazio di swap. Per ulteriori informazioni, consulta Utilizzo della notifica degli eventi di Amazon RDS e Creazione di una regola che si attiva su un evento Amazon RDS. •

Motivo del failover	Descrizione
	Valuta il carico di lavoro per determinare se si sta utilizzando la classe di istanza database appropriata. Per ulteriori informazioni, consulta Classi di istanze database. • Utilizza il monitoraggio avanzato per i parametri del sistema operativo in tempo reale. Per ulteriori informazioni, consulta Monitoraggio dei parametri del sistema operativo con il monitoraggio avanzato. • Utilizza Approfondimenti sulle prestazioni per analizzare eventuali problemi che influiscono sulle prestazioni dell'istanza database. Per ulteriori informazioni, consulta Monitoraggio del carico DB con Performance Insights su Amazon RDS. Per ulteriori informazioni su queste raccomandazioni, consulta Panoramica del monitoraggio dei parametri di Amazon RDS e Best practice per Amazon RDS.
Il volume di storage sottostante l'host principale dell'istanza Multi-AZ RDS ha riportato un errore.	L'implementazione dell'istanza database Multi-AZ ha rilevato un problema di archiviazione nell'istanza DB principale e ha avviato il failover.
L'utente ha richiesto un failover dell'istanza database.	È stata riavviata l'istanza database ed è stata scelta l'opzione Riavvia con failover. Per ulteriori informazioni, consulta Riavvio di un'istanza database.

Per determinare se l'istanza database Multi-AZ è soggetta a failover, è possibile eseguire le seguenti operazioni:

- Configura gli abbonamenti a eventi database per inviare una notifica tramite e-mail o SMS in caso di failover. Per ulteriori informazioni sugli eventi di , consulta Utilizzo della notifica degli eventi di Amazon RDS.
- Visualizza gli eventi database utilizzando la console RDS o le operazioni dell'API.
- Visualizza lo stato corrente dell'implementazione dell'istanza database Multi-AZ utilizzando la console RDS o le chiamate API.

Per informazioni su come rispondere ai failover, ridurre i tempi di ripristino e su altre best practice per Amazon RDS, consulta Best practice per Amazon RDS.

Impostazione di JVM TTL per le ricerche del nome DNS

Il meccanismo di failover modifica automaticamente il record Domain Name System (DNS) dell'istanza database in modo da fare riferimento all'istanza database standby. Di conseguenza, sarà necessario ristabilire le connessioni esistenti alla propria istanza database. In un ambiente Java Virtual Machine (JVM), a causa del funzionamento del meccanismo di memorizzazione nella cache DNS Java, potrebbe essere necessario riconfigurare le impostazioni JVM.

La JVM memorizza nella cache le ricerche del nome DNS. Quando la JVM risolve un nome host in un indirizzo IP, memorizza nella cache l'indirizzo IP per un periodo di tempo specificato, noto come (TTL). time-to-live

Poiché AWS le risorse utilizzano voci di nomi DNS che cambiano occasionalmente, si consiglia di configurare la JVM con un valore TTL non superiore a 60 secondi. Questo garantisce che quando l'indirizzo IP di una risorsa cambia, l'applicazione può ricevere e utilizzare il nuovo indirizzo IP della risorsa richiedendo il DNS.

In alcune configurazioni Java, il TTL predefinito di JVM è impostato in modo da non aggiornare mai le voci DNS finché JVM non viene riavviato. Pertanto, se l'indirizzo IP di una AWS risorsa cambia mentre l'applicazione è ancora in esecuzione, non può utilizzare tale risorsa finché non si riavvia manualmente la JVM e le informazioni IP memorizzate nella cache non vengono aggiornate. In questo caso, è fondamentale impostare il valore TTL della JVM in modo che aggiorni periodicamente le informazioni IP memorizzate nella cache.

È possibile ottenere il TTL predefinito della JVM recuperando il valore della proprietà `networkaddress.cache.ttl`:

```
String ttl = java.security.Security.getProperty("networkaddress.cache.ttl");
```

> **Note**
>
> Il valore TTL predefinito può variare in base alla versione della JVM e a seconda che un security manager sia installato o meno. Molte JVM forniscono un TTL predefinito inferiore a 60 secondi. Se utilizzi una JVM di questo tipo e non utilizzi un security manager, puoi ignorare il resto di questo argomento. Per ulteriori informazioni sui security manager in Oracle, consulta The Security Manager nella documentazione di Oracle.

Per modificare la TTL della JVM, imposta il valore della proprietà `networkaddress.cache.ttl`. Utilizza uno dei seguenti metodi, a seconda delle esigenze:

- Per impostare il valore della proprietà a livello globale per tutte le applicazioni che utilizzano la JVM, imposta `networkaddress.cache.ttl` nel file `$JAVA_HOME/jre/lib/security/java.security`.

  ```
  networkaddress.cache.ttl=60
  ```

- Per impostare la proprietà localmente solo per l'applicazione, imposta `networkaddress.cache.ttl` nel codice di inizializzazione dell'applicazione prima che venga stabilita qualsiasi connessione.

  ```
  java.security.Security.setProperty("networkaddress.cache.ttl" , "60");
  ```

Implementazioni cluster di database multi-AZ

Una distribuzione cluster DB Multi-AZ è una modalità di distribuzione semisincrona e ad alta disponibilità di Amazon RDS con due istanze DB di replica leggibili. Un cluster DB Multi-AZ ha un'istanza DB writer e due istanze DB reader in tre zone di disponibilità separate nella stessa Regione AWS. I cluster di database multi-AZ offrono elevata disponibilità, maggiore capacità per i carichi di lavoro in lettura e minore latenza di scrittura rispetto alle implementazioni di istanze database Multi-AZ.

È possibile importare dati da un database on-premise in un cluster database multi-AZ seguendo le istruzioni riportate in Importazione dei dati in un database Amazon RDS MariaDB o MySQL con tempi di inattività ridotti.

Puoi acquistare istanze database riservate per cluster database Multi-AZ. Per ulteriori informazioni, consulta Istanze database riservate per un cluster di database Multi-AZ.

Il supporto varia a seconda delle versioni specifiche di ciascun motore di database e a seconda delle Regioni AWS. Per ulteriori informazioni sulla disponibilità di versioni e regioni per Amazon RDS con cluster di database Multi-AZ, consulta Regioni e motori DB supportati per cluster DB Multi-AZ in Amazon RDS.

Argomenti

- Riavvio di un cluster di database multi-AZ e istanze database di lettura
- Utilizzo delle repliche di lettura del cluster di database multi-AZ
- Utilizzo della replica logica di PostgreSQL con cluster database multi-AZ
- Per eliminare un cluster di database Multi-AZ
- Limitazioni dei cluster DB Multi-AZ

> **Important**
>
> I cluster di database multi-AZ non sono gli stessi dei cluster di database Aurora. Per ulteriori informazioni sull'utilizzo di cluster di database Aurora, consulta la Guida per l'utente di Amazon Aurora.

Disponibilità di classi di istanze per cluster DB Multi-AZ

Le implementazioni di cluster DB Multi-AZ sono supportate per le seguenti classi di istanze DB:`db.m5d`,`db.m6gd`,`db.m6id`,,`db.m6idn`, `db.r5d` `db.r6gddb.x2iedn`, `db.r6id` e e. `db.r6idn` `db.c6gd`

> **Note**
>
> Le classi di istanze c6gd sono le uniche che supportano la dimensione dell'istanza. `medium`

Per altre informazioni sulle classi di istanza database, consulta the section called “Classi di istanze database”.

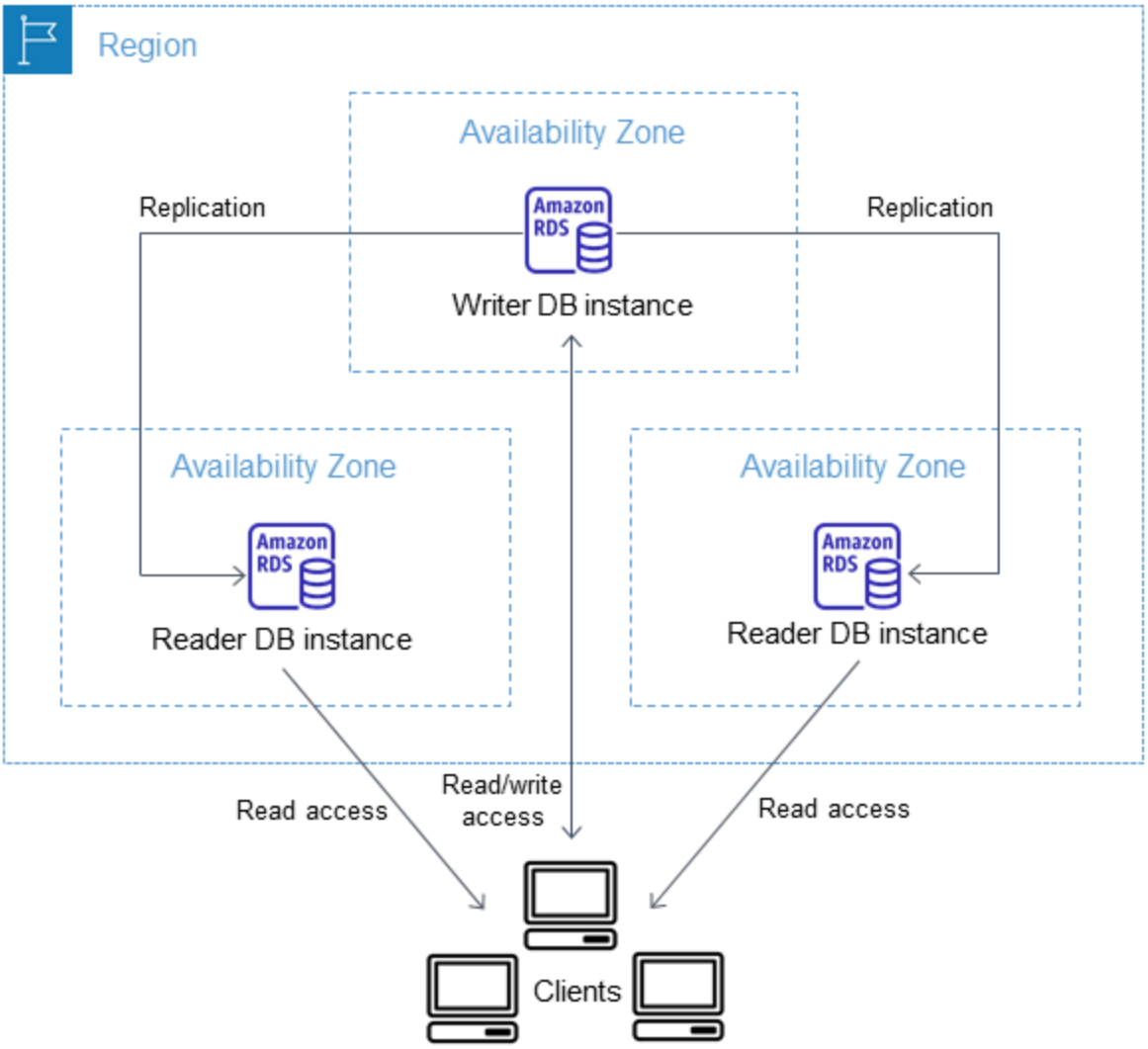
Panoramica dei cluster di database Multi-AZ

Con un cluster di database Multi-AZ, Amazon RDS replica i dati dall'istanza database di scrittore a entrambe le istanze database di lettore utilizzando le funzionalità di replica nativa del motore database. Quando viene apportata una modifica all'istanza database di scrittore, viene inviata a ciascuna istanza database di lettura.

Le implementazioni di cluster Multi-AZ utilizzano la replica semi-sincrona, che richiede la conferma da almeno un'istanza database di lettura affinché venga eseguito il commit di una modifica. Non viene richiesta la conferma dell'avvenuta esecuzione o dell'avvenuto commit degli eventi in tutte le repliche.

Le istanze database di lettore fungono da target di failover automatici e servono anche il traffico di lettura per aumentare il throughput di lettura delle applicazioni. Se si verifica un'interruzione sull'istanza database di scrittura, RDS gestisce il failover su una delle istanze database di lettura. RDS esegue questa operazione in base a quale istanza database di lettura ha il ritardo di replica più recente.

Il seguente diagramma mostra un cluster di database Multi-AZ.



I cluster database Multi-AZ hanno in genere una latenza di scrittura inferiore rispetto alle implementazioni di istanze database AZ multiple. Consentono inoltre l'esecuzione di carichi di lavoro di sola lettura su istanze database di lettura. La console RDS mostra la zona di disponibilità dell'istanza database di scrittore e le zone di disponibilità delle istanze database del lettore. Puoi anche utilizzare il comando describe-db-clustersCLI o l'operazione API DescribedBClusters per trovare queste informazioni.

> **Important**
>
> Per evitare errori di replica nei cluster database multi-AZ RDS per MySQL, consigliamo vivamente di utilizzare una chiave primaria in tutte le tabelle.

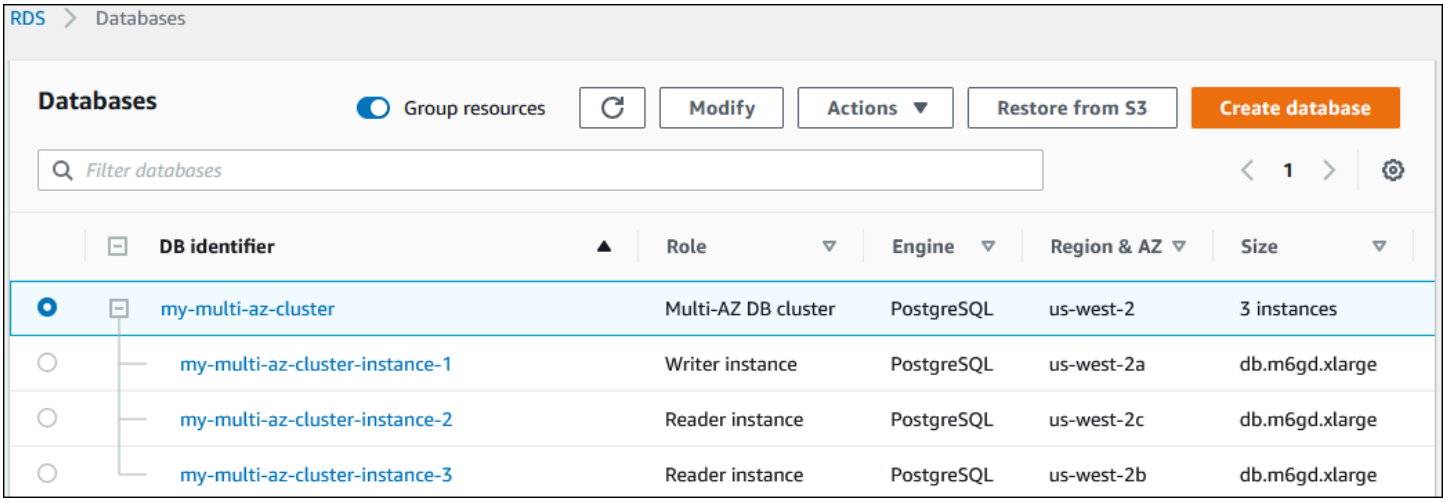
Gestione di un cluster DB Multi-AZ con AWS Management Console

Puoi gestire un cluster di database Multi-AZ con la console.

Gestire un cluster di database Multi-AZ con la console

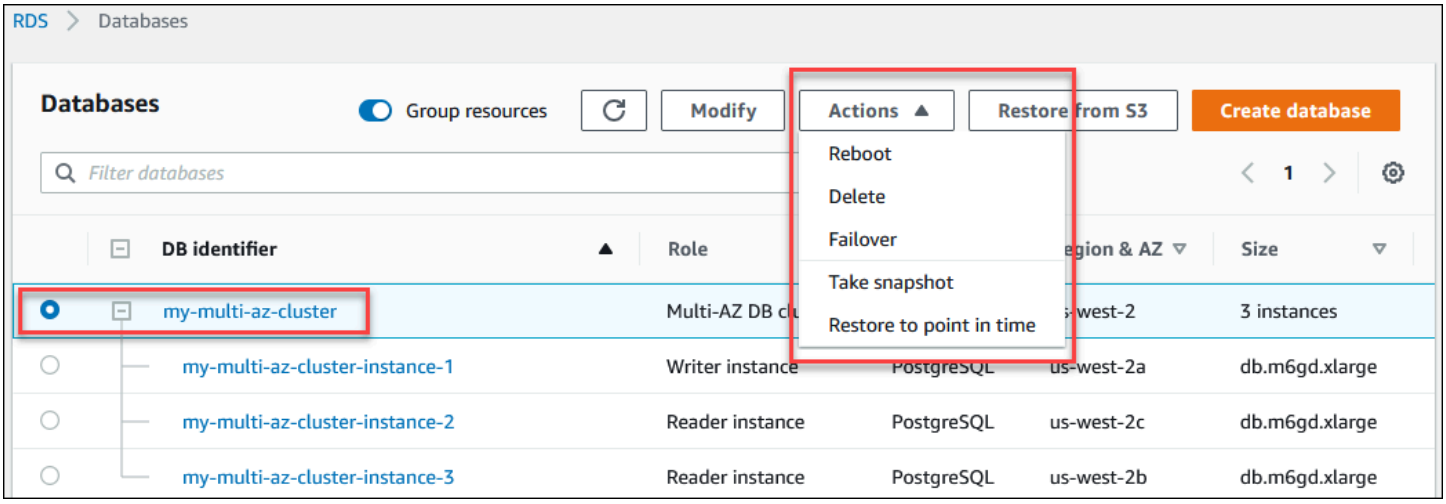
1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel pannello di navigazione, scegliere Databases (Database), quindi scegliere il cluster di database Multi-AZ che si desidera gestire.

L'immagine seguente mostra un cluster di database Multi-AZ nella console.

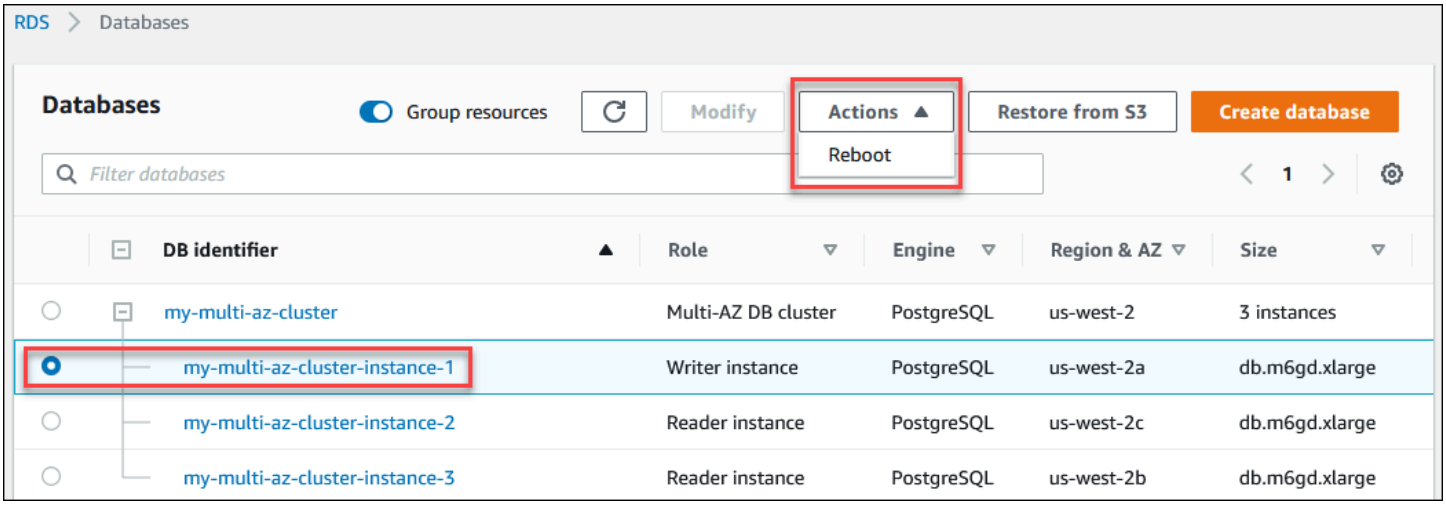


Le azioni disponibili nel menu Actions (Operazioni) dipende dal fatto che siano selezionati o meno il cluster di database Multi-AZ o un'istanza database nel cluster.

Scegliere il cluster di database Multi-AZ per visualizzare i dettagli del cluster ed eseguire operazioni a livello di cluster.



Scegliere un'istanza database in un cluster di database Multi-AZ per visualizzare i dettagli dell'istanza database ed eseguire operazioni a livello di istanza database.



Utilizzo di gruppi di parametri per cluster di database Multi-AZ

In un cluster di database Multi-AZ, un gruppo di parametri del cluster di database funge da container per i valori di configurazione del motore che sono applicati a ogni istanza database in un cluster di database Multi-AZ.

In un cluster di database Multi-AZ, un DB parameter group (Gruppo di parametri database) è impostato sul gruppo di parametri del database predefinito per il motore del database e la versione del motore di database. Le impostazioni del gruppo di parametri del cluster di database vengono utilizzate per tutte le istanze database nel cluster.

Per informazioni sui gruppi di parametri, consultare Utilizzo di gruppi di parametri.

Aggiornamento della versione del motore di un cluster database multi-AZ

Amazon RDS fornisce versioni più recenti di ogni motore di database supportato in modo da poter mantenere aggiornato il cluster DB Multi-AZ. Quando Amazon RDS supporta una nuova versione di un motore di database, puoi scegliere come e quando aggiornare i cluster database multi-AZ.

Esistono due tipi di upgrade che puoi eseguire:

Aggiornamenti delle versioni principali

Un aggiornamento importante della versione del motore può introdurre modifiche non compatibili con le applicazioni esistenti. Quando avvii un aggiornamento di una versione principale, Amazon RDS aggiorna contemporaneamente le istanze Reader e Writer. Pertanto, il cluster DB potrebbe non essere disponibile fino al completamento dell'aggiornamento.

Aggiornamenti di versione minori

Un aggiornamento della versione secondaria include solo modifiche compatibili con le versioni precedenti delle applicazioni esistenti. Quando avvii un aggiornamento di una versione secondaria, Amazon RDS aggiorna innanzitutto le istanze Reader DB una alla volta. Quindi, una delle istanze Reader DB diventa la nuova istanza DB Writer. Amazon RDS aggiorna quindi la vecchia istanza writer (che ora è un'istanza reader).

I tempi di inattività durante l'aggiornamento sono limitati al tempo impiegato da una delle istanze DB di Reader per diventare la nuova istanza DB di Writer. Questo downtime funziona come un failover automatico. Per ulteriori informazioni, consulta the section called “Processo di failover per cluster di database Multi-AZ”. Tieni presente che il ritardo di replica del cluster DB Multi-AZ potrebbe influire sui tempi di inattività. Per ulteriori informazioni, consulta the section called “Ritardo di replica e cluster di database Multi-AZ”.

Per le repliche di lettura del cluster DB RDS per PostgreSQL Multi-AZ, Amazon RDS aggiorna le istanze membri del cluster una alla volta. I ruoli del cluster di lettura e scrittura non cambiano durante l'aggiornamento. Pertanto, il tuo cluster DB potrebbe subire tempi di inattività durante l'aggiornamento dell'istanza di Cluster Writer da parte di Amazon RDS.

> **Note**
>
> Il tempo di inattività per l'aggiornamento di una versione minore di un cluster DB Multi-AZ è in genere di 35 secondi. Se utilizzato con RDS Proxy, è possibile ridurre ulteriormente i tempi di inattività a un secondo o meno. Per ulteriori informazioni, consulta Utilizzo del

Proxy RDS. In alternativa, è possibile utilizzare un proxy di database open source come ProxySQL o il driver PgBouncer AWSJDBC per MySQL.

Attualmente, Amazon RDS supporta gli aggiornamenti delle versioni principali solo per i cluster DB RDS per PostgreSQL Multi-AZ. Amazon RDS supporta aggiornamenti di versione minori per tutti i motori DB che supportano cluster DB Multi-AZ.

Amazon RDS non aggiorna automaticamente le repliche di lettura del cluster database multi-AZ. Per gli aggiornamenti di versioni minori, devi prima aggiornare manualmente tutte le repliche di lettura e quindi aggiornare il cluster. In caso contrario, l'aggiornamento viene bloccato. Quando esegui l'aggiornamento della versione principale di un cluster, lo stato di tutte le repliche di lettura cambia in Terminato. Devi eliminare e ricreare le repliche di lettura al completamento dell'aggiornamento. Per ulteriori informazioni, consulta the section called “Monitoraggio della replica di lettura”.

Il processo di aggiornamento della versione del motore di un cluster database multi-AZ è identico al processo di aggiornamento di una versione del motore di istanze database. Per istruzioni, consulta the section called “Aggiornamento della versione del motore”. L'unica differenza è che quando si usa AWS Command Line Interface (AWS CLI), si usa il modify-db-clustercomando e si specifica il `--db-cluster-identifier` parametro (insieme al `--allow-major-version-upgrade` parametro).

Per ulteriori informazioni sugli aggiornamenti delle versioni principali e secondarie, consultate la seguente documentazione per il motore DB:

- the section called “Aggiornamento del motore del database PostgreSQL”
- the section called “Aggiornamento del motore di database MySQL”

Utilizzo di Server proxy per RDS con cluster di database Multi-AZ

Puoi usare Amazon RDS Proxy per creare un proxy per i tuoi cluster DB Multi-AZ. Utilizzando RDS Proxy, le tue applicazioni possono raggruppare e condividere connessioni al database per migliorare la loro capacità di scalabilità. Ogni proxy esegue il multiplexing delle connessioni, noto anche come riutilizzo delle connessioni. Con il multiplexing, Server proxy per RDS esegue tutte le operazioni per una transazione utilizzando una connessione al database sottostante, RDS Proxy può anche ridurre i tempi di inattività per un aggiornamento di versione minore di un cluster DB Multi-AZ a un secondo o meno. Per ulteriori informazioni sui vantaggi di Server proxy per RDS, consulta Utilizzo del Proxy RDS.

Per configurare un proxy per un cluster di database Multi-AZ, scegli Crea un RDS Proxy durante la creazione del cluster. Per istruzioni su come creare e gestire gli endpoint di Server proxy per RDS, consulta the section called “Utilizzo degli endpoint RDS Proxy”.

Ritardo di replica e cluster di database Multi-AZ

Replica lag (Ritardo di replica) è la differenza di tempo tra l'ultima transazione sull'istanza database di scrittura e l'ultima transazione applicata su un'istanza database di lettura. La CloudWatch metrica Amazon `ReplicaLag` rappresenta questa differenza di fuso orario. Per ulteriori informazioni sulle CloudWatch metriche, consulta. Monitoraggio dei parametri di Amazon RDS con Amazon CloudWatch

Sebbene i cluster di database Multi-AZ consentano prestazioni di scrittura elevate, può comunque verificarsi un ritardo di replica a causa della natura della replica basata sul motore. Poiché qualsiasi failover deve prima risolvere il ritardo di replica prima di promuovere una nuova istanza database di scrittura, il monitoraggio e la gestione di questo ritardo di replica devono essere presi in considerazione.

Per i cluster di database Multi-AZ di RDS for MySQL, il tempo di failover dipende dal ritardo di replica di entrambe le istanze database di lettura rimanenti. Entrambe le istanze database di lettura devono applicare transazioni non applicate prima che una di esse venga promossa a nuova istanza database di scrittura.

Per i cluster di database Multi-AZ di RDS for PostgreSQL, il tempo di failover dipende dal ritardo di replica delle due istanze database di lettura rimanenti. L’istanza database di lettura con il ritardo di replica minore deve applicare transazioni non applicate prima di essere promossa a nuova istanza database di scrittura.

Per un tutorial che mostra come creare un CloudWatch allarme quando il ritardo della replica supera un determinato periodo di tempo, consulta. Tutorial: creazione di un allarme Amazon CloudWatch per il ritardo di replica del cluster di database Multi-AZ

Cause comuni del ritardo di replica

In generale, il ritardo di replica si verifica quando il carico di lavoro in scrittura è troppo alto per consentire alle istanze database di lettura di applicare le transazioni in modo efficiente. Diversi carichi di lavoro possono subire ritardi di replica temporanei o continui. Di seguito sono riportati alcuni esempi di cause comuni:

- Alta concorrenza di scrittura o aggiornamento in batch pesante sull'istanza database di scrittura, che causano il ritardo del processo di applicazione sulle istanze database di lettura.
- Carico di lavoro in lettura pesante che utilizza risorse su una o più istanze database di lettura. L'esecuzione di query lente o di grandi dimensioni può influire sul processo di applicazione e può causare un ritardo di replica.
- Le transazioni che modificano grandi quantità di dati o istruzioni DDL possono talvolta causare un aumento temporaneo del ritardo di replica perché il database deve mantenere l'ordine di commit.

Mitigazione del ritardo di replica

Per i cluster di database Multi-AZ per RDS for MySQL e RDS for PostgreSQL, è possibile ridurre il ritardo di replica riducendo il carico sull'istanza database di scrittura. È inoltre possibile utilizzare il controllo di flusso per ridurre il ritardo di replica. Flow control (Controllo di flusso) funziona limitando le scritture sull'istanza database di scrittura, che garantisce che il ritardo di replica non continui a crescere senza limiti. La limitazione della scrittura viene eseguita aggiungendo un ritardo alla fine di una transazione, che riduce la velocità effettiva di scrittura sull'istanza database di scrittura. Sebbene il controllo di flusso non garantisca l'eliminazione del ritardo, può contribuire a ridurre il ritardo complessivo in molti carichi di lavoro. Le seguenti sezioni forniscono informazioni sull'utilizzo del controllo di flusso con RDS for MySQL e RDS for PostgreSQL.

Mitigazione del ritardo di replica con il controllo di flusso per RDS for MySQL

Quando utilizzi i cluster di database Multi-AZ di RDS for MySQL, il controllo di flusso viene attivato per impostazione predefinita utilizzando il parametro dinamico `rpl_semi_sync_master_target_apply_lag`. Questo parametro specifica il limite superiore desiderato per il ritardo di replica. Man mano che il ritardo di replica si avvicina a questo limite configurato, il controllo del flusso limita le transazioni di scrittura sull'istanza DB di Writer per cercare di contenere il ritardo della replica al di sotto del valore specificato. In alcuni casi, il ritardo di replica può superare il limite specificato. Per impostazione predefinita, questo parametro è impostato su 120 secondi. Per disattivare il controllo del flusso, imposta questo parametro sul valore massimo di 86.400 secondi (un giorno).

Per visualizzare il ritardo corrente inserito dal controllo di flusso, mostra il parametro `Rpl_semi_sync_master_flow_control_current_delay` eseguendo la seguente query.

```
SHOW GLOBAL STATUS like '%flow_control%';
```

L'aspetto dell'output sarà simile al seguente.

```
+-----------------------------------------------+-------+
| Variable_name                                 | Value |
+-----------------------------------------------+-------+
| Rpl_semi_sync_master_flow_control_current_delay | 2010  |
+-----------------------------------------------+-------+
1 row in set (0.00 sec)
```

> **Note**
>
> Il ritardoviene visualizzato in microsecondi.

Quando Performance Insights è attivato per un cluster di database RDS Multi-AZ di RDS for MySQL, è possibile monitorare l'evento di attesa corrispondente a un'istruzione SQL che indica che le query sono state ritardate da un controllo di flusso. Quando un ritardo è stato introdotto da un controllo di flusso, è possibile visualizzare l'evento di attesa `/wait/synch/cond/semisync/semi_sync_flow_control_delay_cond` corrispondente all'istruzione SQL nel pannello di controllo di Performance Insights. Per visualizzare questi parametri, lo schema delle prestazioni deve essere attivato. Per informazioni su Performance Insights, consulta Monitoraggio del carico DB con Performance Insights su Amazon RDS.

Mitigazione del ritardo di replica con il controllo di flusso per RDS for PostgreSQL

Quando utilizzi i cluster di database Multi-AZ di RDS for PostgreSQL, il controllo di flusso viene implementato come estensione. Attiva un dipendente in background per tutte le istanze database nel cluster di database. Per impostazione predefinita, i dipendenti in background sulle istanze database di lettura comunicano il ritardo di replica corrente con il dipendente in background sull'istanza database di scrittura. Se il ritardo supera i due minuti su qualsiasi istanza database di lettura, il dipendente in background sull'istanza database di scrittura aggiunge un ritardo alla fine di una transazione. Per controllare la soglia di ritardo, utilizza il parametro `flow_control.target_standby_apply_lag`.

Quando un controllo di flusso limita un processo PostgreSQL, l'evento di attesa `Extension` in `pg_stat_activity` e Performance Insights lo indica. La funzione `get_flow_control_stats` visualizza i dettagli sull'entità del ritardo attualmente aggiunto.

Il controllo di flusso può beneficiare della maggior parte dei carichi di lavoro OLTP (Online Transaction Processing, elaborazione di transazioni online) che hanno transazioni brevi ma altamente

simultanee. Se il ritardo è causato da transazioni di lunga durata, come le operazioni in batch, il controllo di flusso non fornisce un vantaggio altrettanto forte.

È possibile disattivare il controllo di flusso rimuovendo l'estensione da `shared_preload_libraries` e riavviare l'istanza database.

Processo di failover per cluster di database Multi-AZ

Se si verifica un'interruzione pianificata o non pianificata dell'istanza database di scrittura in un cluster di database Multi-AZ, Amazon RDS esegue automaticamente il failover su un'istanza database di lettura in un'altra zona di disponibilità. Il tempo necessario per il completamento del failover varia in base all'attività del database e ad altre condizioni presenti quando l'istanza database in scrittura diventa non disponibile. Il failover richiede in genere meno di 35 secondi. Il failover viene completato quando entrambe le istanze database del lettore hanno applicato transazioni in sospeso dallo scrittore in errore. Al termine del failover, la modifica della console RDS in base alla nuova zona di disponibilità può richiedere ulteriore tempo.

Argomenti

Failover automatici

Amazon RDS gestisce i failover automaticamente, in modo da consentirti di riprendere le operazioni database il più rapidamente possibile, senza alcun intervento amministrativo. Per eseguire il failover, l'istanza database del scrittore passa automaticamente a un'istanza database del lettore.

Failing manuale su un cluster di database Multi-AZ

Se si esegue manualmente il failover di un cluster DB Multi-AZ, RDS interrompe innanzitutto l'istanza DB principale. Quindi, il sistema di monitoraggio interno rileva che l'istanza DB principale non è integra e promuove un'istanza DB di replica leggibile. Il failover richiede in genere meno di 35 secondi.

È possibile eseguire il failover di un cluster DB Multi-AZ manualmente utilizzando l'API AWS Management Console AWS CLI, the o RDS.

Console

Per eseguire un fail over manuale per un cluster di database Multi-AZ

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, scegli Databases (Database).
3. Scegliere il cluster di database Multi-AZ che si desidera ripristinare.
4. Per Actions (Operazioni), scegliere Failover.

   Viene visualizzata la pagina del cluster Failover DB.
5. Scegliere Failover per confermare il failover manuale.

AWS CLI

Per eseguire manualmente il failover di un cluster DB Multi-AZ, utilizzare il AWS CLI comando. failover-db-cluster

Example

```
aws rds failover-db-cluster --db-cluster-identifier mymultiazdbcluster
```

API RDS

Per eseguire manualmente il failover di un cluster database Multi-AZ, chiamare l'API Amazon RDS FailoverDBCluster e specificare `DBClusterIdentifier`.

Determinare se un cluster di database Multi-AZ ha effettuate un fail over

Per determinare se il cluster database Multi-AZ è soggetto a failover, è possibile eseguire le seguenti operazioni:

- Configura gli abbonamenti a eventi database per inviare una notifica tramite e-mail o SMS in caso di failover. Per ulteriori informazioni sugli eventi di , consulta Utilizzo della notifica degli eventi di Amazon RDS.
- Visualizza gli eventi database utilizzando la console Amazon RDS o le operazioni dell'API.
- Visualizza lo stato attuale del tuo cluster DB Multi-AZ utilizzando la console Amazon RDS, l'API RDS e l' AWS CLI API RDS.

Per informazioni su come rispondere ai failover, ridurre i tempi di ripristino e su altre best practice per Amazon RDS, consulta Best practice per Amazon RDS.

Impostazione di JVM TTL per le ricerche del nome DNS

Il meccanismo di failover modifica automaticamente il record Domain Name System (DNS) dell'istanza database in modo da fare riferimento all'istanza database di lettura. Di conseguenza, sarà necessario ristabilire le connessioni esistenti alla propria istanza database. In un ambiente Java Virtual Machine (JVM), a causa del funzionamento del meccanismo di memorizzazione nella cache DNS Java, potrebbe essere necessario riconfigurare le impostazioni JVM.

La JVM memorizza nella cache le ricerche del nome DNS. Quando la JVM risolve un nome host in un indirizzo IP, memorizza l'indirizzo IP nella cache per un periodo di tempo specificato, noto come (TTL). time-to-live

Poiché AWS le risorse utilizzano voci di nomi DNS che cambiano occasionalmente, si consiglia di configurare la JVM con un valore TTL non superiore a 60 secondi. Questo garantisce che quando l'indirizzo IP di una risorsa cambia, l'applicazione può ricevere e utilizzare il nuovo indirizzo IP della risorsa richiedendo il DNS.

In alcune configurazioni Java, il TTL predefinito di JVM è impostato in modo da non aggiornare mai le voci DNS finché JVM non viene riavviato. Pertanto, se l'indirizzo IP di una AWS risorsa cambia mentre l'applicazione è ancora in esecuzione, non può utilizzare tale risorsa finché non si riavvia manualmente la JVM e le informazioni IP memorizzate nella cache non vengono aggiornate. In questo caso, è fondamentale impostare il valore TTL della JVM in modo che aggiorni periodicamente le informazioni IP memorizzate nella cache.

> **Note**
>
> Il valore TTL predefinito può variare in base alla versione della JVM e a seconda che un security manager sia installato o meno. Molte JVM forniscono un TTL predefinito inferiore a 60 secondi. Se utilizzi una JVM di questo tipo e non utilizzi un security manager, puoi ignorare il resto di questo argomento. Per ulteriori informazioni sui security manager in Oracle, consulta The Security Manager nella documentazione di Oracle.

Per modificare la TTL della JVM, imposta il valore della proprietà `networkaddress.cache.ttl`. Utilizza uno dei seguenti metodi, a seconda delle esigenze:

- Per impostare il valore della proprietà a livello globale per tutte le applicazioni che utilizzano la JVM, imposta `networkaddress.cache.ttl` nel file `$JAVA_HOME/jre/lib/security/java.security`.

```
networkaddress.cache.ttl=60
```

- Per impostare la proprietà localmente solo per l'applicazione, imposta `networkaddress.cache.ttl` nel codice di inizializzazione dell'applicazione prima che venga stabilita qualsiasi connessione.

```
java.security.Security.setProperty("networkaddress.cache.ttl" , "60");
```

Creazione di un cluster di database Multi-AZ

Un cluster di database Multi-AZ ha un'istanza database di scrittore e due istanze database di lettore in tre zone di disponibilità separate. I cluster di database multi-AZ offrono elevata disponibilità, maggiore capacità per i carichi di lavoro in lettura e minore latenza rispetto alle implementazioni Multi-AZ. Per ulteriori informazioni sui cluster di database Multi-AZ, consulta Implementazioni cluster di database multi-AZ.

> **Note**
>
> I cluster di database multi-AZ sono supportati solo per i motori database MySQL e PostgreSQL.

Prerequisiti per i cluster di database

> **Important**
>
> Prima di poter creare un cluster di database Multi-AZ, devi completare le attività descritte in Configurazione di Amazon RDS.

Di seguito sono indicate le procedure preliminari da completare prima di creare un cluster di database Multi-AZ.

Argomenti

- Configurazione della rete per il cluster di database
- Prerequisiti aggiuntivi

Configurazione della rete per il cluster di database

Puoi creare un cluster di database multi-AZ solo in un cloud privato virtuale (VPC) basato sul servizio Amazon VPC. Deve trovarsi in un paese Regione AWS con almeno tre zone di disponibilità. Il gruppo di sottoreti di database scelto per il cluster di database deve coprire almeno tre zone di disponibilità. Questa configurazione garantisce che ogni istanza database del cluster di database si trovi in una zona di disponibilità diversa.

Puoi configurare la connettività tra il tuo nuovo cluster di database e un'istanza Amazon EC2 nello stesso VPC quando crei il cluster di database. Per connetterti al cluster di database da risorse diverse dalle istanze EC2 nello stesso VPC, puoi configurare le connessioni di rete manualmente.

Argomenti

Configurazione della connettività di rete automatica con un'istanza EC2

Quando crei un cluster DB Multi-AZ, puoi utilizzare il AWS Management Console per configurare la connettività tra un'istanza EC2 e il nuovo cluster DB. In questo caso, RDS configura automaticamente il VPC e le impostazioni di rete. Il cluster di database viene creato nello stesso VPC dell'istanza EC2, per consentire all'istanza EC2 di accedere al cluster di database.

Di seguito sono riportati i requisiti per connettere un'istanza EC2 al cluster di database:

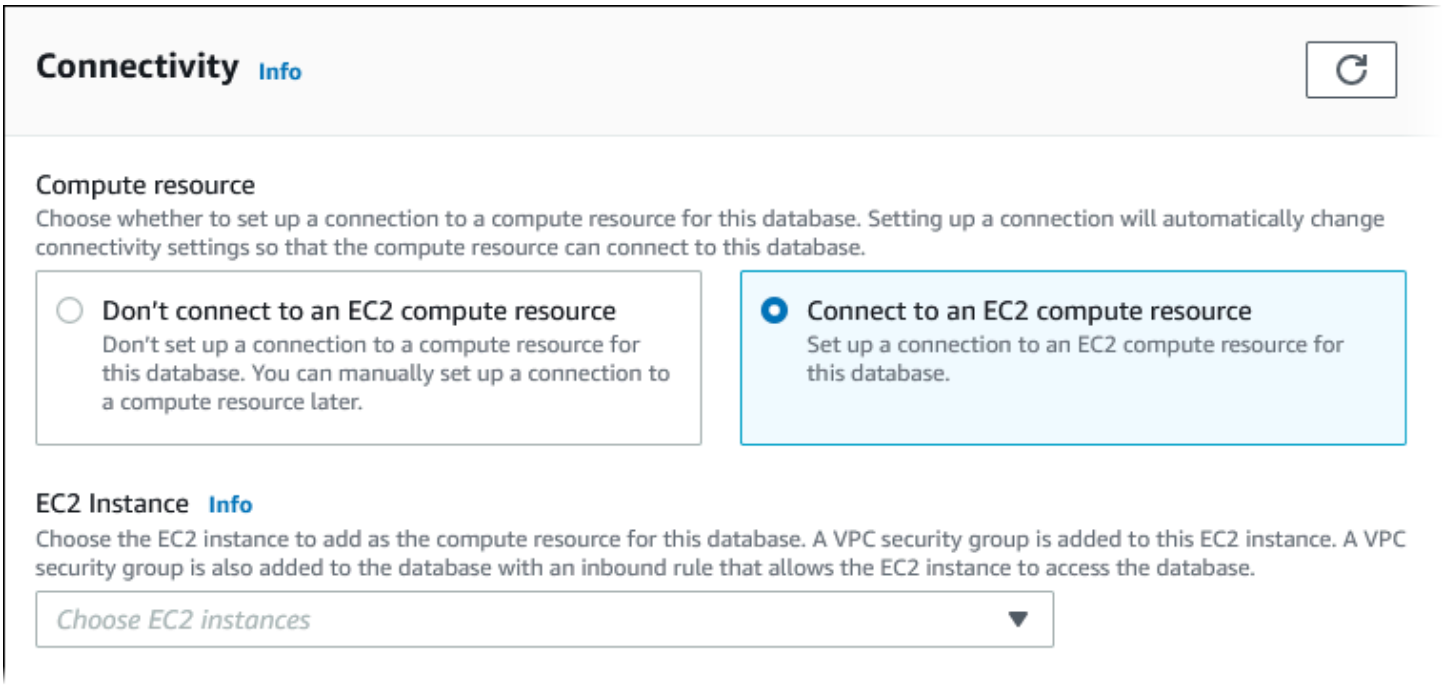
- L'istanza EC2 deve esistere Regione AWS prima di creare il cluster DB.

  Se non esiste alcuna istanza EC2 nel Regione AWS, la console fornisce un collegamento per crearne una.
- L'utente che sta creando il cluster di database deve disporre delle autorizzazioni per eseguire le seguenti operazioni:
  - `ec2:AssociateRouteTable`
  - `ec2:AuthorizeSecurityGroupEgress`
  - `ec2:AuthorizeSecurityGroupIngress`
  - `ec2:CreateRouteTable`
  - `ec2:CreateSubnet`
  - `ec2:CreateSecurityGroup`
  - `ec2:DescribeInstances`
  - `ec2:DescribeNetworkInterfaces`
  - `ec2:DescribeRouteTables`
  - `ec2:DescribeSecurityGroups`
  - `ec2:DescribeSubnets`
  - `ec2:ModifyNetworkInterfaceAttribute`

- `ec2:RevokeSecurityGroupEgress`

L'utilizzo di questa opzione crea un cluster di database privato. Il cluster di database utilizza un gruppo di sottoreti DB con solo sottoreti private per limitare l'accesso alle risorse all'interno del VPC.

Per connettere un'istanza EC2 al cluster di database, scegli Connect to an EC2 compute resource (Connetti a una risorsa di calcolo EC2) nella sezione Connectivity (Connettività) della pagina Create database (Crea database).



Se scegli Connect to an EC2 compute resource (Connetti a una risorsa di calcolo EC2), RDS imposta automaticamente le seguenti opzioni. Queste impostazioni non possono essere modificate a meno che non si scelga di non configurare la connettività con un'istanza EC2 selezionando Don't connect to an EC2 compute resource (Non connetterti a una risorsa di calcolo EC2).

Opzione Console	Impostazione automatica
Virtual Private Cloud (VPC)	RDS imposta il VPC su quello associato all'istanza EC2.
DB subnet group (Gruppo di sottoreti DB)	RDS richiede un gruppo di sottoreti database con una sottorete privata nella stessa zona di disponibilità dell'istanza EC2. Se esiste un gruppo di sottoreti database che soddisfa questo requisito, RDS lo utilizza. Per impostazione predefinita, questa

Opzione Console	Impostazione automatica
	opzione è impostata su Automatic setup (Configurazione automatica). Quando si sceglie Automatic setup (Configurazione automatica) e non esiste un gruppo di sottoreti database che soddisfi questo requisito, viene eseguita la seguente procedura. RDS utilizza tre sottoreti private disponibili in tre zone di disponibilità di cui una è la stessa dell'istanza EC2. Se una sottorete privata non è disponibile in una zona di disponibilità, RDS crea una sottorete privata nella zona di disponibilità. Quindi RDS crea il gruppo di sottoreti database. Quando è disponibile una sottorete privata, RDS utilizza la tabella di instradamento associata alla sottorete e aggiunge tutte le sottoreti create a questa tabella di instradamento. Quando una sottorete privata non è disponibile, RDS crea una tabella di instradamento senza accesso al gateway Internet e aggiunge le sottoreti create alla tabella di instradamento. RDS consente inoltre di utilizzare i gruppi di sottoreti database esistenti. Seleziona Choose existing (Scegli esistente) se desideri utilizzare un gruppo di sottoreti database esistente.
Accesso pubblico	RDS sceglie No in modo che il cluster di database non sia accessibile pubblicamente. Per motivi di sicurezza, come best practice si consiglia di mantenere il database privato e accertarsi che non sia accessibi le da Internet.

Opzione Console	Impostazione automatica
VPC security group (firewall) (Gruppo di sicurezza VPC (firewall)	RDS crea un nuovo gruppo di sicurezza associato al cluster di database. Il gruppo di sicurezza è denominato `rds-ec2-`*n*, dove *n* è un numero. Questo gruppo di sicurezza include una regola in entrata con il gruppo di sicurezza VPC EC2 (firewall) come origine. Questo gruppo di sicurezza associato al cluster di database consente all'istanza EC2 di accedere al cluster. RDS crea, inoltre, un nuovo gruppo di sicurezza associato all'istanza EC2. Il gruppo di sicurezza è denominato `ec2-rds-`*n*, dove *n* è un numero. Questo gruppo di sicurezza include una regola in uscita con il gruppo di sicurezza VPC del cluster di database come origine. Questo gruppo di sicurezza consente all'istanza EC2 di inviare traffico al cluster di database. Puoi aggiungere un nuovo gruppo di sicurezza aggiuntivo scegliendo Create nuovo (Crea nuovo) e digitando il nome del nuovo gruppo di sicurezza. Puoi aggiungere gruppi di sicurezza esistenti scegliendo Choose existing (Scegli esistente) e selezionando i gruppi di sicurezza da aggiungere.
Zona di disponibilità	RDS sceglie la zona di disponibilità dell'istanza EC2 per un'istanza database nell'implementazione del cluster di database Multi-AZ. RDS sceglie casualmente una zona di disponibilità diversa per entrambe le altre istanze database. L'istanza database di scrittura viene creata nella stessa zona di disponibilità dell'istanza EC2. È possibile che vengano addebitati costi aggiuntivi tra zone di disponibilità se si verifica un failover e l'istanza database di scrittura si trova in una zona di disponibilità diversa.

Per ulteriori informazioni su queste impostazioni, consultare Impostazioni per la creazione di cluster di database Multi-AZ.

Se dopo la creazione del cluster di database le impostazioni vengono modificate, le modifiche potrebbero influire sulla connessione tra l'istanza EC2 e il cluster di database.

Configurazione manuale della rete

Per connetterti al cluster di database da risorse diverse dalle istanze EC2 nello stesso VPC, puoi configurare le connessioni di rete manualmente. Se utilizzi il AWS Management Console per creare il tuo cluster DB Multi-AZ, puoi fare in modo che Amazon RDS crei automaticamente un VPC per te. Altrimenti, puoi utilizzare un VPC esistente o crearne uno nuovo per il tuo cluster di database Multi-AZ. Il VPC deve avere una o più sottoreti per ognuna delle almeno tre zone di disponibilità per poterlo utilizzare con un cluster di database Multi-AZ. Per informazioni sui VPC, consulta VPC di Amazon VPC e Amazon RDS.

Se non disponi di un VPC predefinito o non hai creato un VPC e non prevedi di utilizzare la console, procedi come segue:

- Crea un VPC con almeno una sottorete in ognuna delle almeno tre zone di disponibilità nella AWS regione in cui desideri implementare il tuo cluster DB. Per ulteriori informazioni, consulta Uso di un'istanza database in un VPC.
- Specificare un gruppo di sicurezza VPC che autorizzi le connessioni al proprio cluster di database . Per ulteriori informazioni, consultare Fornisci accesso alla istanza database nel VPC creando un gruppo di sicurezza e Controllo dell'accesso con i gruppi di sicurezza.
- Specifica di un gruppo di sottoreti del database RDS che definisca almeno due sottoreti nel VPC che possono essere utilizzate dal cluster di database Multi-AZ. Per ulteriori informazioni, consulta Utilizzo di gruppi di sottoreti database.

Per informazioni sui limiti relativi ai cluster di database Multi-AZ, consulta Limitazioni dei cluster DB Multi-AZ.

Se desideri connetterti a una risorsa che non si trova nello stesso VPC del cluster di database multi-AZ, consulta gli scenari appropriati descritti in Scenari per accedere a un'istanza database in un VPC.

Prerequisiti aggiuntivi

Prima di creare il cluster di database Multi-AZ, considera i seguenti prerequisiti aggiuntivi:

- Per connettersi AWS utilizzando le credenziali AWS Identity and Access Management (IAM), il tuo AWS account deve disporre di determinate politiche IAM. che concedono le autorizzazioni

necessarie per eseguire le operazioni Amazon RDS. Per ulteriori informazioni, consulta Gestione accessi e identità per Amazon RDS.

Se utilizzi IAM per accedere alla console RDS, accedi prima AWS Management Console con le tue credenziali utente IAM. Quindi, passa alla console RDS all'indirizzo https://console.aws.amazon.com/rds/.

- Se desideri personalizzare i parametri di configurazione per il cluster di database, specifica un gruppo di parametri cluster di database con le impostazioni dei parametri richieste. Per ulteriori informazioni sulla creazione o la modifica di un gruppo di parametri del cluster di database, consulta Utilizzo di gruppi di parametri per cluster di database Multi-AZ.
- Determina il numero di porta TCP/IP da specificare per il cluster di database. I firewall presso alcune aziende bloccano le connessioni alle porte predefinite. Se il firewall della tua azienda blocca la porta predefinita, scegli un'altra porta per il cluster di database. Tutte le istanze database in un cluster di database utilizzano la stessa porta.
- Se la versione principale del motore per il database ha raggiunto la data di fine del supporto standard RDS, è necessario utilizzare l'opzione Extended Support CLI o il parametro API RDS. Per ulteriori informazioni, vedere RDS Extended Support inImpostazioni per la creazione di cluster di database Multi-AZ.

Creazione di un cluster di database

È possibile creare un cluster DB Multi-AZ utilizzando AWS Management Console, the o l' AWS CLI API RDS.

Console

È possibile creare un cluster di database Multi-AZ scegliendo Multi-AZ DB cluster (Cluster di database Multi-AZ) nella sezione Availability and durability (Disponibilità e durabilità).

Per creare un cluster di database Multi-AZ tramite la console

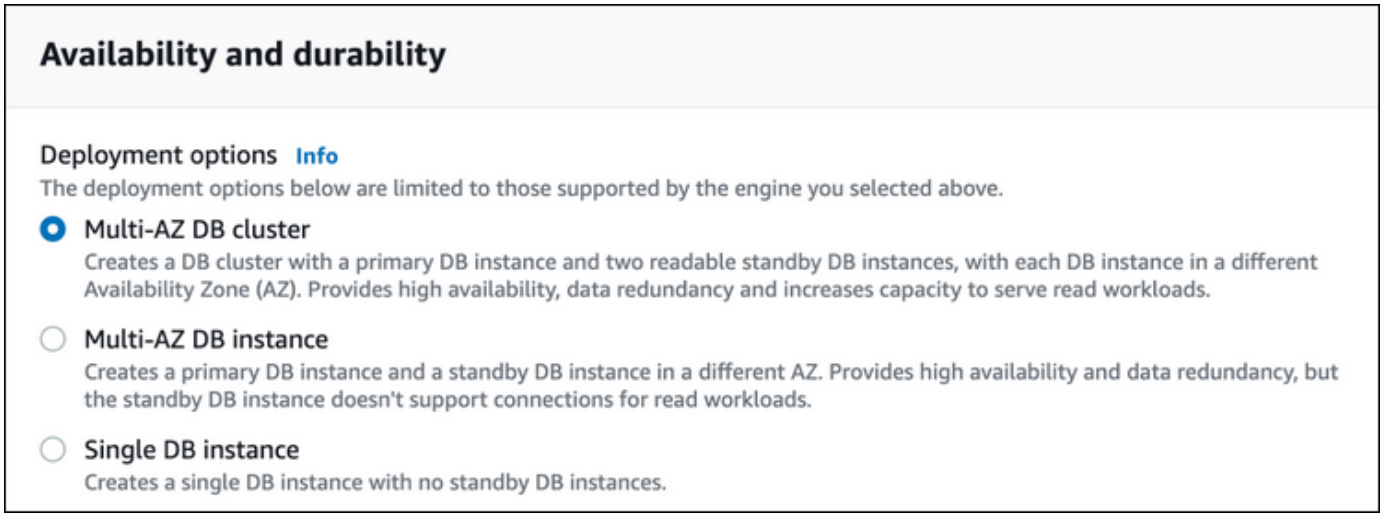
1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nell'angolo in alto a destra di AWS Management Console, scegli il cluster Regione AWS in cui desideri creare il cluster DB.

   Per informazioni sui cluster DB Regioni AWS che supportano i cluster DB Multi-AZ, consulta. Limitazioni dei cluster DB Multi-AZ

3. Nel riquadro di navigazione, scegliere Databases (Database).
4. Scegliere Create database (Crea database).

   Per creare un cluster di database Multi-AZ, assicurati che Standard Create (Creazione standard) sia è selezionato e che Easy Create (Creazione semplice) non lo sia.
5. In Engine type (Tipo di motore), scegli MySQL o PostgreSQL.
6. In Version (Versione), scegliere la versione del motore di database.

   Per informazioni sulle versioni del motore del database che supportano i cluster di database Multi-AZ, consulta Limitazioni dei cluster DB Multi-AZ.
7. In Templates (Modelli), scegli il modello appropriato per l'implementazione.
8. In Availability and durability (Disponibilità e durabilità), scegliere Multi-AZ DB cluster (Cluster di database Multi-AZ).



9. In DB cluster identifier (Identificatore cluster DB), inserisci l'identificatore per il cluster di database.
10. In Master username (Nome utente master), inserisci il tuo nome utente master o mantieni l'impostazione predefinita.
11. Inserire la password master:

    a. Nella sezione Settings (Impostazioni), aprire Credential Settings (Impostazioni credenziali).

    b. Se si desidera specificare una password, deselezionare la casella di spunta Auto generate a password (Genera password automaticamente) se è selezionata.

    c. (Facoltativo) Cambiare il valore Master username (Nome utente master).

d. Inserisci la stessa password in Master password (Password master) e Confirm password (Conferma password).

12. In Classe di istanza database, scegli la classe di istanza database. Per l'elenco delle classi di istanza database supportate, consulta the section called “Disponibilità di classi di istanze per cluster DB Multi-AZ”.
13. (Facoltativo) Configura una connessione a una risorsa di calcolo per questo cluster di database.

    Puoi configurare la connettività tra un'istanza Amazon EC2 e il nuovo cluster di database durante la creazione del cluster di database. Per ulteriori informazioni, consulta Configurazione della connettività di rete automatica con un'istanza EC2.
14. Nella sezione Connettività in Gruppo di sicurezza VPC (firewall), se selezioni Crea nuovo, viene creato un gruppo di sicurezza VPC con una regola in entrata che consente all'indirizzo IP del computer locale di accedere al database.
15. Per le restanti sezioni, specifica le impostazioni del cluster di database. Per informazioni su ciascuna impostazione, consulta Impostazioni per la creazione di cluster di database Multi-AZ.
16. Scegliere Create database (Crea database).

    Se scegli di utilizzare una password generata in modo automatico, il pulsante View credential details (Vedi dettagli delle credenziali) appare nella paginaDatabases.

    Per vedere nome utente e password per il cluster di database, seleziona View credential details (Vedi dettagli delle credenziali).

    Per connetterti al cluster di database come utente principale, utilizza il nome utente e la password visualizzati.

    > **Important**
    >
    > Non potrai visualizzare di nuovo la password dell'utente principale.

17. Per Databases, seleziona il nome del nuovo cluster di database.

Nella console RDS vengono visualizzati i dettagli per il nuovo cluster di database. Lo stato del cluster di database ha lo stato creating (in creazione) fino al completamento della creazione, quando sarà pronto per essere impiegato. Quando lo stato cambia in Available (disponibile), è possibile connettersi al cluster di database. A seconda della classe di cluster di database e dell’archiviazione allocata, potrebbero trascorrere diversi minuti prima che il nuovo cluster di database sia disponibile.

AWS CLI

Prima di creare un cluster DB Multi-AZ utilizzando il AWS CLI, assicurati di soddisfare i prerequisiti richiesti. come la creazione di un VPC e di un gruppo di sottoreti di database RDS. Per ulteriori informazioni, consulta Prerequisiti per i cluster di database.

Per creare un cluster DB Multi-AZ utilizzando AWS CLI, chiamate il comando. create-db-cluster Specificare il valore di `--db-cluster-identifier`. Per l'opzione `--engine`, specificare `mysql` o `postgres`.

Per ulteriori informazioni su ciascuna opzione, consulta Impostazioni per la creazione di cluster di database Multi-AZ.

Per informazioni sui motori DB e sulle Regioni AWS versioni dei motori DB che supportano i cluster DB Multi-AZ, vedere. Limitazioni dei cluster DB Multi-AZ

Il comando `create-db-cluster` crea l'istanza database di scrittore per il cluster di database e due istanze database di lettore. Ogni istanza database si trova in una zona di disponibilità diversa.

Il comando seguente crea ad esempio un nuovo cluster di database Multi-AZ MySQL 8.0 denominato `mysql-multi-az-db-cluster`.

Example

PerLinux, omacOS: Unix

```
aws rds create-db-cluster \
   --db-cluster-identifier mysql-multi-az-db-cluster \
   --engine mysql \
   --engine-version 8.0.28  \
   --master-username admin \
   --manage-master-user-password  \
   --port 3306 \
   --backup-retention-period 1  \
   --db-subnet-group-name default \
   --allocated-storage 4000 \
   --storage-type io1 \
   --iops 10000 \
   --db-cluster-instance-class db.m5d.xlarge
```

Per Windows:

```
aws rds create-db-cluster ^
   --db-cluster-identifier mysql-multi-az-db-cluster ^
   --engine mysql ^
   --engine-version 8.0.28 ^
   --manage-master-user-password ^
   --master-username admin ^
   --port 3306 ^
   --backup-retention-period 1 ^
   --db-subnet-group-name default ^
   --allocated-storage 4000 ^
   --storage-type io1 ^
   --iops 10000 ^
   --db-cluster-instance-class db.m5d.xlarge
```

Il comando seguente crea un nuovo cluster di database Multi-AZ PostgreSQL 13.4 denominato `postgresql-multi-az-db-cluster`.

Example

Per LinuxmacOS, oUnix:

```
aws rds create-db-cluster \
   --db-cluster-identifier postgresql-multi-az-db-cluster \
   --engine postgres \
   --engine-version 13.4 \
   --manage-master-user-password \
   --master-username postgres \
   --port 5432 \
   --backup-retention-period 1  \
   --db-subnet-group-name default \
   --allocated-storage 4000 \
   --storage-type io1 \
   --iops 10000 \
   --db-cluster-instance-class db.m5d.xlarge
```

Per Windows:

```
aws rds create-db-cluster ^
   --db-cluster-identifier postgresql-multi-az-db-cluster ^
   --engine postgres ^
   --engine-version 13.4 ^
   --manage-master-user-password ^
```

```
    --master-username postgres ^
    --port 5432 ^
    --backup-retention-period 1 ^
    --db-subnet-group-name default ^
    --allocated-storage 4000 ^
    --storage-type io1 ^
    --iops 10000 ^
    --db-cluster-instance-class db.m5d.xlarge
```

API RDS

Per poter creare un cluster di database Multi-AZ tramite API RDS, assicurarsi che vengano soddisfatti i prerequisiti necessari, come la creazione di un VPC e di un gruppo di sottoreti del database RDS. Per ulteriori informazioni, consulta Prerequisiti per i cluster di database.

Per creare un cluster di database Multi-AZ con l'API RDS, esegui l'operazione CreateDBCluster. Specificare il valore di `DBClusterIdentifier`. Per il parametro `Engine`, specificare tra `mysql` o `postgresql`.

Per ulteriori informazioni su ciascuna opzione, consulta Impostazioni per la creazione di cluster di database Multi-AZ.

L'operazione `CreateDBCluster` crea l'istanza database di scrittore per il cluster di database e due istanze database di lettore. Ogni istanza database si trova in una zona di disponibilità diversa.

Impostazioni per la creazione di cluster di database Multi-AZ

Per i dettagli sulle impostazioni disponibili quando crei un cluster di database Multi-AZ, consulta la tabella seguente. Per ulteriori informazioni sulle AWS CLI opzioni, vedere create-db-cluster. Per ulteriori informazioni sui parametri API RDS, consulta CreateComputeEnvironment.

Impostazione della console	Descrizione impostazione	Opzione CLI e parametro API di RDS
Allocated storage (Storage allocato)	La quantità di archiviazione, in gibibyte, da allocare per ciascuna istanza database nel cluster di database. Per ulteriori informazioni, consulta Storage delle istanze di database Amazon RDS.	Opzione CLI: `--allocated-storage` Parametro API: `AllocatedStorage`

Impostazione della console	Descrizione impostazione	Opzione CLI e parametro API di RDS
Auto minor version upgrade (Aggiornamento automatico della versione secondaria)	Abilita l'aggiornamento automatico della versione secondaria per far sì che il cluster di database riceva automaticamente gli aggiornamenti della versione del motore di database secondaria preferita quando diventano disponibili. Amazon RDS esegue aggiornamenti automatici di versioni secondarie nella finestra di manutenzione.	Opzione CLI: `--auto-minor-version-upgrade` `--no-auto-minor-version-upgrade` Parametro API: `AutoMinorVersionUpgrade`
Backup retention period (Periodo di retention dei backup)	Il numero di giorni in cui desideri eseguire il backup automatico del cluster di database da mantenere. Per un cluster di database Multi-AZ, questo valore deve essere impostato sul valore uguale o maggiore di **1**. Per ulteriori informazioni, consulta Introduzione ai backup.	Opzione CLI: `--backup-retention-period` Parametro API: `BackupRetentionPeriod`
Backup window (Finestra di backup)	Il periodo di tempo durante il quale Amazon RDS esegue automaticamente un backup del cluster di database. A meno che non si abbiano preferenze specifiche per l'ora di esecuzione del backup del database, usare il valore predefinito No Preference (Nessuna preferenza). Per ulteriori informazioni, consulta Introduzione ai backup.	Opzione CLI: `--preferred-backup-window` Parametro API: `PreferredBackupWindow`

Impostazione della console	Descrizione impostazione	Opzione CLI e parametro API di RDS
Autorità di certificazione	L'autorità di certificazione (CA) per il certificato del server utilizzato dal cluster DB. Per ulteriori informazioni, consulta .	Opzione CLI: `--ca-certificate-identifier` Parametro API RDS: `CACertificateIdentifier`
Copy tags to snapshots (Copia tag in snapshot)	Questa opzione consente di copiare i tag del cluster di database in uno snapshot DB quando si crea uno snapshot. Per ulteriori informazioni, consulta Tagging delle risorse Amazon RDS.	Opzione CLI: `-copy-tags-to-snapshot` `-no-copy-tags-to-snapshot` Parametro API RDS: `CopyTagsToSnapshot`
Database authentication (Autenticazione del database)	Password authentication (Autenticazione password) è supportato solo cluster di database Multi-AZ.	Nessuna perché l'autenticazione con password è predefinita.

Impostazione della console	Descrizione impostazione	Opzione CLI e parametro API di RDS
Database port (Porta del database)	La porta che si desidera utilizzare per al cluster di database. La porta predefinita è visualizzata. La porta non può essere modificat a dopo la creazione del cluster di database. I firewall presso alcune aziende bloccano le connessioni alle porte predefinite. Se il firewall della tua azienda blocca la porta predefinita, entra da un'altra porta per il cluster di database.	Opzione CLI: `--port` Parametro API RDS: `Port`
DB Cluster Identifier (Identificatore cluster DB)	Nome per il cluster di database. Assegna un nome ai cluster di database nello stesso modo in cui assegni un nome ai server in locale. L'identificatore del cluster DB può contenere fino a 63 caratteri alfanumerici e deve essere univoco per il tuo account nella AWS regione che hai scelto.	Opzione CLI: `--db-cluster-identifier` Parametro API RDS: `DBClusterIdentifier`

Impostazione della console	Descrizione impostazione	Opzione CLI e parametro API di RDS
DB instance class (Classe istanza database)	La capacità di calcolo e di memoria di ciascuna istanza database nel cluster di Multi-AZ, ad esempio `db.m5d.xlarge` . Se possibile, scegliere una classe di istanza database sufficien temente ampia da poter tenere in memoria un tipico set di lavoro di query. Quando i set di lavoro sono conservati in memoria, il sistema può evitare di scrivere sul disco, migliorando le prestazioni. Per l'elenco delle classi di istanza database supportate, consulta the section called “Disponibilità di classi di istanze per cluster DB Multi-AZ”.	Opzione CLI: `--db-cluster-instance-class` Parametro API RDS: `DBClusterInstanceClass`
DB cluster parameter group (Gruppo di parametri del cluster database)	Gruppo di parametri del cluster di database da associare al cluster di database. Per ulteriori informazioni, consulta Utilizzo di gruppi di parametri per cluster di database Multi-AZ.	Opzione CLI: `--db-cluster-parameter-group-name` Parametro API RDS: `DBClusterParameterGroupName`
DB engine version (Versione motore del database	Versione del motore del database da utilizzare.	Opzione CLI: `--engine-version` Parametro API RDS: `EngineVersion`

Impostazione della console	Descrizione impostazione	Opzione CLI e parametro API di RDS
DB cluster parameter group (Gruppo di parametri del cluster database)	Il gruppo di parametri dell'istanza DB da associare al cluster DB. Per ulteriori informazioni, consulta Utilizzo di gruppi di parametri per cluster di database Multi-AZ.	Opzione CLI: `--db-cluster-parameter-group-name` Parametro API RDS: **`DBClusterParameterGroupName`**
DB subnet group (Gruppo di sottoreti DB)	Il gruppo di sottoreti database da utilizzare per il cluster di database. Seleziona Choose existing (Scegli esistente) per utilizzare un gruppo di sottoreti database esistente. Quindi scegli il gruppo di sottoreti richiesto dall'elenco a discesa Existing DB subnet groups (Gruppi di sottoreti DB esistenti). Scegli Automatic setup (Configurazione automatica) per consentire a RDS di selezionare un gruppo di sottoreti database compatibile. Se non ne esiste uno, RDS crea un nuovo gruppo di sottoreti per il cluster. Per ulteriori informazioni, consulta Utilizzo di gruppi di sottoreti database.	Opzione CLI: `--db-subnet-group-name` Parametro API RDS: `DBSubnetGroupName`

Impostazione della console	Descrizione impostazione	Opzione CLI e parametro API di RDS
Deletion protection (Protezione da eliminazione)	L'opzione Enable deletion protection (Abilita protezione da eliminazione) permette di impedire l'eliminazione del cluster di database. Se si crea un cluster di database di produzione con la console, la protezione da eliminazione è accesa per impostazione predefinita. Per ulteriori informazioni, consulta Eliminazione di un'istanza database.	Opzione CLI: `--deletion-protection` `--no-deletion-protection` Parametro API RDS: `DeletionProtection`
Encryption (Crittografia)	Enable encryption (Abilita crittografia) per abilitare la crittografia al resto di tale cluster di database. La crittografia è attivata per impostazione predefinita per i cluster di database Multi-AZ. Per ulteriori informazioni, consulta Crittografia delle risorse Amazon RDS.	Opzioni CLI: `--kms-key-id` `--storage-encrypted` `--no-storage-encrypted` Parametri API RDS: `KmsKeyId` `StorageEncrypted`

Impostazione della console	Descrizione impostazione	Opzione CLI e parametro API di RDS
Enhanced Monitoring	Scegliere Enable enhanced monitoring (Abilita monitoraggio avanzato) per abilitare la raccolta di parametri in tempo reale per il sistema operativo su cui viene eseguito il cluster di database. Per ulteriori informazioni, consulta Monitoraggio dei parametri del sistema operativo con il monitoraggio avanzato.	Opzioni CLI: `--monitoring-interval` `--monitoring-role-arn` Parametri API RDS: `MonitoringInterval` `MonitoringRoleArn`

Impostazione della console	Descrizione impostazione	Opzione CLI e parametro API di RDS
Initial database name (Nome del database iniziale)	Il nome per il database nel cluster di database. Se non fornisci un nome, Amazon RDS non crea un database nel cluster di database per MySQL. Tuttavia, crea un database nel cluster di database per PostgreSQL. Il nome non può essere una parola riservata del motore di database. Prevede altri vincoli a seconda del motore di database. MySQL: • Deve contenere da 1 a 64 caratteri alfanumerici. PostgreSQL: • Deve contenere da 1 a 63 caratteri alfanumerici. • Deve iniziare con una lettera o un trattino basso. I caratteri successivi possono essere lettere, trattini bassi o cifre (0-9). • Il nome del database iniziale è `postgres`.	Opzione CLI: `--database-name` Parametro API RDS: `DatabaseName`

Impostazione della console	Descrizione impostazione	Opzione CLI e parametro API di RDS
Log exports (Esportazioni log)	I tipi di file di log del database da pubblicare su Amazon CloudWatch Logs. Per ulteriori informazioni, consulta Pubblicazione di log di database su Amazon CloudWatch Logs.	Opzione CLI: `-enable-cloudwatch-logs-exports` Parametro API RDS: `EnableCloudwatchLogsExports`
Maintenance window (Finestra di manutenzione)	La finestra di 30 minuti entro cui vengono applicate la modifiche in corso al cluster di database. Se il periodo di tempo non è rilevante, scegli No Preference (Nessuna preferenza). Per ulteriori informazioni, consulta Finestra di manutenzione Amazon RDS.	Opzione CLI: `--preferred-maintenance-window` Parametro API RDS: `PreferredMaintenanceWindow`

Impostazione della console	Descrizione impostazione	Opzione CLI e parametro API di RDS
Gestisci le credenziali principali in AWS Secrets Manager	Seleziona Manage master credentials in AWS Secrets Manager (Gestione credenziali master in AWS Secrets Manager) per gestire la password dell'utente master in un segreto di Secrets Manager. Facoltativamente, scegli la chiave KMS da utilizzare per proteggere il segreto. Scegliere tra le chiavi KMS presenti nell'account o inserire la chiave da un altro account. Per ulteriori informazioni, consulta Gestione delle password con Amazon RDS e AWS Secrets Manager.	Opzione CLI: `--manage-master-user-password | --no-manage-master-user-password` `--master-user-secret-kms-key-id` Parametro API RDS: `ManageMasterUserPassword` `MasterUserSecretKmsKeyId`
Master password (Password master)	La password dell'account utente master.	Opzione CLI: `--master-user-password` Parametro API RDS: `MasterUserPassword`

Impostazione della console	Descrizione impostazione	Opzione CLI e parametro API di RDS
Master username (Nome utente master)	Nome da utilizzare come nome utente master per accedere al cluster di database con tutti i privilegi del database. • Può contenere da 1 a 16 caratteri alfanumerici e caratteri di sottolineatura. • Il primo carattere deve essere una lettera. • Non può essere una parola riservata del motore del database. Dopo la creazione del cluster di database Multi-AZ, non è possibile modificare il nome utente master. Per ulteriori informazioni sui privilegi concessi all'utente master, consultare Privilegi dell'account utente master.	Opzione CLI: `--master-username` Parametro API RDS: `MasterUsername`

Impostazione della console	Descrizione impostazione	Opzione CLI e parametro API di RDS
Approfondimenti sulle prestazioni	Enable Approfondimenti sulle prestazioni (Abilita Approfondimenti sulle prestazioni) per monitorare il carico dei cluster di database e consentire l'analisi e la risoluzione dei problemi di prestazioni del database. Seleziona un periodo di mantenimento per stabilire quanta cronologia di dati di Approfondimenti sulle prestazioni conservare. L'impostazione del periodo di conservazione nel livello gratuito è Default (7 days) (Impostazione predefinita (7 giorni)). Per mantenere i dati sulle prestazioni più a lungo, specifica da 1 a 24 mesi. Per altre informazioni sui periodi di conservazione, consulta Prezzi e conservazione dei dati per Performance Insights. Scegliere la chiave master da utilizzare per proteggere la chiave usata per crittografare questo volume di database. Scegliere tra le chiavi master presenti nell'account o inserire la chiave da un altro account. Per ulteriori informazioni, consulta Monitoraggio del carico DB con Performance Insights su Amazon RDS.	Opzioni CLI: `--enable-performance-insights` `--no-enable-performance-insights` `--performance-insights-retention-period` `--performance-insights-kms-key-id` Parametri API RDS: `EnablePerformanceInsights` `PerformanceInsightsRetentionPeriod` `PerformanceInsightsKMSKeyId`

Impostazione della console	Descrizione impostazione	Opzione CLI e parametro API di RDS
IOPS con provisioning	Quantità di IOPS con provisioning (operazioni di input/output al secondo) da allocare inizialmente al cluster di database.	Opzione CLI: `--iops` Parametro API RDS: `Iops`

Impostazione della console	Descrizione impostazione	Opzione CLI e parametro API di RDS
Accesso pubblico	Publicly accessible (Accessibile pubblicamente) per assegnare al cluster di database un indirizzo IP pubblico, ovvero renderla accessibile al di fuori del VPC. Per essere accessibile pubblicamente, il cluster di database deve anche trovarsi in una sottorete pubblica nel VPC. Not publicly accessible (Non accessibile pubblicamente) per rendere il cluster di database accessibile solo dall'interno del VPC. Per ulteriori informazioni, consulta Nascondere istanze database in un VPC da Internet. Per connettersi a un cluster di database dall'esterno del proprio VPC, il cluster di database deve essere accessibile pubblicamente. Inoltre, l'accesso deve essere concesso utilizzando le regole in ingresso del gruppo di sicurezza del cluster di database e devono essere soddisfatti altri requisiti. Per ulteriori informazioni, consulta Impossibile connettersi all'istanza database di Amazon RDS. Se il tuo cluster DB non è accessibile pubblicamente, puoi utilizzare una connessione AWS VPN da	Opzione CLI: `--publicly-accessible` `--no-publicly-accessible` Parametro API RDS: `PubliclyAccessible`

Impostazione della console	Descrizione impostazione	Opzione CLI e parametro API di RDS
	sito a sito o AWS Direct Connect una connessione per accedervi da una rete privata. Per ulteriori informazioni, consulta Riservatezza del traffico Internet.	
Supporto esteso RDS	Seleziona Enable RDS Extended Support per consentire alle principali versioni del motore supportate di continuare a funzionare oltre la data di fine del supporto standard RDS. Quando crei un cluster DB, Amazon RDS utilizza per impostazione predefinita RDS Extended Support. Per impedire la creazione di un nuovo cluster DB dopo la data di fine del supporto standard RDS e per evitare addebiti per RDS Extended Support, disabilita questa impostazione. I cluster DB esistenti non verranno addebitati fino alla data di inizio dei prezzi di RDS Extended Support. Per ulteriori informazioni, consulta Utilizzo dell'estensione del supporto per Amazon RDS.	Opzione CLI: `--engine-lifecycle-support` Parametro API RDS: `EngineLifecycleSupport`

Impostazione della console	Descrizione impostazione	Opzione CLI e parametro API di RDS
Velocità di trasmissione effettiva per archiviazione	Il valore del throughput di archiviazione per il cluster DB. Questa impostazione è visibile solo se si sceglie General Purpose SSD (gp3) per il tipo di archiviazione. Questa impostazione non è configurabile e viene impostata automaticamente in base agli IOPS specificati. Per ulteriori informazioni, consulta archiviazione gp3 (consigliata).	Questo valore viene calcolato automaticamente e non dispone di un'opzione CLI.
Server proxy per RDS	Scegli Create an RDS Proxy (Crea un server proxy per RDS) per creare un proxy per il tuo cluster database. Amazon RDS crea automaticamente per il proxy un ruolo IAM e un segreto in Secrets Manager.	Non disponibile durante la creazione di un cluster database.
Storage Type (Tipo di storage)	Il tipo di archiviazione per il cluster di database. Sono supportati solo gli storage General Purpose SSD (gp3), Provisioned IOPS (io1) e Provisioned IOPS SSD (io2). Per ulteriori informazioni, consulta Tipi di storage Amazon RDS.	Opzione CLI: `--storage-type` Parametro API RDS: `StorageType`

Impostazione della console	Descrizione impostazione	Opzione CLI e parametro API di RDS
Virtual Private Cloud (VPC)	Un VPC basato sul servizio Amazon VPC da associare al cluster di database. Per ulteriori informazioni, consulta VPC di Amazon VPC e Amazon RDS.	Per CLI e API, specificare gli ID del gruppo di sicurezza VPC.
VPC security group (firewall) (Gruppo di sicurezza VPC (firewall)	I gruppi di sicurezza da associare al cluster di database. Per ulteriori informazioni, consulta Panoramica dei gruppi di sicurezza VPC.	Opzione CLI: `--vpc-security-group-ids` Parametro API RDS: `VpcSecurityGroupIds`

Impostazioni che non si applicano durante la creazione di cluster DB Multi-AZ

Le seguenti impostazioni nel AWS CLI comando `create-db-cluster`e nell'operazione dell'API RDS non si applicano ai cluster DB Multi-AZ. `CreateDBCluster`

Inoltre, non è possibile specificare queste impostazioni per i cluster di database Multi-AZ nella console.

AWS CLI impostazione	Impostazione API RDS
`--availability-zones`	`AvailabilityZones`
`--backtrack-window`	`BacktrackWindow`
`--character-set-name`	`CharacterSetName`
`--domain`	`Domain`
`--domain-iam-role-name`	`DomainIAMRoleName`

AWS CLI impostazione	Impostazione API RDS
`--enable-global-write-forwarding | --no-enable-global-write-forwarding`	`EnableGlobalWriteForwarding`
`--enable-http-endpoint | --no-enable-http-endpoint`	`EnableHttpEndpoint`
`--enable-iam-database-authentication | --no-enable-iam-database-authentication`	`EnableIAMDatabaseAuthentication`
`--global-cluster-identifier`	`GlobalClusterIdentifier`
`--option-group-name`	`OptionGroupName`
`--pre-signed-url`	`PreSignedUrl`
`--replication-source-identifier`	`ReplicationSourceIdentifier`
`--scaling-configuration`	`ScalingConfiguration`

Connessione a un cluster di database multi-AZ

Un cluster di database Multi-AZ dispone di tre istanze database anziché di una singola istanza database. Ogni connessione viene gestita da un'istanza database specifica. Quando ti connetti a un cluster di database multi-AZ, il nome host e la porta specificati puntano a un nome di dominio completo chiamato endpoint. Il cluster di database multi-AZ utilizza il meccanismo di endpoint per astrarre queste connessioni in modo che non sia necessario specificare esattamente l'istanza database del cluster di database multi-AZ. Pertanto, non è necessario codificare tutti i nomi host o scrivere una propria logica per il reindirizzamento delle connessioni quando alcune istanze database non sono disponibili.

L'endpoint di scrittura si connette all'istanza database di scrittura del cluster di database, che supporta operazioni di lettura e scrittura. L'endpoint di lettura si collega a una delle due istanze database di lettura, che supportano solo le operazioni di lettura.

Usando gli endpoint puoi associare ogni connessione all'istanza database o al gruppo di istanze database appropriato in base al caso d'uso. Ad esempio, per eseguire le istruzioni DDL e DML puoi connetterti a qualsiasi istanza database sia l'istanza database di scrittura. Per eseguire le query, puoi connetterti all'endpoint di lettura, mentre il cluster di database Multi-AZ gestisce automaticamente le connessioni tra le istanze database di lettura. Per la diagnosi o l'ottimizzazione, puoi connetterti a un endpoint di istanza database specifico per esaminare i dettagli su una determinata istanza database.

Per informazioni sulla connessione a un'istanza database, consulta Connessione a un'istanza database Amazon RDS.

Argomenti

Tipi di endpoint cluster di database Multi-AZ

Un endpoint è rappresentato da un identificatore univoco contenente un indirizzo host. Di seguito sono riportati i tipi di endpoint disponibili da un cluster di database Multi-AZ:

Endpoint del cluster

Per endpoint del cluster (o endpoint di scrittura) si intende un endpoint per un cluster di datababse Multi-AZ che si connette all'istanza database di scrittura corrente di quel cluster di database. Questo endpoint è l'unico in grado di eseguire operazioni di scrittura come le istruzioni DDL e DML. Questo endpoint può anche eseguire operazioni di lettura.

Ciascun cluster di database Multi-AZ ha un endpoint del cluster e un'istanza database di scrittura.

L'endpoint del cluster si usa per tutte le operazioni di scrittura sul cluster database, inclusi aggiornamenti, inserimenti, eliminazioni e modifiche DDL. Puoi anche utilizzare l'endpoint del cluster per le operazioni di lettura, come ad esempio le query.

In caso di errore dell'istanza database di scrittura corrente di un cluster database, il cluster di database Multi-AZ esegue automaticamente il failover su una nuova istanza database di scrittura. Durante un failover, il cluster database continua a servire le richieste di connessione all'endpoint del cluster dalla nuova istanza database di scrittura, riducendo al minimo l'interruzione del servizio.

L'esempio seguente mostra un endpoint del cluster per un cluster di database Multi-AZ.

```
mydbcluster.cluster-123456789012.us-east-1.rds.amazonaws.com
```

Endpoint di lettura

Un endpoint di lettura per un cluster di database Multi-AZ fornisce supporto per le connessioni di sola lettura al cluster di database. Puoi utilizzare l'endpoint di lettura per le operazioni di lettura, come ad esempio le query `SELECT`. Elaborando tali istruzioni nelle istanze database del lettore, questo endpoint riduce il sovraccarico sull'istanza database di scrittore. Consente inoltre al cluster di dimensionare la capacità di gestire simultaneamente query `SELECT`. Ogni cluster database Multi-AZ ha un endpoint di lettura.

L'endpoint di lettura invia ogni richiesta di connessione a una delle istanze database di lettura. Quando utilizzi l'endpoint di lettura per una sessione, è possibile eseguire solo istruzioni di sola lettura come `SELECT` in quella sessione.

L'esempio seguente mostra un endpoint di lettura per un cluster di database Multi-AZ. La modalità di sola lettura di un endpoint di lettura è indicato dal parametro `-ro` all'interno del nome dell'endpoint del cluster.

```
mydbcluster.cluster-ro-123456789012.us-east-1.rds.amazonaws.com
```

Endpoint dell'istanza

Un endpoint dell'istanza si connette a un'istanza database specifica all'interno di un cluster di database Multi-AZ. Ogni istanza database in un cluster database dispone del proprio endpoint dell'istanza univoco. Pertanto esiste un endpoint dell'istanza per l'istanza database di scrittura corrente del cluster di database e un endpoint dell'istanza per ciascuna istanza database di lettore nel cluster di database.

L'endpoint dell'istanza fornisce controllo diretto sulle connessioni al cluster di database. Questo controllo può aiutarti a risolvere scenari in cui l'utilizzo dell'endpoint del cluster o dell'endpoint di lettura potrebbe non essere appropriato. Ad esempio, l'applicazione client potrebbe richiedere un maggiore bilanciamento del carico granulare in base al tipo di carico di lavoro. In questo caso, è possibile configurare più client per connettersi a istanze database di lettore in un cluster database per distribuire i carichi di lavoro in lettura.

L'esempio seguente mostra un endpoint dell'istanza per un'istanza database in un cluster di database Multi-AZ.

```
mydbinstance.123456789012.us-east-1.rds.amazonaws.com
```

Visualizzazione degli endpoint per un cluster di database Multi-AZ

Nella AWS Management Console, puoi vedere l'endpoint del cluster e l'endpoint del lettore nella pagina dei dettagli di ogni cluster DB Multi-AZ. L'endpoint dell'istanza viene visualizzato nella pagina dei dettagli di ogni database.

Con AWS CLI, puoi vedere gli endpoint writer e reader nell'output del comando. describe-db-clusters Ad esempio, il comando seguente mostra gli attributi degli endpoint per tutti i cluster nella regione corrente. AWS

```
aws rds describe-db-cluster-endpoints
```

Con l'API Amazon RDS, recuperi gli endpoint chiamando l'azione DescribeDB. ClusterEndpoints L'output mostra anche gli endpoint del cluster di database Amazon Aurora DB, se presenti.

Utilizzo dell'endpoint del cluster

Ogni cluster di database Multi-AZ ha un singolo endpoint cluster integrato, il cui nome e altri attributi sono gestiti da Amazon RDS. Non puoi creare, eliminare o modificare questo tipo di endpoint.

L'endpoint del cluster viene utilizzato per la gestione del cluster di database, l'esecuzione di operazioni di estrazione, trasformazione, caricamento (ETL) o di applicazioni di sviluppo e test. L'endpoint del cluster si connette all'istanza di scrittura del cluster di database. L'istanza database di scrittura è l'unica istanza database in cui è possibile creare tabelle e indici, eseguire istruzioni `INSERT` e altre operazioni DDL e DML.

L'indirizzo IP fisico indicato dall'endpoint del cluster cambia quando il meccanismo di failover promuove una nuova istanza database di scrittura come istanza primaria di lettura-scrittura per il cluster. Se utilizzi qualsiasi forma di pool di connessioni o altro tipo di multiplexing, preparati a eliminare o ridurre le informazioni DNS memorizzate nella cache. time-to-live In tal modo si evita di provare a stabilire una connessione in lettura-scrittura a un'istanza database che non è più disponibile o che ora è di sola lettura a causa di un failover.

Utilizzo dell'endpoint di lettura

Usi l'endpoint di lettura fornisce per le connessioni di sola lettura al cluster di database Multi-AZ. Questo endpoint aiuta il cluster di database a gestire un carico di lavoro che implica numerose query. L'endpoint di lettura è quello che fornisci alle applicazioni che eseguono report o altre operazioni di sola lettura sul cluster. L'endpoint di lettura invia le connessioni solo alle istanze database di lettura in un cluster di database Multi-AZ.

Ogni cluster Multi-AZ ha un singolo endpoint di lettura integrato, il cui nome e altri attributi sono gestiti da Amazon RDS. Non puoi creare, eliminare o modificare questo tipo di endpoint.

Utilizzo degli endpoint di istanza

Ogni istanza database in un cluster di database Multi-AZ ha un proprio endpoint di istanza integrato, il cui nome e altri attributi sono gestiti da Amazon RDS. Non puoi creare, eliminare o modificare questo tipo di endpoint. In genere con un cluster di database Multi-AZ si utilizzano gli endpoint di scrittura e lettura più spesso degli endpoint di istanza.

Nelle day-to-day operazioni, il modo principale di utilizzare gli endpoint delle istanze è diagnosticare problemi di capacità o prestazioni che riguardano una specifica istanza DB in un cluster DB Multi-AZ. Durante la connessione a un'istanza database specifica, puoi esaminare le variabili di stato, i

parametri e così via. Ciò può aiutarti a determinare cosa sta succedendo di diverso per quell'istanza database da ciò che accade per le altre istanze database nel cluster.

Come gli endpoint database Multi-AZ funzionano con elevata disponibilità

Per i cluster database Multi-AZ in cui è importante la disponibilità elevata, utilizza l'endpoint di scrittura per le connessioni di lettura-scrittura o a scopo generale e l'endpoint di lettura per le connessioni di sola lettura. Gli endpoint di scrittura e lettura gestiscono il failover delle istanze DB meglio degli endpoint di istanza. A differenza degli endpoint istanza, gli endpoint di scrittura e lettura modificano automaticamente l'istanza database a cui si connettono se un'istanza database nel cluster diventa non disponibile.

In caso di errore dell'istanza database di scrittura di un cluster database, Amazon RDS esegue automaticamente il failover su una nuova istanza database di scrittura. Lo fa promuovendo un'istanza database di lettore in una nuova istanza database di scrittore. Se si verifica un failover, è possibile utilizzare l'endpoint di scrittura per riconnettersi all'istanza database di scrittura appena promossa. Oppure è possibile utilizzare l'endpoint di lettura per riconnettersi a una delle istanze database di lettore nel cluster di database. Durante un failover, l'endpoint di lettura potrebbe dirigere le connessioni alla nuova istanza database di scrittura di un cluster di database per un breve periodo di tempo dopo che un'istanza database di lettura viene promossa a nuova istanza database di scrittura. Se progetti la tua logica applicativa per gestire le connessioni agli endpoint di istanza, puoi rilevare a livello di codice o manualmente il set risultante di istanze database disponibili nel cluster database.

Connessione ai cluster DB Multi-AZ con i driver AWS

La AWS suite di driver è stata progettata per fornire supporto per tempi di switchover e failover più rapidi e l'autenticazione con AWS Secrets Manager, AWS Identity and Access Management (IAM) e Federated Identity. I AWS driver si basano sul monitoraggio dello stato del cluster DB e sulla conoscenza della topologia del cluster per determinare il nuovo writer. Questo approccio riduce i tempi di switchover e failover a secondi a una cifra, rispetto alle decine di secondi dei driver open source.

Con l'introduzione di nuove funzionalità di servizio, l'obiettivo della AWS suite di driver è disporre di un supporto integrato per queste funzionalità di servizio.

Connessione a cluster DB Multi-AZ con il driver JDBC Amazon Web Services (AWS)

Il driver JDBC di Amazon Web Services (AWS) è progettato come wrapper JDBC avanzato per aiutare le applicazioni a sfruttare le funzionalità dei database in cluster. Questo wrapper

è complementare e amplia le funzionalità di un driver JDBC esistente. Il driver è compatibile direttamente con i seguenti driver della community:

- Connettore MySQL/J
- MariaDB Connector/J
- PgJDBC

Per installare il driver AWS JDBC, aggiungi il file.jar del driver AWS JDBC (che si trova nell'applicazione`CLASSPATH`) e mantieni i riferimenti al rispettivo driver della community. Aggiorna il rispettivo prefisso dell'URL di connessione come segue:

- `jdbc:mysql://` Da a `jdbc:aws-wrapper:mysql://`
- `jdbc:mariadb://` Da a `jdbc:aws-wrapper:mariadb://`
- `jdbc:postgresql://` Da a `jdbc:aws-wrapper:postgresql://`

Per ulteriori informazioni sul driver AWS JDBC e istruzioni complete per il suo utilizzo, consulta l'archivio dei driver JDBC di Amazon Web Services (AWS). GitHub

Connessione a cluster DB Multi-AZ con il driver AWS Python di Amazon Web Services ()

Il driver Python di Amazon Web Services (AWS) è progettato come wrapper Python avanzato. Questo wrapper è complementare ed estende le funzionalità del driver open source Psycopg. Il AWS Python Driver supporta le versioni Python 3.8 e successive. È possibile installare il `aws-advanced-python-wrapper` pacchetto utilizzando il `pip` comando, insieme ai pacchetti open source. `psycopg`

Per ulteriori informazioni sul driver AWS Python e istruzioni complete per il suo utilizzo, consulta il repository Amazon Web Services ()AWS Python Driver. GitHub

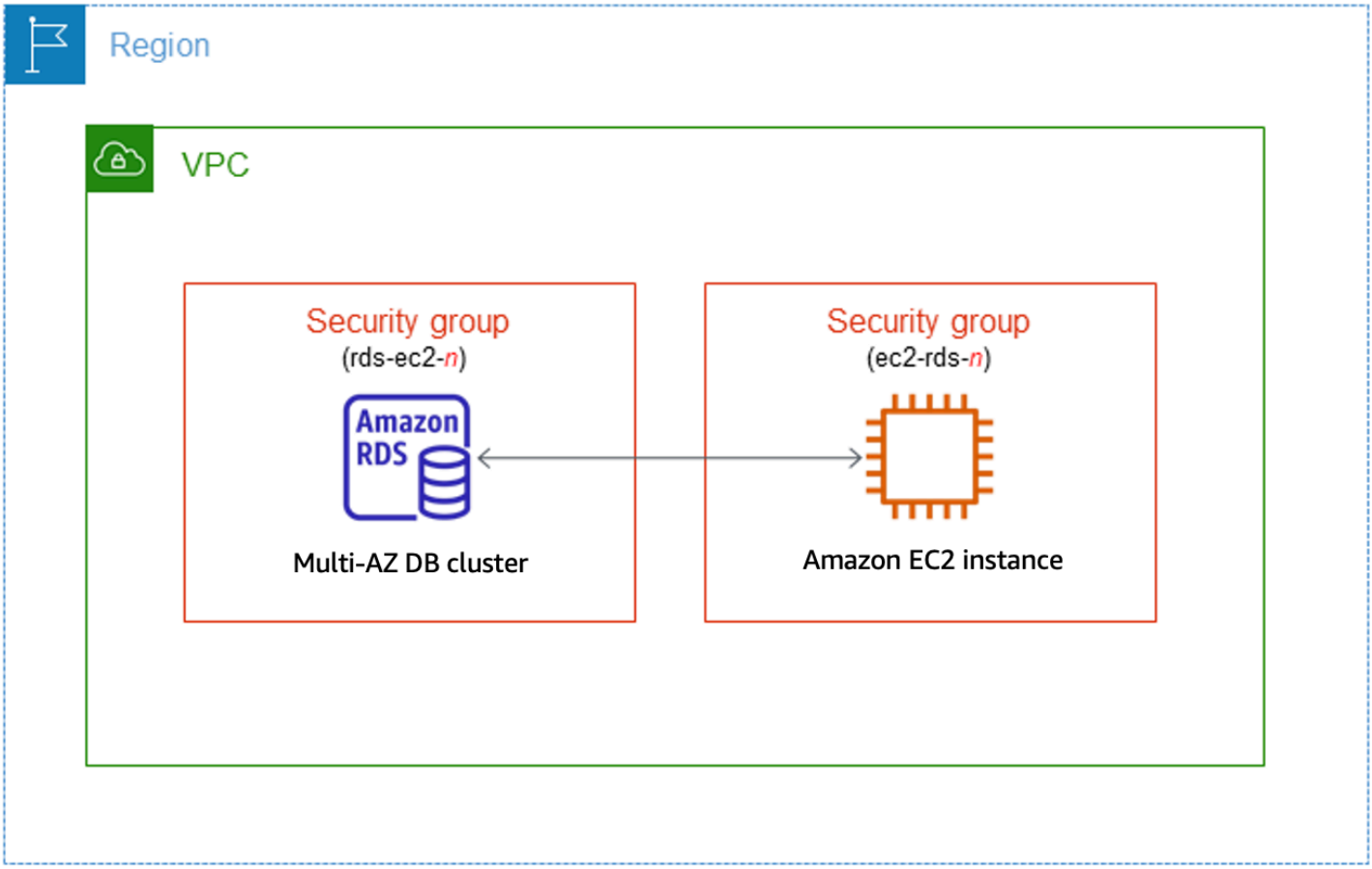
Connessione automatica di una risorsa di calcolo AWS e di un cluster database Multi-AZ

Puoi connettere automaticamente a un cluster database Multi-AZ e risorse di calcolo AWS quali istanze di Amazon Elastic Compute Cloud (Amazon EC2) e funzioni AWS Lambda.

Argomenti

Connessione automatica di un'istanza EC2 e un cluster database Multi-AZ

È possibile usare la console Amazon RDS per semplificare l'impostazione di una connessione tra un'istanza Amazon Elastic Compute Cloud (Amazon EC2) e un cluster database Multi-AZ. Spesso, il cluster database Multi-AZ si trova in una sottorete privata e l'istanza EC2 si trova in una sottorete pubblica all'interno di un cloud privato virtuale (VPC). È possibile utilizzare un client SQL sull'istanza EC2 per connettersi al cluster database Multi-AZ. L'istanza EC2 può anche eseguire server o applicazioni web che accedono al cluster database Multi-AZ privato.



Se desideri connetterti a un'istanza EC2 che non si trova nello stesso VPC del cluster di database multi-AZ, consulta gli scenari descritti in the section called “Scenari per accedere a un'istanza database in un VPC”.

Argomenti

- Panoramica della connettività automatica con un'istanza EC2
- Connessione automatica di un'istanza EC2 e un cluster di database multi-AZ
- Visualizzazione delle risorse di calcolo connesse

Panoramica della connettività automatica con un'istanza EC2

Quando si imposta una connessione tra un'istanza EC2 e un cluster database Multi-AZ, Amazon RDS configura il gruppo di sicurezza VPC per l'istanza EC2 e per il cluster database.

Di seguito sono riportati i requisiti per connettere un'istanza EC2 a un cluster di database multi-AZ:

- L'istanza EC2 deve risiedere nello stesso VPC del cluster di database multi-AZ.

Se nello stesso VPC non esistono istanze EC2, la console fornisce un collegamento per crearne una.

- L'utente che sta configurando la connettività deve disporre delle autorizzazioni per eseguire le seguenti operazioni EC2:
  - `ec2:AuthorizeSecurityGroupEgress`
  - `ec2:AuthorizeSecurityGroupIngress`
  - `ec2:CreateSecurityGroup`
  - `ec2:DescribeInstances`
  - `ec2:DescribeNetworkInterfaces`
  - `ec2:DescribeSecurityGroups`
  - `ec2:ModifyNetworkInterfaceAttribute`
  - `ec2:RevokeSecurityGroupEgress`

Quando si configura una connessione a un'istanza EC2, Amazon RDS opera in base alla configurazione corrente dei gruppi di sicurezza associati al cluster database Multi-AZ e all'istanza EC2, come descritto nella tabella seguente.

Configurazione del gruppo di sicurezza RDS corrente	Configurazione del gruppo di sicurezza EC2 corrente	Operazione RDS
Esistono uno o più gruppi di sicurezza associati al cluster di database multi-AZ con un nome che corrisponde al modello `rds-ec2-`*n* (dove *n* è un numero). Un gruppo di sicurezza che corrispon de al modello non è stato modificato. Questo gruppo di sicurezza include una sola regola in uscita con il gruppo di sicurezza VPC dell'istanza EC2 come origine.	Esistono uno o più gruppi di sicurezza associati all'istan za EC2 con un nome che corrisponde al modello `rds-ec2-`*n* (dove *n* è un numero). Un gruppo di sicurezza che corrisponde al modello non è stato modificato. Questo gruppo di sicurezza ha una sola regola in uscita con il gruppo di sicurezza VPC del cluster di database multi-AZ come origine.	Amazon RDS non esegue alcuna azione. Una connessione è già configurata automaticamente tra l'istanza EC2 e il cluster di database multi-AZ. Poiché esiste già una connessione tra l'istanza EC2 e il database RDS, i gruppi di sicurezza non vengono modificati.

Configurazione del gruppo di sicurezza RDS corrente	Configurazione del gruppo di sicurezza EC2 corrente	Operazione RDS
Si applica una delle due condizioni seguenti: • Non esiste un gruppo di sicurezza associato al cluster di database multi-AZ con un nome che corrispon de al modello `rds-ec2-`*n*. • Esistono uno o più gruppi di sicurezza associati al cluster di database multi-AZ con un nome che corrisponde al modello `rds-ec2-`*n*. Tuttavia, nessuno di questi gruppi di sicurezza può essere utilizzato per la connessio ne all'istanza EC2. Un gruppo di sicurezza non può essere utilizzato se non dispone di una regola in entrata con il gruppo di sicurezza VPC dell'istanza EC2 come origine. Inoltre, un gruppo di sicurezza non può essere utilizzato se è stato modificato. Esempi di modifiche sono l'aggiunta di una regola o la modifica della porta di una regola esistente.	Si applica una delle due condizioni seguenti: • Non esiste un gruppo di sicurezza associato all'istan za EC2 con un nome che corrisponde al modello `ec2-rds-`*n*. • Esistono uno o più gruppi di sicurezza associati all'istan za EC2 con un nome che corrisponde al modello `ec2-rds-`*n*. Tuttavia, nessuno di questi gruppi di sicurezza può essere utilizzato per la connessio ne al cluster di database multi-AZ. Un gruppo di sicurezza non può essere utilizzato se non dispone di una regola in uscita con il gruppo di sicurezza VPC del cluster di database multi-AZ come origine. Inoltre, un gruppo di sicurezza non può essere utilizzato se è stato modificato.	RDS action: create new security groups

Configurazione del gruppo di sicurezza RDS corrente	Configurazione del gruppo di sicurezza EC2 corrente	Operazione RDS
Esistono uno o più gruppi di sicurezza associati al cluster di database multi-AZ con un nome che corrispon de al modello `rds-ec2-`*n*. Un gruppo di sicurezza che corrisponde al modello non è stato modificato. Questo gruppo di sicurezza include una sola regola in uscita con il gruppo di sicurezza VPC dell'istanza EC2 come origine.	Esistono uno o più gruppi di sicurezza associati all'istan za EC2 con un nome che corrisponde al modello `ec2-rds-`*n*. Tuttavia, nessuno di questi gruppi di sicurezza può essere utilizzato per la connessione al cluster di database multi-AZ. Un gruppo di sicurezza non può essere utilizzato se non dispone di una regola in uscita con il gruppo di sicurezza VPC del cluster di database multi-AZ come origine. Inoltre, un gruppo di sicurezza non può essere utilizzato se è stato modificato.	RDS action: create new security groups
Esistono uno o più gruppi di sicurezza associati al cluster di database multi-AZ con un nome che corrispon de al modello `rds-ec2-`*n*. Un gruppo di sicurezza che corrisponde al modello non è stato modificato. Questo gruppo di sicurezza include una sola regola in uscita con il gruppo di sicurezza VPC dell'istanza EC2 come origine.	Esiste un gruppo di sicurezza EC2 valido per la connessio ne, ma non è associato all'istanza EC2. Questo gruppo di sicurezza ha un nome che corrisponde al modello `rds-ec2-`*n*. Non è stato modificato. Dispone di una sola regola in uscita con il gruppo di sicurezza VPC del cluster di database multi-AZ come origine.	RDS action: associate EC2 security group

Configurazione del gruppo di sicurezza RDS corrente	Configurazione del gruppo di sicurezza EC2 corrente	Operazione RDS
Si applica una delle due condizioni seguenti: • Non esiste un gruppo di sicurezza associato al cluster di database multi-AZ con un nome che corrisponde al modello `rds-ec2-`*n*. • Esistono uno o più gruppi di sicurezza associati al cluster di database multi-AZ con un nome che corrisponde al modello `rds-ec2-`*n*. Tuttavia, nessuno di questi gruppi di sicurezza può essere utilizzato per la connessione all'istanza EC2. Un gruppo di sicurezza non può essere utilizzato se non dispone di una regola in entrata con il gruppo di sicurezza VPC dell'istanza EC2 come origine. Inoltre, un gruppo di sicurezza non può essere utilizzato se è stato modificato.	Esistono uno o più gruppi di sicurezza associati all'istanza EC2 con un nome che corrisponde al modello `rds-ec2-`*n*. Un gruppo di sicurezza che corrisponde al modello non è stato modificato. Questo gruppo di sicurezza ha una sola regola in uscita con il gruppo di sicurezza VPC del cluster di database multi-AZ come origine.	RDS action: create new security groups

RDS: creazione di nuovi gruppi di sicurezza

Amazon RDS esegue le seguenti operazioni:

- Crea un nuovo gruppo di sicurezza che corrisponde al modello `rds-ec2-`*n*. Questo gruppo di sicurezza include una regola in uscita con il gruppo di sicurezza VPC dell'istanza EC2 come origine. Questo gruppo di sicurezza associato al cluster di database multi-AZ consente all'istanza EC2 di accedere al cluster di database multi-AZ.
- Crea un nuovo gruppo di sicurezza che corrisponde al modello `ec2-rds-`*n*. Questo gruppo di sicurezza include una regola in uscita con il gruppo di sicurezza VPC del cluster di database multi-AZ come origine. Questo gruppo di sicurezza è associato all'istanza EC2 e consente all'istanza EC2 di inviare il traffico al cluster di database multi-AZ.

Azione RDS: associazione del gruppo di sicurezza EC2

Amazon RDS associa il gruppo di sicurezza EC2 esistente, valido all'istanza EC2. Questo gruppo di sicurezza consente all'istanza EC2 di inviare traffico al cluster di database multi-AZ.

Connessione automatica di un'istanza EC2 e un cluster di database multi-AZ

Prima di configurare una connessione tra un'istanza EC2 e un database RDS assicurati di aver soddisfatto i requisiti descritti in Panoramica della connettività automatica con un'istanza EC2.

Se modifichi i gruppi di sicurezza dopo avere configurato la connettività, le modifiche potrebbero influenzare la connessione tra l'istanza EC2 e il database RDS.

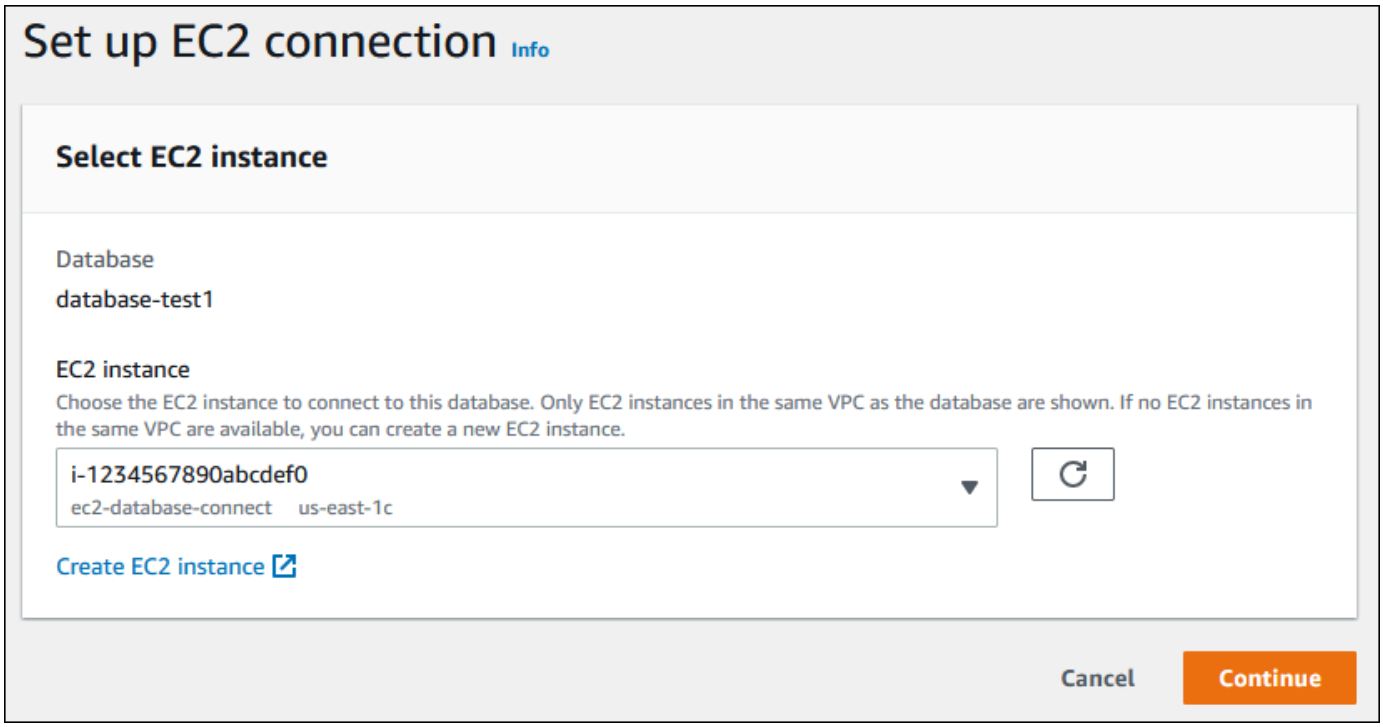
> **Note**
>
> È possibile configurare automaticamente una connessione tra un'istanza EC2 e un database RDS solo utilizzando la AWS Management Console. Non puoi configurare automaticamente una connessione con l'API AWS CLI o RDS.

Per connettere automaticamente un'istanza EC2 e un database RDS

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, scegli Databases (Database), quindi seleziona il database RDS.
3. In Operazioni, scegli Configura connessione EC2.

   Viene visualizzata la pagina Set up EC2 connection (Configura connessione EC2).

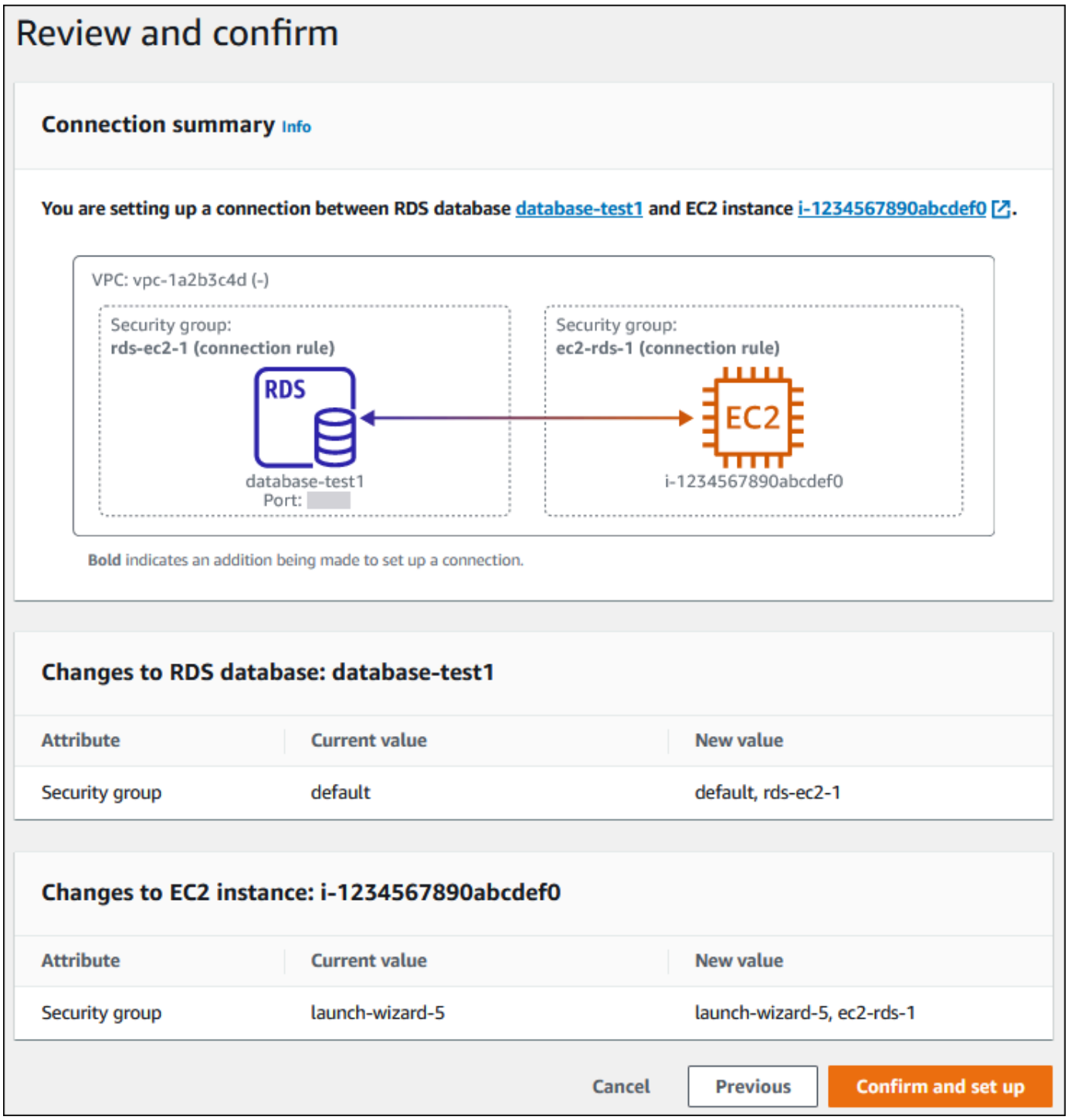
4. Nella pagina Set up EC2 connection (Configura connessione EC2), scegli l'istanza EC2.



   Se nello stesso VPC non esistono istanze EC2, scegli Create EC2 instance (Crea istanza EC2) per crearne una. In questo caso, assicurati che la nuova istanza EC2 si trovi nello stesso VPC del database RDS.

5. Scegli Continua.

   Viene visualizzata la pagina Review and confirm (Rivedi e conferma).



6. Nella pagina Review and confirm (Rivedi e conferma), esamina le modifiche che RDS apporterà per configurare la connettività con l'istanza EC2.

   Se le modifiche sono corrette, scegli Conferma e configura.

   Se le modifiche non sono corrette, scegli Previous (Precedente) o Cancel (Annulla).

Visualizzazione delle risorse di calcolo connesse

È possibile utilizzare il AWS Management Console per visualizzare le risorse di calcolo connesse a un cluster DB di database RDS. Le risorse mostrate includono le connessioni delle risorse di calcolo configurate automaticamente. È possibile configurare la connettività delle risorse di calcolo automaticamente nei modi seguenti:

- È possibile selezionare la risorsa di calcolo quando si crea il database.

  Per ulteriori informazioni, consultare Creazione di un'istanza database Amazon RDS e Creazione di un cluster di database Multi-AZ.
- È possibile configurare la connettività tra un database esistente e una risorsa di calcolo.

  Per ulteriori informazioni, consulta Connessione automatica di un'istanza EC2 e di un database RDS.

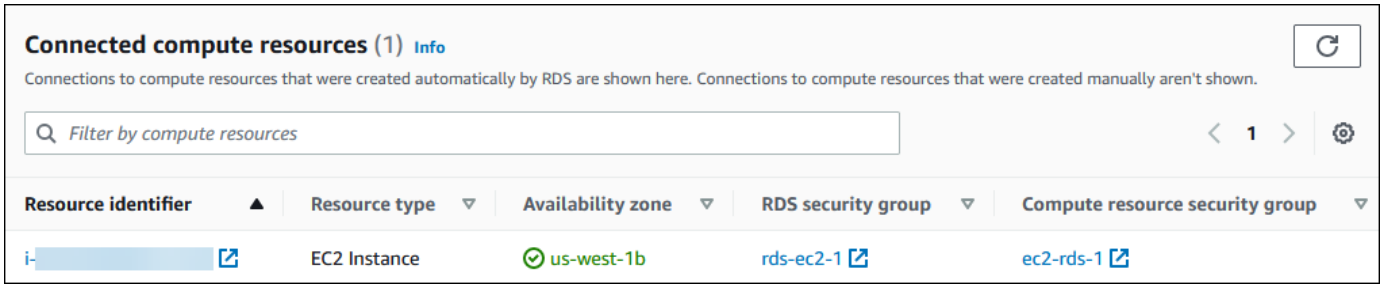
Le risorse di calcolo elencate non includono quelle connesse al database manualmente. Ad esempio, è possibile consentire manualmente a una risorsa di calcolo di accedere a un database aggiungendo una regola al gruppo di sicurezza VPC associato al database.

Per garantire la presenza della risorsa di calcolo nell'elenco, è necessario che siano soddisfatte le condizioni elencate di seguito.

- Il nome del gruppo di sicurezza associato alla risorsa di calcolo corrisponde al modello `ec2-rds-`*n* (dove *n* è un numero).
- Il gruppo di sicurezza associato alla risorsa di calcolo ha una regola in uscita con l'intervallo di porte impostato sulla porta utilizzata dal database RDS.
- Il gruppo di sicurezza associato alla risorsa di calcolo ha una regola in uscita con l'origine impostata su un gruppo di sicurezza associato al database RDS.
- Il nome del gruppo di sicurezza associato al database RDS corrisponde al modello `rds-ec2-`*n* (dove *n* è un numero).
- Il gruppo di sicurezza associato al database RDS ha una regola in entrata con l'intervallo di porte impostato sulla porta utilizzata dal database RDS.
- Il gruppo di sicurezza associato al database RDS ha una regola in entrata con l'origine impostata su un gruppo di sicurezza associato alla risorsa di calcolo.

Per visualizzare le risorse di calcolo connesse a un database RDS

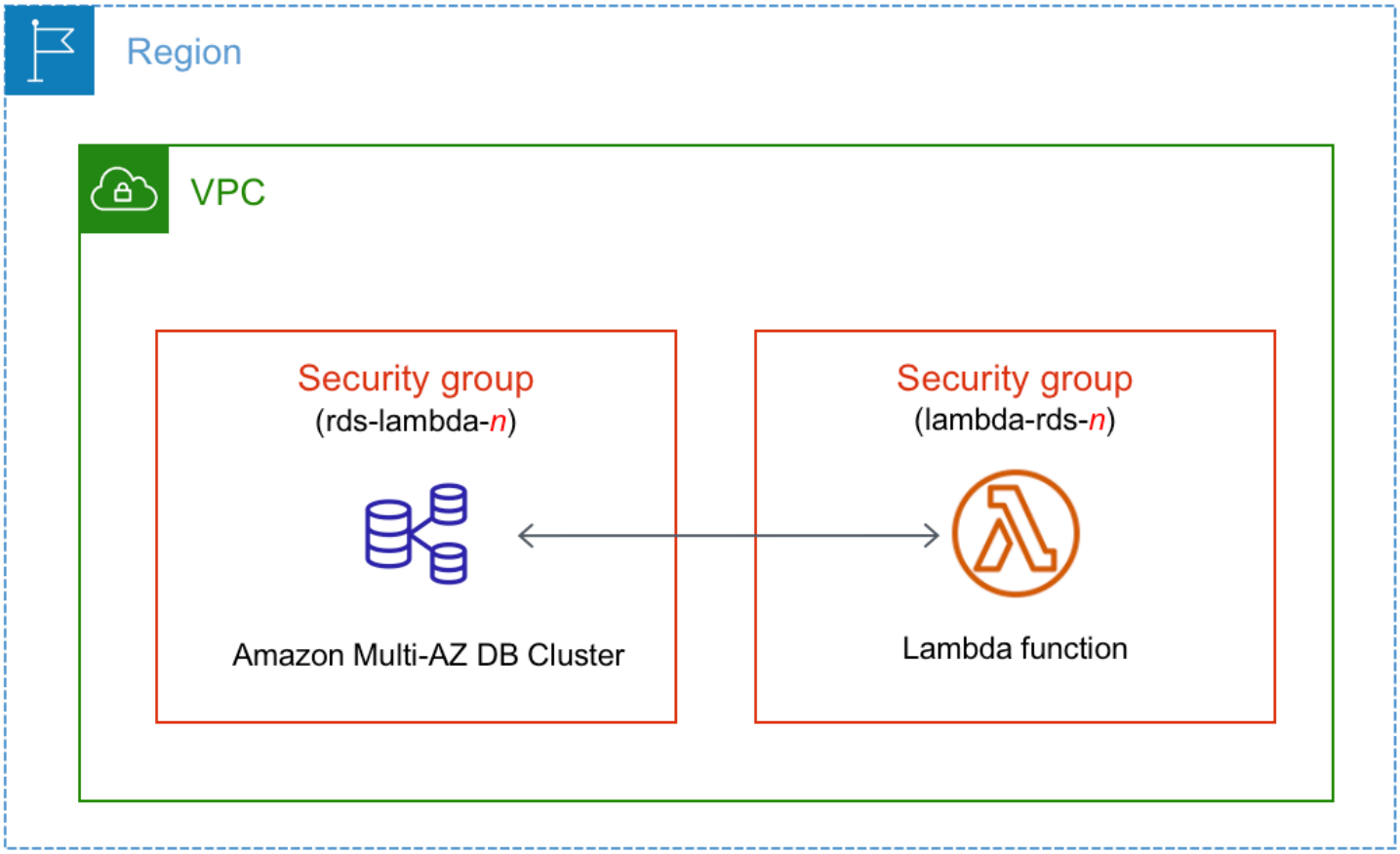
1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, scegli Databases (Database), quindi seleziona il nome del database RDS.
3. Nella scheda Connectivity & security (Connettività e sicurezza), visualizza le risorse di calcolo in Connected compute resources (Risorse di calcolo connesse).



Connessione automatica di una funzione Lambda e di un cluster database Multi-AZ

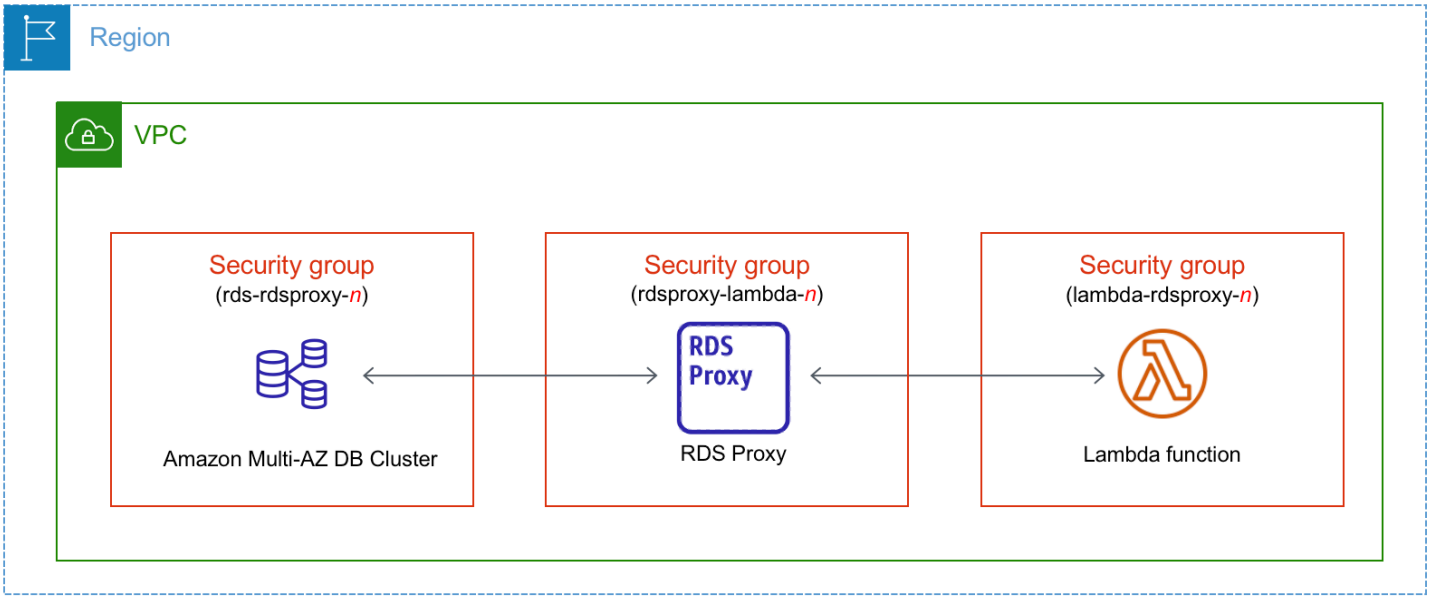
È possibile usare la console RDS per semplificare l'impostazione di una connessione tra una funzione Lambda e un cluster database Multi-AZ. È possibile usare la console RDS per semplificare l'impostazione di una connessione tra una funzione Lambda e un cluster database Multi-AZ. Spesso, il cluster database Multi-AZ si trova in una sottorete privata all'interno di un VPC. La funzione Lambda può essere utilizzata dalle applicazioni per accedere al cluster database Multi-AZ privato.

L'immagine seguente mostra una connessione diretta tra il cluster database Multi-AZ e la funzione Lambda.



È possibile configurare la connessione tra la funzione Lambda e il database tramite RDS Proxy per migliorare le prestazioni e la resilienza del database. Spesso, le funzioni Lambda effettuano connessioni database brevi e frequenti che traggono vantaggio dal pool di connessioni offerto da RDS Proxy. È possibile sfruttare qualsiasi autenticazione IAM già disponibile per le funzioni Lambda, anziché gestire le credenziali del database nel codice dell'applicazione Lambda. Per ulteriori informazioni, consulta Utilizzo di Server proxy per Amazon RDS.

È possibile usare la console per creare automaticamente un proxy per la connessione. È anche possibile selezionare i proxy esistenti. La console aggiorna il gruppo di sicurezza proxy per consentire le connessioni dal database e dalla funzione Lambda. È possibile inserire le credenziali del database o selezionare il segreto di Secrets Manager richiesto per accedere al database.



Argomenti

- Panoramica della connettività automatica a una funzione Lambda
- Connessione automatica di una funzione Lambda e di un cluster database Multi-AZ
- Visualizzazione delle risorse di calcolo connesse

Panoramica della connettività automatica a una funzione Lambda

Quando si imposta automaticamente una connessione tra una funzione Lambda e un cluster database Multi-AZ, Amazon RDS configura il gruppo di sicurezza VPC per la funzione Lambda e per il cluster database.

Di seguito sono riportati i requisiti per connettere una funzione Lambda a un cluster database Multi-AZ:

- La funzione Lambda deve esistere nello stesso VPC del cluster database Multi-AZ.

  Se nessuna funzione Lambda esiste nello stesso VPC, la console fornisce un collegamento per crearne una.
- L'utente che configura la connettività deve disporre delle autorizzazioni per eseguire le seguenti operazioni Amazon RDS, Amazon EC2, Lambda, Secrets Manager e IAM:
  - Amazon RDS
    - `rds:CreateDBProxies`

    - `rds:DescribeDBInstances`
    - `rds:DescribeDBProxies`
    - `rds:ModifyDBInstance`
    - `rds:ModifyDBProxy`
    - `rds:RegisterProxyTargets`
- Amazon EC2
    - `ec2:AuthorizeSecurityGroupEgress`
    - `ec2:AuthorizeSecurityGroupIngress`
    - `ec2:CreateSecurityGroup`
    - `ec2:DeleteSecurityGroup`
    - `ec2:DescribeSecurityGroups`
    - `ec2:RevokeSecurityGroupEgress`
    - `ec2:RevokeSecurityGroupIngress`
- Lambda
    - `lambda:CreateFunctions`
    - `lambda:ListFunctions`
    - `lambda:UpdateFunctionConfiguration`
- Secrets Manager
    - `sercetsmanager:CreateSecret`
    - `secretsmanager:DescribeSecret`
- IAM
    - `iam:AttachPolicy`
    - `iam:CreateRole`
    - `iam:CreatePolicy`
- AWS KMS
    - `kms:describeKey`

Quando si imposta una connessione tra una funzione Lambda e un cluster database Multi-AZ, Amazon RDS configura il gruppo di sicurezza VPC per la funzione Lambda e per il cluster database Multi-AZ. Se si utilizza il proxy RDS, Amazon RDS configura anche il gruppo di sicurezza VPC per

il proxy. Amazon RDS opera in base alla configurazione corrente dei gruppi di sicurezza associati all'istanza al cluster database Multi-AZ e alla funzione Lambda, come descritto nella tabella seguente.

Configurazione del gruppo di sicurezza RDS corrente	Configurazione del gruppo di sicurezza Lambda corrente	Configurazione del gruppo di sicurezza proxy corrente	Operazione RDS
Amazon RDS non esegue azioni perché i gruppi di sicurezza di tutte le risorse seguono lo schema di denominazione corretto e dispongono delle regole in entrata e in uscita corrette.	Esistono uno o più gruppi di sicurezza associati al cluster database Multi-AZ con un nome che corrisponde al modello `rds-lambda-` *n* (dove *n* è un numero) o se il `TargetHealth` di un proxy associato è `AVAILABLE`. Un gruppo di sicurezza che corrisponde al modello non è stato modificato. Questo gruppo di sicurezza include una sola regola in entrata con il gruppo di sicurezza VPC della funzione Lambda o del proxy come l'origine.	Esistono uno o più gruppi di sicurezza associati alla funzione Lambda con un nome che corrisponde al modello `lambda-rds-` *n* o `lambda-rdsproxy-` *n* (dove *n* è un numero). Un gruppo di sicurezza che corrisponde al modello non è stato modificato. Questo gruppo di sicurezza dispone di una sola regola in uscita con il gruppo di sicurezza VPC dell'istanza del cluster database Multi-AZ o il proxy come la destinazione.	Esistono uno o più gruppi di sicurezza associati al proxy con un nome che corrisponde al modello `rdsproxy-lambda-` *n* (dove *n* è un numero). Un gruppo di sicurezza che corrisponde al modello non è stato modificato. Questo gruppo di sicurezza dispone di regole in entrata e in uscita con i gruppi di sicurezza VPC della funzione Lambda e il cluster database Multi-AZ.
Si applica una delle due condizioni seguenti:	Si applica una delle due condizioni seguenti:	Si applica una delle due condizioni seguenti:	RDS action: create new security groups

Configurazione del gruppo di sicurezza RDS corrente	Configurazione del gruppo di sicurezza Lambda corrente	Configurazione del gruppo di sicurezza proxy corrente	Operazione RDS
• Non esiste alcun gruppo di sicurezza associato al cluster database Multi-AZ con un nome che corrisponde al modello `rds-lambda-` *n* o se il `TargetHealth` di un proxy associato è `AVAILABLE`. • Esistono uno o più gruppi di sicurezza associati al cluster database Multi-AZ con un nome che corrisponde al modello `rds-lambda-` *n* o se `TargetHealth` di un proxy associato è `AVAILABLE`. Tuttavia, Amazon RDS non può utilizzare nessuno di questi gruppi di sicurezza per la connessione alla funzione Lambda.	• Non esiste un gruppo di sicurezza associato alla funzione Lambda con un nome che corrisponde al modello `lambda-rds-` *n* o `lambda-rdsproxy-` *n*. • Esistono uno o più gruppi di sicurezza associati alla funzione Lambda con un nome che corrisponde al modello `lambda-rds-` *n* o `lambda-rdsproxy-` *n*. Tuttavia, Amazon RDS non può utilizzare nessuno di questi gruppi di sicurezza per la connessione al cluster database Multi-AZ. Amazon RDS non può utilizzare un gruppo di sicurezza se non	• Non esiste un gruppo di sicurezza associato al proxy con un nome che corrisponde al modello `rdsproxy-lambda-` *n*. • Esistono uno o più gruppi di sicurezza associati al proxy con un nome che corrisponde al modello `rdsproxy-lambda-` *n*. Tuttavia, Amazon RDS non può utilizzare nessuno di questi gruppi di sicurezza per la connessione al cluster database Multi-AZ o alla funzione Lambda. Amazon RDS non può utilizzare un gruppo di sicurezza che non dispone di una regola in entrata e in uscita con il gruppo	

Configurazione del gruppo di sicurezza RDS corrente	Configurazione del gruppo di sicurezza Lambda corrente	Configurazione del gruppo di sicurezza proxy corrente	Operazione RDS
Amazon RDS non può utilizzare un gruppo di sicurezza che non dispone di una regola in entrata con il gruppo di sicurezza VPC della funzione Lambda o del proxy come l'origine. Inoltre, Amazon RDS non può utilizzare un gruppo di sicurezza che è stato modificato. Esempi di modifiche sono l'aggiunta di una regola o la modifica della porta di una regola esistente.	dispone di una regola in uscita con il gruppo di sicurezza VPC del cluster database Multi-AZ o proxy come origine. Inoltre, Amazon RDS non può utilizzare un gruppo di sicurezza che è stato modificato.	di sicurezza VPC del cluster database Multi-AZ e la funzione Lambda. Inoltre, Amazon RDS non può utilizzare un gruppo di sicurezza che è stato modificato.	

Configurazione del gruppo di sicurezza RDS corrente	Configurazione del gruppo di sicurezza Lambda corrente	Configurazione del gruppo di sicurezza proxy corrente	Operazione RDS
Esistono uno o più gruppi di sicurezza associati al cluster database Multi-AZ con un nome che corrisponde al modello `rds-lambda-` *n* o se `TargetHealth` di un proxy associato è `AVAILABLE`. Un gruppo di sicurezza che corrisponde al modello non è stato modificato. Questo gruppo di sicurezza include una sola regola in entrata con il gruppo di sicurezza VPC della funzione Lambda o del proxy come l'origine.	Esistono uno o più gruppi di sicurezza associati alla funzione Lambda con un nome che corrisponde al modello `lambda-rds-` *n* o `lambda-rdsproxy-` *n*. Tuttavia, Amazon RDS non può utilizzare nessuno di questi gruppi di sicurezza per la connessione al cluster database Multi-AZ. Amazon RDS non può utilizzare un gruppo di sicurezza che non dispone di una regola in uscita con il gruppo di sicurezza VPC del cluster database Multi-AZ come la destinazione. Inoltre, Amazon RDS non può utilizzare un gruppo di sicurezza che è stato modificato.	Esistono uno o più gruppi di sicurezza associati al proxy con un nome che corrisponde al modello `rdsproxy-lambda-` *n*. Tuttavia, Amazon RDS non può utilizzare nessuno di questi gruppi di sicurezza per la connessione al cluster database Multi-AZ o alla funzione Lambda. Amazon RDS non può utilizzare un gruppo di sicurezza che non dispone di una regola in entrata e in uscita con il gruppo di sicurezza VPC del cluster database Multi-AZ e la funzione Lambda. Inoltre, Amazon RDS non può utilizzare un gruppo di sicurezza che è stato modificato.	RDS action: create new security groups

Configurazione del gruppo di sicurezza RDS corrente	Configurazione del gruppo di sicurezza Lambda corrente	Configurazione del gruppo di sicurezza proxy corrente	Operazione RDS
Esistono uno o più gruppi di sicurezza associati al cluster database Multi-AZ con un nome che corrisponde al modello `rds-lambda-` *n* o se `TargetHealth` di un proxy associato è `AVAILABLE`. Un gruppo di sicurezza che corrisponde al modello non è stato modificato. Questo gruppo di sicurezza include una sola regola in entrata con il gruppo di sicurezza VPC della funzione Lambda o del proxy come l'origine.	Esiste un gruppo di sicurezza Lambda valido per la connessione, ma non è associato alla funzione Lambda. Questo gruppo di sicurezza ha un nome che corrisponde al modello `lambda-rds-` *n* o `lambda-rdsproxy-` *n*. Non è stato modificato. Dispone di una sola regola in uscita con il gruppo di sicurezza VPC del cluster database Multi-AZ o del proxy come la destinazione.	Esiste un gruppo di sicurezza proxy valido per la connessione, ma non è associato al proxy. Questo gruppo di sicurezza ha un nome che corrisponde al modello `rdsproxy-lambda-` *n*. Non è stato modificato. Dispone di regole in entrata e in uscita con il gruppo di sicurezza VPC del cluster database Multi-AZ e della funzione Lambda.	RDS action: associate Lambda security group

Configurazione del gruppo di sicurezza RDS corrente	Configurazione del gruppo di sicurezza Lambda corrente	Configurazione del gruppo di sicurezza proxy corrente	Operazione RDS
Si applica una delle due condizioni seguenti: • Non esiste alcun gruppo di sicurezza associato al cluster database Multi-AZ con un nome che corrisponde al modello `rds-lambda-` *n* o se il `TargetHealth` di un proxy associato è `AVAILABLE` . • Esistono uno o più gruppi di sicurezza associati al cluster database Multi-AZ con un nome che corrisponde al modello `rds-lambda-` *n* o se `TargetHealth` di un proxy associato è `AVAILABLE` . Tuttavia, Amazon RDS non può	Esistono uno o più gruppi di sicurezza associati alla funzione Lambda con un nome che corrisponde al modello `lambda-rds-` *n* o `lambda-rdsproxy-` *n*. Un gruppo di sicurezza che corrisponde al modello non è stato modificato. Questo gruppo di sicurezza dispone di una sola regola in uscita con il gruppo di sicurezza VPC del cluster database Multi-AZ o il proxy come la destinazione.	Esistono uno o più gruppi di sicurezza associati al proxy con un nome che corrisponde al modello `rdsproxy-lambda-` *n*. Un gruppo di sicurezza che corrisponde al modello non è stato modificato. Questo gruppo di sicurezza dispone di regole in entrata e in uscita con il gruppo di sicurezza VPC del cluster database Multi-AZ e della funzione Lambda.	RDS action: create new security groups

Configurazione del gruppo di sicurezza RDS corrente	Configurazione del gruppo di sicurezza Lambda corrente	Configurazione del gruppo di sicurezza proxy corrente	Operazione RDS
utilizzare nessuno di questi gruppi di sicurezza per la connessione alla funzione Lambda o al proxy. Amazon RDS non può utilizzare un gruppo di sicurezza che non dispone di una regola in entrata con il gruppo di sicurezza VPC della funzione Lambda o del proxy come l'origine. Inoltre, Amazon RDS non può utilizzare un gruppo di sicurezza che è stato modificato.			

Configurazione del gruppo di sicurezza RDS corrente	Configurazione del gruppo di sicurezza Lambda corrente	Configurazione del gruppo di sicurezza proxy corrente	Operazione RDS
Esistono uno o più gruppi di sicurezza associati al cluster di database multi-AZ con un nome che corrisponde al modello `rds-rdsproxy-` *n* (dove *n* è un numero).	Si applica una delle due condizioni seguenti: • Non esiste un gruppo di sicurezza associato alla funzione Lambda con un nome che corrisponde al modello `lambda-rds-` *n* o `lambda-rdsproxy-` *n*. • Esistono uno o più gruppi di sicurezza associati alla funzione Lambda con un nome che corrisponde al modello `lambda-rds-` *n* o `lambda-rdsproxy-` *n*. Tuttavia, Amazon RDS non può utilizzare nessuno di questi gruppi di sicurezza per la connessione al cluster database Multi-AZ.	Si applica una delle due condizioni seguenti: • Non esiste un gruppo di sicurezza associato al proxy con un nome che corrispon de al modello `rdsproxy-lambda-` *n*. • Esistono uno o più gruppi di sicurezza associati al proxy con un nome che corrispon de al modello `rdsproxy-lambda-` *n*. Tuttavia, Amazon RDS non può utilizzare nessuno di questi gruppi di sicurezza per la connessione al cluster database Multi-AZ o alla funzione Lambda.	RDS action: create new security groups

Configurazione del gruppo di sicurezza RDS corrente	Configurazione del gruppo di sicurezza Lambda corrente	Configurazione del gruppo di sicurezza proxy corrente	Operazione RDS
	Amazon RDS non può utilizzare un gruppo di sicurezza che non dispone di una regola in uscita con il gruppo di sicurezza VPC del cluster database Multi-AZ come la destinazione. Inoltre, Amazon RDS non può utilizzare un gruppo di sicurezza che è stato modificato.	Amazon RDS non può utilizzare un gruppo di sicurezza che non dispone di una regola in entrata e in uscita con il gruppo di sicurezza VPC del cluster database Multi-AZ e la funzione Lambda. Inoltre, Amazon RDS non può utilizzare un gruppo di sicurezza che è stato modificato.	

Azione RDS: creazione di nuovi gruppi di sicurezza

Amazon RDS esegue le seguenti operazioni:

- Crea un nuovo gruppo di sicurezza che corrisponde al modello `rds-lambda-`*n*. Questo gruppo di sicurezza dispone di una regola in entrata con il gruppo di sicurezza VPC della funzione Lambda o del proxy come l'origine. Questo gruppo di sicurezza è associato al cluster database Multi-AZ e consente alla funzione o al proxy di accedere al cluster database Multi-AZ.
- Crea un nuovo gruppo di sicurezza che corrisponde al modello `lambda-rds-`*n*. Questo gruppo di sicurezza dispone di una regola in uscita con il gruppo di sicurezza VPC del cluster database Multi-AZ o il proxy come la destinazione. Questo gruppo di sicurezza è associato alla funzione Lambda e consente alla funzione Lambda di inviare traffico al cluster database Multi-AZ o inviare traffico tramite un proxy.
- Crea un nuovo gruppo di sicurezza che corrisponde al modello `rdsproxy-lambda-`*n*. Questo gruppo di sicurezza dispone di regole in entrata e in uscita con il gruppo di sicurezza VPC del cluster database Multi-AZ e della funzione Lambda.

Azione RDS: associazione del gruppo di sicurezza Lambda

Amazon RDS associa il gruppo di sicurezza Lambda valido, esistente alla funzione Lambda. Questo gruppo di sicurezza consente alla funzione di inviare traffico al cluster database Multi-AZ o inviare traffico tramite un proxy.

Connessione automatica di una funzione Lambda e di un cluster database Multi-AZ

È possibile utilizzare la console Amazon RDS per connettere automaticamente una funzione Lambda al cluster database Multi-AZ. Ciò semplifica il processo di configurazione di una connessione tra queste risorse.

È anche possibile usare RDS Proxy per includere un proxy nella connessione. Funzioni Lambda effettuano connessioni database brevi e frequenti che traggono vantaggio dal pool di connessioni offerto da RDS Proxy. È anche possibile utilizzare qualsiasi autenticazione IAM che è già stata configurata per le funzioni Lambda, anziché gestire le credenziali del database nel codice dell'applicazione Lambda.

È possibile connettere un cluster database Multi-AZ esistente a funzioni Lambda nuove ed esistenti utilizzando la pagina Configurazione della connessione Lambda. Il processo di configurazione consente di impostare automaticamente i gruppi di sicurezza richiesti.

Prima di configurare una connessione tra una funzione Lambda e un cluster database Multi-AZ, assicurati che:

- La funzione Lambda e il cluster database Multi-AZ si trovino nello stesso VPC.
- Disponi delle autorizzazioni corrette per l'account utente. Per ulteriori informazioni sui requisiti, consulta Panoramica della connettività automatica a una funzione Lambda.

Se modifichi i gruppi di sicurezza dopo la configurazione della connettività, le modifiche potrebbero influenzare la connessione tra la funzione Lambda e il cluster database Multi-AZ.

> **Note**
>
> È possibile configurare automaticamente una connessione tra un cluster database Multi-AZ e una funzione Lambda solo nella AWS Management Console. Per connettere una funzione Lambda, tutte le istanze nel cluster database Multi-AZ devono essere nello stato Disponibile.

Per connettere automaticamente una funzione Lambda e un cluster database Multi-AZ

<result>

Dopo aver confermato la configurazione, Amazon RDS avvia il processo di connessione della funzione Lambda, di RDS Proxy (se hai usato un proxy) e del cluster database Multi-AZ. La console mostra la finestra di dialogo Dettagli di connessione, in cui sono elencate le modifiche al gruppo di sicurezza che consentono le connessioni tra le risorse.
</result>

1. Accedi alla AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, scegli Database, quindi seleziona il cluster database Multi-AZ che desideri connettere a una funzione Lambda.
3. Per Operazioni, scegli Configura connessione Lambda.
4. Nella pagina Configurazione della connessione Lambda, in Seleziona la funzione Lambda, effettua una delle seguenti operazioni:
   - Se disponi di una funzione Lambda esistente nello stesso VPC del cluster database Multi-AZ, seleziona Scegli una funzione esistente, quindi scegli la funzione.
   - Se non disponi di una funzione Lambda nello stesso VPC, seleziona Crea una nuova funzione, quindi inserisci un Nome della funzione. Il runtime predefinito è impostato su Nodejs.18. Puoi modificare le impostazioni per la nuova funzione Lambda nella console Lambda dopo aver completato la configurazione della connessione.
5. (Facoltativo) In RDS Proxy, seleziona Connessione tramite RDS Proxy, quindi esegui una delle seguenti operazioni:
   - Se disponi di un proxy esistente che desideri utilizzare, seleziona Scegli un proxy esistente, quindi seleziona il proxy.
   - Se non disponi di un proxy e desideri che uno venga creato automaticamente da Amazon RDS, seleziona, seleziona Crea nuovo proxy. Quindi, per Credenziali del database, esegui una delle seguenti operazioni:

     a. Seleziona Nome utente e password del database, quindi inserisci Nome utente e Password per il cluster database Multi-AZ.

     b. Seleziona Segreti Secrets Manager. Quindi, per Seleziona segreto, scegli un segreto AWS Secrets Manager. Se non disponi di un segreto di Secrets Manager, seleziona Crea nuovo segreto di Secrets Manager per creare un nuovo segreto. Dopo aver creato il segreto, per Seleziona segreto, scegli il nuovo segreto.

Dopo aver creato il nuovo proxy, seleziona Scegli un proxy esistente, quindi seleziona il proxy. Tieni presente che prima che il proxy sia disponibile per la connessione, potrebbe trascorrere del tempo.

6. (Facoltativo) Espandi Riepilogo della connessione e verifica gli aggiornamenti evidenziati per le risorse.
7. Scegliere Set up (Configura).

Visualizzazione delle risorse di calcolo connesse

È possibile utilizzare la AWS Management Console per visualizzare le risorse di calcolo connesse al cluster database Multi-AZ. Le risorse mostrate includono le connessioni delle risorse di calcolo configurate automaticamente da Amazon RDS.

Le risorse di calcolo elencate non includono quelle connesse manualmente al cluster database Multi-AZ. Ad esempio, è possibile autorizzare manualmente una risorsa di calcolo ad accedere al cluster database Multi-AZ aggiungendo una regola al gruppo di sicurezza VPC associato al cluster.

Affinché la console elenchi una funzione Lambda, devono essere soddisfatte le seguenti condizioni:

- Il nome del gruppo di sicurezza associato alla risorsa di calcolo corrisponde al modello `lambda-rds-`*n* o `lambda-rdsproxy-`*n* (dove *n* è un numero).
- Il gruppo di sicurezza associato alla risorsa di calcolo dispone di una regola in uscita con l'intervallo di porte impostato sulla porta del cluster database Multi-AZ o di un proxy associato. La destinazione della regola in uscita deve essere impostata su un gruppo di sicurezza associato al cluster database Multi-AZ o a un proxy associato.
- Il nome del gruppo di sicurezza collegato al proxy associato al database corrisponde al modello `rds-rdsproxy-`*n* (dove *n* è un numero).
- Il gruppo di sicurezza associato alla funzione dispone di una regola in uscita con la porta impostata sulla porta utilizzata dal cluster database Multi-AZ o da un proxy associato. La destinazione deve essere impostata su un gruppo di sicurezza associato al cluster database Multi-AZ o a un proxy associato.

Per visualizzare le risorse di calcolo connesse automaticamente a un cluster database Multi-AZ

1. Accedi alla AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.

2. Nel riquadro di navigazione, scegli Database, quindi seleziona il cluster database Multi-AZ.
3. Nella scheda Connettività e sicurezza, visualizza le risorse di calcolo in Risorse di calcolo connesse.

Modifica di un cluster di database Multi-AZ

Un cluster di database Multi-AZ ha un'istanza database di scrittore e due istanze database di lettore in tre zone di disponibilità separate. I cluster di database multi-AZ offrono elevata disponibilità, maggiore capacità per i carichi di lavoro in lettura e minore latenza rispetto alle implementazioni Multi-AZ. Per ulteriori informazioni sui cluster di database Multi-AZ, consulta Implementazioni cluster di database multi-AZ.

È possibile modificare un cluster database Multi-AZ per modificarne le impostazioni. È inoltre possibile eseguire operazioni su un cluster di database Multi-AZ, ad esempio farne uno snapshot.

> **Important**
>
> Non è possibile modificare le istanze DB all'interno di un cluster DB Multi-AZ. Tutte le modifiche devono essere eseguite a livello di cluster DB. L'unica operazione che è possibile eseguire su un'istanza DB all'interno di un cluster DB Multi-AZ è il riavvio.

È possibile modificare un cluster DB Multi-AZ utilizzando l'API AWS Management Console AWS CLI, the o RDS.

Console

Per modificare un cluster di database Multi-AZ

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel pannello di navigazione, scegliere Databases (Database), quindi scegliere il cluster di database Multi-AZ che si desidera modificare.
3. Scegliere Modify (Modifica). Viene visualizzata la pagina Modify DB cluster (Modifica cluster di database).
4. Modificare le impostazioni desiderate. Per informazioni su ciascuna impostazione, consulta Impostazioni per la creazione di cluster di database Multi-AZ.
5. Quando tutte le modifiche sono come le desideri, seleziona Continue (Continua) e controlla il riepilogo delle modifiche.
6. (Facoltativo) Scegliere Applica immediatamente per applicare immediatamente le modifiche. In alcuni casi, la chiusura di questa opzione può causare un'interruzione. Per ulteriori informazioni, consulta Applicazione immediata delle modifiche).

7. Nella pagina di conferma esaminare le modifiche. Se sono corrette, selezionare Modify cluster DB (Modifica cluster di database) per salvare le modifiche.

   Oppure scegliere Back (Indietro) per cambiare le modifiche o Cancel (Annulla) per annullare le modifiche.

AWS CLI

Per modificare un cluster DB Multi-AZ utilizzando AWS CLI, chiama il modify-db-clustercomando. Specifica l'identificatore cluster di database e i valori per le impostazioni da modificare. Per ulteriori informazioni su ciascuna opzione, consulta Impostazioni per la creazione di cluster di database Multi-AZ.

Example

Il codice seguente modifica `my-multi-az-dbcluster` impostando il periodo di retention dei backup a 1 settimana (7 giorni). Il codice abilita la protezione da eliminazione utilizzando `--deletion-protection`. Per disattivare la protezione da eliminazione, utilizzare `--no-deletion-protection`. Le modifiche vengono applicate durante la prossima finestra di manutenzione utilizzando `--no-apply-immediately`. Utilizza `--apply-immediately` per applicare immediatamente le modifiche. Per ulteriori informazioni, consulta Applicazione immediata delle modifiche).

Per LinuxmacOS, oUnix:

```
aws rds modify-db-cluster \
    --db-cluster-identifier my-multi-az-dbcluster \
    --backup-retention-period 7 \
    --deletion-protection \
    --no-apply-immediately
```

Per Windows:

```
aws rds modify-db-cluster ^
    --db-cluster-identifier my-multi-az-dbcluster ^
    --backup-retention-period 7 ^
    --deletion-protection ^
    --no-apply-immediately
```

API RDS

Per modificare un cluster di database Multi-AZ tramite l'API Amazon RDS, chiamare l'operazione ModifyDBCluster. Specifica l'identificatore cluster di database e i parametri per le impostazioni da modificare. Per informazioni su ciascun parametro, consulta Impostazioni per la creazione di cluster di database Multi-AZ.

Applicazione immediata delle modifiche)

Quando modifichi un cluster di database Multi-AZ, puoi applicare le modifiche immediatamente. Per applicare le modifiche immediatamente, scegli l'opzione Applica immediatamente nella AWS Management Console. Oppure usi l'`--apply-immediately`opzione quando chiami AWS CLI o imposta il `ApplyImmediately` parametro su `true` quando usi l'API Amazon RDS.

Se non scegli di applicare le modifiche immediatamente, le modifiche vengono inserite nella coda delle modifiche in sospeso. Durante la finestra di manutenzione successiva, le eventuali modifiche in sospeso incluse nella coda vengono eseguite. Se scegli di applicare le modifiche immediatamente, verranno applicate le nuove modifiche e tutte le modifiche nella coda delle modifiche in sospeso.

> **Important**
>
> Se una qualsiasi delle modifiche in sospeso richiede che il cluster di database non sia temporaneamente disponibile (downtime), la scelta dell'opzione Applica immediatamente può causare tempi di inattività imprevisti.
> Se scegli di applicare subito una modifica, devi tener presente che saranno applicate immediatamente tutte le modifiche, invece che durante la prossima finestra di manutenzione.
> Se non vuoi che una modifica in sospeso venga applicata nella prossima finestra di manutenzione, puoi modificare l'istanza database per annullare la modifica. Puoi farlo utilizzando AWS CLI e specificando l'`--apply-immediately`opzione.

Le modifiche ad alcune impostazioni di database vengono applicate immediatamente, anche se scegli di rinviarle. Per vedere come le diverse impostazioni del database interagiscono con l'impostazione Applica immediatamente, consulta Impostazioni per la creazione di cluster di database Multi-AZ.

Impostazioni per la creazione di cluster di database Multi-AZ

Per i dettagli sulle impostazioni disponibili per modificare un cluster di database Multi-AZ, consultare la tabella seguente. Per ulteriori informazioni sulle AWS CLI opzioni, vedere modify-db-cluster. Per ulteriori informazioni sui parametri API RDS, consulta ModifyDBCluster (Modifica cluster di database).

Impostazione della console	Descrizione impostazione	Opzione CLI e parametro API di RDS	Quando avvengono le modifiche	Note sui tempi di inattività
Allocated storage (Storage allocato)	La quantità di archiviazione, in gibibyte, da allocare per ciascuna istanza database nel cluster di database. Per ulteriori informazioni, consulta Storage delle istanze di database Amazon RDS.	Opzione CLI: `--allocated-storage` Parametro API RDS: `Allocated Storage`	Se si sceglie di applicare la modifica immediatamente, questa si verifica immediatamente. Se non si sceglie di applicare la modifica immediatamente, questa si verifica durante la finestra di manutenzione successiva.	Durante questa modifica non si verifica un'interruzione.
Auto minor version upgrade (Aggiornamento automatico della versione secondaria)	Abilita l'aggiornamento automatico della versione secondaria per far sì che il cluster di database riceva automaticamente gli aggiornamenti della versione del motore di database	Opzione CLI: `--auto-minor-version-upgrade` `--no-auto-minor-version-upgrade` Parametro API RDS:	La modifica avviene immediatamente. Questa impostazione ignora l'impostazione Applica immediatamente.	Durante questa modifica non si verifica un'interruzione.

Impostazione della console	Descrizione impostazione	Opzione CLI e parametro API di RDS	Quando avvengono le modifiche	Note sui tempi di inattività
	secondaria preferita quando diventano disponibili. Amazon RDS esegue aggiornamenti automatici di versioni secondarie nella finestra di manutenzione.	`AutoMinorVersionUpgrade`		

Impostazione della console	Descrizione impostazione	Opzione CLI e parametro API di RDS	Quando avvengono le modifiche	Note sui tempi di inattività
Backup retention period (Periodo di retention dei backup)	Il numero di giorni in cui desideri eseguire il backup automatico del cluster di database da mantenere. Per un cluster di database non cruciale, impostare questo valore su **1** o su un valore maggiore. Per ulteriori informazioni, consulta Introduzione ai backup.	Opzione CLI: `--backup-retention-period` Parametro API RDS: `BackupRetentionPeriod`	Se si sceglie di applicare la modifica immediatamente, questa si verifica immediatamente. Se non si sceglie di applicare la modifica immediatamente ed è possibile modificare l'impostazione da un valore diverso da zero a un altro valore diverso da zero, la modifica viene applicata in modo asincrono, appena possibile. In caso contrario, la modifica avviene durante la finestra di manutenzione successiva.	Si verifica un'interruzione se cambi da 0 a un valore diverso da zero o da un valore diverso da zero a 0.

Impostazione della console	Descrizione impostazione	Opzione CLI e parametro API di RDS	Quando avvengono le modifiche	Note sui tempi di inattività
Backup window (Finestra di backup)	Il periodo di tempo durante il quale Amazon RDS esegue automaticamente un backup del cluster di database. A meno che non si abbiano preferenze specifiche per l'ora di esecuzione del backup del database, usare il valore predefinito No Preference (Nessuna preferenza). Per ulteriori informazioni, consulta Introduzione ai backup.	Opzione CLI: `--preferred-backup-window` Parametro API RDS: `PreferredBackupWindow`	La modifica viene applicata in modo asincrono, appena possibile.	Durante questa modifica non si verifica un'interruzione.

Impostazione della console	Descrizione impostazione	Opzione CLI e parametro API di RDS	Quando avvengono le modifiche	Note sui tempi di inattività
Autorità di certificazione	L'autorità di certificazione (CA) per il certificato del server utilizzato dal cluster DB. Per ulteriori informazioni, consulta .	Opzione CLI: `--ca-certificate-identifier` Parametro API RDS: `CACertificateIdentifier`	Se si sceglie di applicare la modifica immediatamente, questa si verifica immediatamente. Se non si sceglie di applicare la modifica immediatamente, questa si verifica durante la finestra di manutenzione successiva.	I tempi di interruzione si verificano solo se il motore database non supporta la rotazione senza riavvio. È possibile utilizzare il describe-db-engine-versions AWS CLI comando per determinare se il motore DB supporta la rotazione senza riavvio.
Copy tags to snapshot (Copia tag in snapshot	Questa opzione consente di copiare i tag del cluster di database in uno snapshot DB quando si crea uno snapshot. Per ulteriori informazioni, consulta Tagging delle risorse Amazon RDS.	Opzione CLI: `-copy-tags-to-snapshot` `-no-copy-tags-to-snapshot` Parametro API RDS: `CopyTagsToSnapshot`	La modifica avviene immediatamente. Questa impostazione ignora l'impostazione Applica immediatamente.	Durante questa modifica non si verifica un'interruzione.

Impostazione della console	Descrizione impostazione	Opzione CLI e parametro API di RDS	Quando avvengono le modifiche	Note sui tempi di inattività
Database authentication (Autenticazione del database	Password authentication (Autenticazione password) è supportato solo cluster di database Multi-AZ.	Nessuna perché l'autenticazione con password è predefinita.	Se si sceglie di applicare la modifica immediatamente, questa si verifica immediatamente. Se non si sceglie di applicare la modifica immediatamente, questa si verifica durante la finestra di manutenzione successiva.	Durante questa modifica non si verifica un'interruzione.

Impostazione della console	Descrizione impostazione	Opzione CLI e parametro API di RDS	Quando avvengono le modifiche	Note sui tempi di inattività
DB Cluster Identifier (Identificatore cluster DB)	L'identificatore del cluster di database. Questo valore è archiviato come stringa in caratteri minuscoli. Quando modifichi l'identificatore del cluster database, cambiano anche l'endpoint del cluster database e gli identificatori e gli endpoint delle istanze database nel cluster database. Il nome del nuovo cluster database deve essere univoco. La lunghezza massima è 63 caratteri. I nomi delle istanze database nel cluster database vengono	Opzione CLI: `--new-db-cluster-identifier` Parametro API RDS: `NewDBClusterIdentifier`	Se si sceglie di applicare la modifica immediatamente, questa si verifica immediatamente. Se non si sceglie di applicare la modifica immediatamente, questa si verifica durante la finestra di manutenzione successiva.	Durante questa modifica non si verifica un'interruzione.

Impostazione della console	Descrizione impostazione	Opzione CLI e parametro API di RDS	Quando avvengono le modifiche	Note sui tempi di inattività
	modificati in modo che corrispondano al nuovo nome del cluster database. Il nome di una nuova istanza database non può essere uguale al nome di un'istanza database esistente. Ad esempio, se si modifica il nome del cluster database in `maz`, è possibile che il nome dell'istanza database venga modificato in `maz-instance-1` . In questo caso, non può esistere un'istanza database denominata `maz-instance-1` . Per ulteriori informazioni, consulta			

Impostazione della console	Descrizione impostazione	Opzione CLI e parametro API di RDS	Quando avvengono le modifiche	Note sui tempi di inattività
	Assegnazione di un nuovo nome a un cluster database multi-AZ.			

Impostazione della console	Descrizione impostazione	Opzione CLI e parametro API di RDS	Quando avvengono le modifiche	Note sui tempi di inattività
Classe istanza del cluster di database	La capacità di calcolo e di memoria di ciascuna istanza database nel cluster di Multi-AZ, ad esempio `db.r6gd.xlarge` . Se possibile, scegliere una classe di istanza database sufficientemente ampia da poter tenere in memoria un tipico set di lavoro di query. Quando i set di lavoro sono conservati in memoria, il sistema può evitare di scrivere sul disco, migliorando le prestazioni. Per ulteriori informazioni, consulta the section called	Opzione CLI: `--db-cluster-instance-class` Parametro API RDS: `DBClusterInstanceClass`	Se si sceglie di applicare la modifica immediatamente, questa si verifica immediatamente. Se non si sceglie di applicare la modifica immediatamente, questa si verifica durante la finestra di manutenzione successiva.	Durante questa modifica si verifica un'interruzione.

Impostazione della console	Descrizione impostazione	Opzione CLI e parametro API di RDS	Quando avvengono le modifiche	Note sui tempi di inattività
	"Disponibilità di classi di istanze per cluster DB Multi-AZ".			
DB cluster parameter group (Gruppo di parametri del cluster database	Gruppo di parametri del cluster di database da associare al cluster di database. Per ulteriori informazioni, consulta Utilizzo di gruppi di parametri per cluster di database Multi-AZ.	Opzione CLI: `--db-cluster-parameter-group-name` Parametro API RDS: `DBClusterParameterGroupName`	La modifica del gruppo di parametri avviene immediatamente.	Durante questa modifica non si verifica un'interruzione. Quando modifichi il gruppo di parametri, le modifiche apportate ad alcuni parametri vengono applicate alle istanze database nel cluster di database Multi-AZ immediatamente senza un riavvio. Le modifiche apportate ad altri parametri vengono applicate solo dopo che le istanze database vengano riavviate.

Impostazione della console	Descrizione impostazione	Opzione CLI e parametro API di RDS	Quando avvengono le modifiche	Note sui tempi di inattività
DB engine version (Versione motore del database	Versione del motore del database da utilizzare.	Opzione CLI: `--engine-version` Parametro API RDS: `EngineVersion`	Se si sceglie di applicare la modifica immediatamente, questa si verifica immediatamente. Se non si sceglie di applicare la modifica immediatamente, questa si verifica durante la finestra di manutenzione successiva.	Durante questa modifica si verifica un'interruzione.
Deletion protection (Protezione da eliminazione)	L'opzione Enable deletion protection (Abilita protezione da eliminazione) permette di impedire l'eliminazione del cluster di database. Per ulteriori informazioni, consulta Eliminazione di un'istanza database.	Opzione CLI: `--deletion-protection` `--no-deletion-protection` Parametro API RDS: `DeletionProtection`	La modifica avviene immediatamente. Questa impostazione ignora l'impostazione Applica immediatamente.	Durante questa modifica non si verifica un'interruzione.

Impostazione della console	Descrizione impostazione	Opzione CLI e parametro API di RDS	Quando avvengono le modifiche	Note sui tempi di inattività
Maintenance window (Finestra di manutenzione)	La finestra di 30 minuti entro cui vengono applicate la modifiche in corso al cluster di database. Se il periodo di tempo non è rilevante, scegli No Preference (Nessuna preferenza). Per ulteriori informazioni, consulta Finestra di manutenzione Amazon RDS.	Opzione CLI: `--preferred-maintenance-window` Parametro API RDS: `PreferredMaintenanceWindow`	La modifica avviene immediatamente. Questa impostazione ignora l'impostazione Applica immediatamente.	Se sono presenti una o più operazioni in sospeso che causano un'interruzione e la finestra di manutenzione viene modificata per includere l'ora corrente, tali operazioni in sospeso vengono applicate immediatamente e si verifica un'interruzione.

Impostazione della console	Descrizione impostazione	Opzione CLI e parametro API di RDS	Quando avvengono le modifiche	Note sui tempi di inattività
Gestisci le credenziali principali in AWS Secrets Manager	Seleziona Manage master credentials in AWS Secrets Manager (Gestione credenziali master in AWS Secrets Manager) per gestire la password dell'utente master in un segreto di Secrets Manager. Facoltativamente, scegli la chiave KMS da utilizzare per proteggere il segreto. Scegliere tra le chiavi KMS presenti nell'account o inserire la chiave da un altro account. Se RDS sta già gestendo la password dell'utente master per il cluster database,	Opzione CLI: `--manage-master-user-password | --no-manage-master-user-password` `--master-user-secret-kms-key-id` `--rotate-master-user-password | --no-rotate-master-user-password` Parametro API RDS: `ManageMasterUserPassword` `MasterUserSecretKmsKeyId` `RotateMasterUserPassword`	Quando attivi o disattivi la gestione automatica delle password dell'utente master, la modifica viene applicata immediatamente. Questa modifica ignora l'impostazione Apply immediately (Applica immediatamente). Quando ruoti la password dell'utente master, è necessario specificare che la modifica venga applicata immediatamente.	Durante questa modifica non si verifica un'interruzione.

Impostazione della console	Descrizione impostazione	Opzione CLI e parametro API di RDS	Quando avvengono le modifiche	Note sui tempi di inattività
	puoi ruotare la password dell'utente master scegliendo Rotate secret immediately (Ruota il segreto immediatamente). Per ulteriori informazioni, consulta Gestione delle password con Amazon RDS e AWS Secrets Manager.			
New master passworc (Nuova passworc master)	La password dell'account utente master.	Opzione CLI: `--master-user-password` Parametro API RDS: `MasterUserPassword`	La modifica viene applicata in modo asincrono, appena possibile. Questa impostazione ignora l'impostazione Applica immediatamente.	Durante questa modifica non si verifica un'interruzione.

Impostazione della console	Descrizione impostazione	Opzione CLI e parametro API di RDS	Quando avvengono le modifiche	Note sui tempi di inattività
IOPS con provisioning	Quantità di IOPS con provisioning (operazioni di input/output al secondo) da allocare inizialmente al cluster di database.	Opzione CLI: `--iops` Parametro API RDS: `Iops`	Se si sceglie di applicare la modifica immediatamente, questa si verifica immediatamente. Se non si sceglie di applicare la modifica immediatamente, questa si verifica durante la finestra di manutenzione successiva.	Durante questa modifica non si verifica un'interruzione.

Impostazione della console	Descrizione impostazione	Opzione CLI e parametro API di RDS	Quando avvengono le modifiche	Note sui tempi di inattività
Accesso pubblico	Publicly accessible (Accessibile pubblicamente) per assegnare al cluster di database un indirizzo IP pubblico, ovvero renderlo accessibile al di fuori del cloud privato virtuale (VPC). Per essere accessibile pubblicamente, il cluster di database deve anche trovarsi in una sottorete pubblica nel VPC. Not publicly accessible (Non accessibile pubblicamente) per rendere il cluster di database accessibile solo dall'interno del VPC.	Non disponibile durante la modifica di un cluster di database.	La modifica avviene immediatamente. Questa impostazione ignora l'impostazione Applica immediatamente.	Durante questa modifica non si verifica un'interruzione.

Impostazione della console	Descrizione impostazione	Opzione CLI e parametro API di RDS	Quando avvengono le modifiche	Note sui tempi di inattività
	Per ulteriori informazioni, consulta Nascondere istanze database in un VPC da Internet. Per connettersi a un cluster di database dall'esterno del proprio VPC, il cluster di database deve essere accessibile pubblicamente. Inoltre, l'accesso deve essere concesso utilizzando le regole in ingresso del gruppo di sicurezza del cluster di database e devono essere soddisfatti altri requisiti. Per ulteriori informazioni, consulta Impossibile connetter			

Impostazione della console	Descrizione impostazione	Opzione CLI e parametro API di RDS	Quando avvengono le modifiche	Note sui tempi di inattività
	si all'istanza database di Amazon RDS. Se il tuo cluster DB non è accessibile pubblicamente, puoi utilizzare una connessione AWS VPN da sito a sito o AWS Direct Connect una connessione per accedervi da una rete privata. Per ulteriori informazioni, consulta Riservatezza del traffico Internet.			

Impostazione della console	Descrizione impostazione	Opzione CLI e parametro API di RDS	Quando avvengono le modifiche	Note sui tempi di inattività
Storage Type (Tipo di storage)	Il tipo di archiviazione per il cluster di database. Sono supportati solo gli storage General Purpose SSD (gp3), Provisioned IOPS (io1) e Provisioned IOPS SSD (io2). Per ulteriori informazioni, consulta Tipi di storage Amazon RDS.	Opzione CLI: `--storage-type` Parametro API RDS: `StorageType`	Se si sceglie di applicare la modifica immediatamente, questa si verifica immediatamente. Se non si sceglie di applicare la modifica immediatamente, questa si verifica durante la finestra di manutenzione successiva.	Durante questa modifica non si verifica un'interruzione.
Gruppo di sicurezza VPC	I gruppi di sicurezza da associare al cluster di database. Per ulteriori informazioni, consulta Panoramica dei gruppi di sicurezza VPC.	Opzione CLI: `--vpc-security-group-ids` Parametro API RDS: `VpcSecurityGroupIds`	La modifica viene applicata in modo asincrono, appena possibile. Questa impostazione ignora l'impostazione Applica immediatamente.	Durante questa modifica non si verifica un'interruzione.

Impostazioni che non si applicano durante la modifica di cluster di database Multi-AZ

Le seguenti impostazioni nel AWS CLI comando modify-db-clustere nell'operazione dell'API RDS ModifyDBCluster non si applicano ai cluster DB Multi-AZ.

Inoltre, non è possibile specificare queste impostazioni per i cluster di database Multi-AZ nella console.

AWS CLI impostazione	Impostazione API RDS
`--backtrack-window`	`BacktrackWindow`
`--cloudwatch-logs-export-configuration`	`CloudwatchLogsExportConfiguration`
`--copy-tags-to-snapshot | --no-copy-tags-to-snapshot`	`CopyTagsToSnapshot`
`--db-instance-parameter-group-name`	`DBInstanceParameterGroupName`
`--domain`	`Domain`
`--domain-iam-role-name`	`DomainIAMRoleName`
`--enable-global-write-forwarding | --no-enable-global-write-forwarding`	`EnableGlobalWriteForwarding`
`--enable-http-endpoint | --no-enable-http-endpoint`	`EnableHttpEndpoint`
`--enable-iam-database-authentication | --no-enable-iam-database-authentication`	`EnableIAMDatabaseAuthentication`
`--option-group-name`	`OptionGroupName`
`--port`	`Port`

AWS CLI impostazione	Impostazione API RDS
`--scaling-configuration`	`ScalingConfiguration`
`--storage-type`	`StorageType`

Assegnazione di un nuovo nome a un cluster database multi-AZ

Puoi rinominare un cluster database multi-AZ utilizzando la AWS Management Console, il comando AWS CLI `modify-db-cluster` o l'operazione `ModifyDBCluster` dell'API Amazon RDS. L'assegnazione di un nuovo nome a un cluster database multi-AZ può avere effetti significativi. Di seguito è riportato un elenco di elementi da tenere in considerazione prima di rinominare un cluster database multi-AZ.

- Quando assegni un nuovo nome a un cluster database multi-AZ, gli endpoint per il cluster database multi-AZ cambiano. Questi endpoint vengono modificati per includere il nome assegnato al cluster database multi-AZ. Puoi reindirizzare il traffico da un vecchio endpoint a uno nuovo. Per ulteriori informazioni sugli endpoint dei cluster database multi-AZ, consulta Connessione a un cluster di database multi-AZ.
- Quando si rinomina un cluster database multi-AZ, il nome DNS usato in precedenza dal cluster database multi-AZ viene eliminato immediatamente, ma può rimanere memorizzato nella cache per alcuni minuti. Il nuovo nome DNS del cluster database multi-AZ assegnato diventa effettivo dopo circa due minuti. Il cluster database multi-AZ rinominato non è disponibile fino a quando il nuovo nome non diventa effettivo.
- Non è possibile utilizzare un nome di cluster database multi-AZ esistente per rinominare un cluster.
- Se riutilizzi un nome di cluster database multi-AZ, le metriche e gli eventi associati a tale nome vengono mantenuti.
- I tag del cluster database multi-AZ rimangono nel cluster database multi-AZ, indipendentemente dall'assegnazione del nuovo nome.
- Gli snapshot del cluster database vengono mantenuti per un cluster database multi-AZ rinominato.

> **Note**
>
> Un cluster database multi-AZ è un ambiente di database isolato in esecuzione nel cloud. Un cluster database multi-AZ può ospitare più database. Per informazioni sulla modifica del nome di un database, consulta la documentazione relativa al motore database.

Assegnazione di un nuovo nome per la sostituzione di un cluster database multi-AZ

Gli scenari più comuni per la ridenominazione di un cluster DB Multi-AZ includono il ripristino dei dati da un'istantanea del cluster DB o l'esecuzione del point-in-time ripristino (PITR). Rinominando il

cluster database multi-AZ, è possibile sostituirlo senza modificare alcun codice dell'applicazione che faccia riferimento al cluster database multi-AZ. In questi casi, completa i passaggi seguenti:

1. Arresta tutto il traffico diretto al cluster database multi-AZ. Puoi reindirizzare il traffico che accede ai database nel cluster database multi-AZ o scegliere un altro modo per impedire che il traffico acceda ai database nel cluster database multi-AZ.
2. Assegna un nuovo nome al cluster database multi-AZ.
3. Crea un nuovo cluster database multi-AZ eseguendo il ripristino da uno snapshot database o il ripristino point-in-time. Assegna quindi al nuovo cluster database multi-AZ il nome del precedente cluster database multi-AZ.

Se si elimina il vecchio cluster database multi-AZ, vengono eliminati tutti gli snapshot di database indesiderati del vecchio cluster database multi-AZ.

Console

Per rinominare un cluster database multi-AZ

1. Accedi alla AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel pannello di navigazione, scegliere Databases (Database).
3. Seleziona il cluster database multi-AZ che desideri rinominare.
4. Scegli Modifica.
5. In Settings (Impostazioni) immetti un nuovo nome per DB cluster identifier (Identificatore cluster database).
6. Scegli Continue (Continua).
7. Per applicare immediatamente le modifiche, scegliere Apply immediately (Applica immediatamente). In alcuni casi, la chiusura di questa opzione può causare un'interruzione. Per ulteriori informazioni, consulta Applicazione immediata delle modifiche).
8. Nella pagina di conferma esaminare le modifiche. Se sono corrette, selezionare Modify cluster (Modifica cluster) per salvare le modifiche.

   In alternativa, scegli Back (Indietro) per cambiare le modifiche o Cancel (Annulla) per annullare le modifiche.

AWS CLI

Per rinominare un cluster DB Multi-AZ, utilizzare il comando. AWS CLI modify-db-cluster Fornisci il valore corrente di `--db-cluster-identifier` e il parametro `--new-db-cluster-identifier` con il nuovo nome del cluster database multi-AZ.

Example

PerLinux, omacOS: Unix

```
aws rds modify-db-cluster \
    --db-cluster-identifier DBClusterIdentifier \
    --new-db-cluster-identifier NewDBClusterIdentifier
```

Per Windows:

```
aws rds modify-db-cluster ^
    --db-cluster-identifier DBClusterIdentifier ^
    --new-db-cluster-identifier NewDBClusterIdentifier
```

API RDS

Per ripristinare un cluster database multi-AZ chiama l'operazione API Amazon RDS ModifyDBCluster con i seguenti parametri:

- `DBClusterIdentifier`: il nome del cluster database esistente.
- `NewDBClusterIdentifier`: il nome del nuovo cluster database.

Riavvio di un cluster di database multi-AZ e istanze database di lettura

Potrebbe essere necessario riavviare il cluster di database Multi-AZ, in genere per motivi di manutenzione. Se, ad esempio, vengono apportate determinate modifiche oppure se viene modificato il gruppo di parametri cluster di database associato al cluster di database, riavviare il cluster di database. In questo modo, le modifiche saranno effettive.

Se il cluster di database non usa le modifiche più recenti apportate al gruppo di parametri del cluster di database associato, la AWS Management Console mostra il gruppo di parametri del cluster di database con stato pending-reboot (riavvio in attesa). Lo stato dei gruppi di parametro pending-reboot non prevede un riavvio automatico nel corso della prossima finestra di manutenzione. Per applicare le ultime modifiche del parametro su quel cluster di database, riavviare manualmente il cluster di database. Per ulteriori informazioni sui gruppi di parametri, consultare Utilizzo di gruppi di parametri per cluster di database Multi-AZ.

Il riavvio di un cluster di database comporta il riavvio del servizio del motore di database. Il riavvio di un cluster di database comporta un'interruzione temporanea, durante la quale lo stato del cluster di database viene impostato su rebooting (riavvio in corso).

Non puoi riavviare il cluster di database se il suo stato non è disponibile. Il database può non essere disponibile per diversi motivi, ad esempio un backup in corso, una modifica richiesta in precedenza o un'operazione della finestra di manutenzione.

Il tempo necessario per riavviare il cluster di database dipende dal processo di ripristino dell'arresto anomalo, l'attività del database al momento del riavvio e il comportamento del cluster di database specifico. Per ottimizzare i tempi di riavvio, è consigliabile ridurre l'attività del database il più possibile durante il processo di riavvio. Riducendo l'attività del database si riduce l'attività di rollback per le transazioni in transito.

> **Important**
>
> I cluster di database multi-AZ non supportano il riavvio con un failover. Quando si riavvia l'istanza di scrittura di un cluster di database Multi-AZ, ciò non influisce sulle istanze database del lettore in quel cluster di database e non si verifica alcun failover. Quando riavvii un'istanza database di lettore, non si verifica alcun failover. Per eseguire il failover di un cluster di database Multi-AZ, scegliere Failover nella console, richiamare il comando AWS CLI `failover-db-cluster` o richiamare l'operazione API `FailoverDBCluster`.

Console

Per riavviare un cluster di database

1. Accedi alla AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel pannello di navigazione, scegliere Databases (Database), quindi scegliere il cluster di database Multi-AZ che si desidera riavviare.
3. In Actions (Operazioni), scegliere Reboot (Riavvia).

   Viene visualizzata la pagina Riavvia cluster di database.
4. Per riavviare il cluster di database, scegliere Reboot (Riavvia).

   Oppure scegliere Cancel (Annulla).

AWS CLI

Per riavviare un cluster di database Multi-AZ tramite AWS CLI, chiamare il comando reboot-db-cluster.

```
aws rds reboot-db-cluster --db-cluster-identifier mymultiazdbcluster
```

API RDS

Per riavviare un cluster di database Multi-AZ utilizzando l'API Amazon RDS, chiamare l'operazioneRebootDBCluster.

Utilizzo delle repliche di lettura del cluster di database multi-AZ

Una replica di lettura del cluster di database è un tipo speciale di cluster che viene creato da un'istanza database di origine. Dopo aver creato una replica di lettura, gli aggiornamenti applicati all'istanza database primaria vengono copiati in modo asincrono nella replica di lettura del cluster di database Multi-AZ. Puoi ridurre il carico sull'istanza database di database primaria instradando le query di lettura dalle applicazioni alla replica di lettura. Tramite le repliche di lettura puoi dimensionare in modo elastico la capacità oltre i vincoli di una singola istanza database per carichi di lavoro di database particolarmente gravosi in lettura.

Puoi anche creare una o più repliche di lettura dell'istanza database da un cluster database Multi-AZ. Le repliche di lettura delle istanze database consentono di superare la capacità di calcolo o di I/O del cluster database Multi-AZ di origine indirizzando il traffico di lettura in eccesso verso le repliche di lettura. Al momento, non è possibile creare una replica di lettura del cluster database Multi-AZ da un cluster database Multi-AZ esistente.

Argomenti

Migrazione a un cluster database multi-AZ tramite una replica di lettura

Per eseguire la migrazione di un'implementazione single-AZ o di un'implementazione di istanza database multi-AZ a un'implementazione di cluster database multi-AZ con tempi di inattività ridotti, è possibile creare una replica di lettura del cluster database multi-AZ. Per l'origine, si specifica l'istanza database nell'implementazione single-AZ o l'istanza database primaria nell'implementazione di istanza database multi-AZ. L'istanza database può elaborare le transazioni di scrittura durante la migrazione a un cluster database multi-AZ.

Di seguito sono indicati i requisiti da considerare prima di creare la replica di lettura del cluster di database multi-AZ:

- La versione dell'istanza database di origine deve supportare il cluster database multi-AZ. Per ulteriori informazioni, consulta Regioni e motori DB supportati per cluster DB Multi-AZ in Amazon RDS.
- La replica di lettura del cluster database multi-AZ deve avere la stessa versione principale dell'origine e la stessa versione secondaria o successiva.

- Attiva i backup automatici nell'istanza database di origine impostando il periodo di conservazione dei backup su un valore diverso da zero.
- Lo spazio di archiviazione allocato dell'istanza database di origine deve essere pari o superiore a 100 GiB.
- Per RDS per MySQL, i parametri `gtid-mode` e `enforce_gtid_consistency` devono entrambi essere impostati su `ON` per l'istanza database di origine. È necessario utilizzare un gruppo di parametri personalizzati e non il gruppo parametri predefiniti. Per ulteriori informazioni, consulta the section called “Utilizzo di gruppi di parametri di database”.
- Una transazione attiva a esecuzione prolungata può rallentare il processo di creazione della replica di lettura. Ti consigliamo di attendere il completamento delle transazioni a esecuzione prolungata prima di creare una replica di lettura.
- Se elimini l'istanza database di origine per una replica di lettura del cluster database multi-AZ, la replica di lettura viene promossa a cluster database multi-AZ autonomo.

Creazione e promozione della replica di lettura del cluster database multi-AZ

È possibile creare e promuovere una replica di lettura del cluster DB Multi-AZ utilizzando AWS Management Console AWS CLI, o l'API RDS.

> **Note**
>
> Ti consigliamo vivamente di creare tutte le repliche di lettura nello stesso cloud privato virtuale (VPC) utilizzando Amazon VPC come istanza database di origine.
> Se crei una replica di lettura in un VPC diverso dall'istanza DB di origine, gli intervalli CIDR (Classless Inter-Domain Routing) possono sovrapporsi tra la replica e il sistema Amazon RDS. La sovrapposizione CIDR rende la replica instabile, influendo negativamente sulle applicazioni che si connettono. Se viene visualizzato un errore durante la creazione della replica di lettura, scegli un gruppo di sottoreti DB di destinazione diverso. Per ulteriori informazioni, consulta Uso di un'istanza database in un VPC.

Console

Per eseguire la migrazione di un'implementazione single-AZ o di un'implementazione di istanza database multi-AZ a un cluster database multi-AZ tramite una replica di lettura, completa i seguenti passaggi utilizzando la AWS Management Console.

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Crea la replica di lettura del cluster database multi-AZ.

    a. Nel riquadro di navigazione, scegliere Databases (Database).

    b. Scegli l'istanza database da usare come origine per la replica di lettura.

    c. Per Actions (Operazioni), scegliere Create read replica (Crea replica di lettura).

    d. Per Availability and durability (Disponibilità e durabilità), scegli Multi-AZ database cluster (Cluster di database multi-AZ).

    e. Per DB instance identifier (Identificatore istanze DB) inserire un nome per la replica di lettura.

    f. Per le restanti sezioni, specifica le impostazioni del cluster di database. Per informazioni sull'impostazione, consulta Impostazioni per la creazione di cluster di database Multi-AZ.

    g. Scegli Create read replica (Crea replica di lettura).
3. A questo punto, è possibile promuovere la replica di lettura a cluster database multi-AZ autonomo:

    a. Arresta la scrittura delle transazioni nell'istanza database di origine e quindi attendi l'applicazione di tutti gli aggiornamenti alla replica di lettura.

    Gli aggiornamenti del database vengono eseguiti nella replica di lettura dopo essere stati completati nell'istanza database primaria. Questo ritardo della replica può variare in modo significativo. Utilizzare il parametro `ReplicaLag` per determinare quando sono stati applicati tutti gli aggiornamenti alla replica di lettura. Per ulteriori informazioni sul ritardo della replica, consulta Monitoraggio della replica di lettura.

    b. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.

    c. Nella console Amazon RDS scegliere Databases (Database).

    Verrà visualizzato il riquadro Databases (Database). Ogni replica di lettura mostra la Replica nella colonna Role (Ruolo).

    d. Scegli la replica di lettura del cluster database multi-AZ che desideri promuovere.

    e. In Actions (Operazioni), seleziona Promote (Promuovi).

f. Nella pagina Promote read replica (Promuovi replica di lettura) immetti il periodo di conservazione dei backup e la finestra di backup per il nuovo cluster database multi-AZ promosso.

g. Dopo aver definito tutte le impostazioni desiderate, scegli Promote read replica (Promuovi replica di lettura).

h. Attendi che lo stato del cluster database multi-AZ promosso diventi `Available`.

i. Configura le applicazioni in modo che utilizzino il cluster database multi-AZ promosso.

Facoltativamente, elimina l'implementazione single-AZ o l'implementazione di istanza database multi-AZ, se non è più necessaria. Per istruzioni, consulta Eliminazione di un'istanza database.

AWS CLI

Per eseguire la migrazione di un'implementazione single-AZ o di un'implementazione di istanza database multi-AZ a un cluster database multi-AZ tramite una replica di lettura, completa i seguenti passaggi utilizzando la AWS CLI.

1. Crea la replica di lettura del cluster database multi-AZ.

   Per creare una replica di lettura dall'istanza DB di origine, usa il AWS CLI comando. `create-db-cluster` Per `--replication-source-identifier`, specifica il nome della risorsa Amazon (ARN) dell'istanza database di origine.

   Per LinuxmacOS, oUnix:

```
aws rds create-db-cluster \
  --db-cluster-identifier mymultiazdbcluster \
  --replication-source-identifier arn:aws:rds:us-
east-2:123456789012:db:mydbinstance
  --engine postgres \
  --db-cluster-instance-class db.m5d.large \
  --storage-type io1 \
  --iops 1000 \
  --db-subnet-group-name defaultvpc \
  --backup-retention-period 1
```

   Per Windows:

```
aws rds create-db-cluster ^
  --db-cluster-identifier mymultiazdbcluster ^
  --replication-source-identifier arn:aws:rds:us-
east-2:123456789012:db:mydbinstance
  --engine postgres ^
  --db-cluster-instance-class db.m5d.large ^
  --storage-type io1 ^
  --iops 1000 ^
  --db-subnet-group-name defaultvpc ^
  --backup-retention-period 1
```

2. Arresta la scrittura delle transazioni nell'istanza database di origine e quindi attendi l'applicazione di tutti gli aggiornamenti alla replica di lettura.

   Gli aggiornamenti del database vengono eseguiti nella replica di lettura dopo essere stati completati nell'istanza database primaria. Questo ritardo della replica può variare in modo significativo. Utilizzare il parametro `Replica Lag` per determinare quando sono stati applicati tutti gli aggiornamenti alla replica di lettura. Per ulteriori informazioni sul ritardo della replica, consulta Monitoraggio della replica di lettura.
3. A questo punto, è possibile promuovere la replica di lettura a cluster database multi-AZ autonomo.

   Per promuovere una replica di lettura del cluster database multi-AZ, utilizza il comando AWS CLI `promote-read-replica-db-cluster`. Per `--db-cluster-identifier`, specifica l'identificatore della replica di lettura del cluster database multi-AZ.

   ```
   aws rds promote-read-replica-db-cluster --db-cluster-identifier mymultiazdbcluster
   ```
4. Attendi che lo stato del cluster database multi-AZ promosso diventi `Available`.
5. Configura le applicazioni in modo che utilizzino il cluster database multi-AZ promosso.

Facoltativamente, elimina l'implementazione single-AZ o l'implementazione di istanza database multi-AZ, se non è più necessaria. Per istruzioni, consulta Eliminazione di un'istanza database.

API RDS

Per eseguire la migrazione di un'implementazione single-AZ o di un'implementazione di istanza database multi-AZ a un cluster database multi-AZ tramite una replica di lettura, completa i seguenti passaggi utilizzando l'API RDS.

1. Crea la replica di lettura del cluster database multi-AZ.

   Per creare una replica di lettura del cluster database multi-AZ, utilizza l'operazione `CreateDBCluster` con il parametro `DBClusterIdentifier` richiesto. Per `ReplicationSourceIdentifier`, specifica il nome della risorsa Amazon (ARN) dell'istanza database di origine.
2. Arresta la scrittura delle transazioni nell'istanza database di origine e quindi attendi l'applicazione di tutti gli aggiornamenti alla replica di lettura.

   Gli aggiornamenti del database vengono eseguiti nella replica di lettura dopo essere stati completati nell'istanza database primaria. Questo ritardo della replica può variare in modo significativo. Utilizzare il parametro `Replica Lag` per determinare quando sono stati applicati tutti gli aggiornamenti alla replica di lettura. Per ulteriori informazioni sul ritardo della replica, consulta Monitoraggio della replica di lettura.
3. A questo punto, è possibile promuovere la replica di lettura a cluster database multi-AZ autonomo.

   Per promuovere una replica di lettura del cluster database multi-AZ, utilizza l'operazione `PromoteReadReplicaDBCluster` con il parametro `DBClusterIdentifier` richiesto. Specifica l'identificatore della replica di lettura del cluster database multi-AZ.
4. Attendi che lo stato del cluster database multi-AZ promosso diventi `Available`.
5. Configura le applicazioni in modo che utilizzino il cluster database multi-AZ promosso.

Facoltativamente, elimina l'implementazione single-AZ o l'implementazione di istanza database multi-AZ, se non è più necessaria. Per istruzioni, consulta Eliminazione di un'istanza database.

Limitazioni alla creazione di una replica di lettura del cluster database multi-AZ

Le seguenti limitazioni si applicano alla creazione di una replica di lettura del cluster database multi-AZ da un'implementazione di istanza database single-AZ o multi-AZ.

- Non è possibile creare una replica di lettura del cluster DB Multi-AZ in un Account AWS ambiente diverso da quello Account AWS che possiede l'istanza DB di origine.
- Non è possibile creare una replica di lettura del cluster DB Multi-AZ in un'istanza DB diversa Regione AWS da quella di origine.
- Non è possibile eseguire il ripristino point-in-time di una replica di lettura del cluster database multi-AZ.

- La crittografia di archiviazione deve avere le stesse impostazioni per l'istanza database di origine e il cluster database multi-AZ.
- Se l'istanza database di origine è crittografata, la replica di lettura del cluster database multi-AZ deve essere crittografata utilizzando la stessa chiave KMS.
- Se l'istanza DB di origine utilizza lo storage General Purpose SSD (gp3) e dispone di meno di 400 GiB di storage allocato, non è possibile modificare gli IOPS assegnati per la replica di lettura del cluster DB Multi-AZ.
- Per eseguire un aggiornamento della versione secondaria nell'istanza database di origine, è necessario innanzitutto eseguire l'aggiornamento della versione secondaria nella replica di lettura del cluster database multi-AZ.
- Quando si esegue un aggiornamento di versione minore su una replica di lettura del cluster DB RDS for PostgreSQL Multi-AZ, l'istanza DB reader non passa all'istanza DB writer dopo l'aggiornamento. Pertanto, il tuo cluster DB potrebbe subire tempi di inattività durante l'aggiornamento dell'istanza writer da parte di Amazon RDS.
- Non è possibile eseguire un aggiornamento di versione principale su una replica di lettura del cluster DB Multi-AZ.
- È possibile eseguire un aggiornamento della versione principale nell'istanza database di origine di una replica di lettura del cluster database multi-AZ, ma l'esecuzione della replica di lettura si arresta e non può essere riavviata.
- La replica di lettura del cluster database multi-AZ non supporta le repliche di lettura a cascata.
- In RDS per PostgreSQL, le repliche di lettura del cluster database multi-AZ non possono eseguire il failover.

Creazione di una replica di lettura di un'istanza database da un cluster database Multi-AZ

Puoi creare una replica di lettura dell'istanza database da un cluster database Multi-AZ per superare la capacità di calcolo o di I/O del cluster per i carichi di lavoro di database con un uso intensivo delle operazioni di lettura. Puoi indirizzare questo traffico in lettura in eccesso a una o più repliche di lettura dell'istanza database. Puoi anche utilizzare le repliche di lettura per eseguire la migrazione da un cluster database Multi-AZ a un'istanza database.

Per creare una replica di lettura, specifica un cluster database Multi-AZ come origine della replica. Una delle istanze di lettura del cluster database Multi-AZ è sempre l'origine della replica e non

l'istanza di scrittura. Questa condizione garantisce che la replica sia sempre sincronizzata con il cluster di origine, anche in caso di failover.

Argomenti

Confronto tra istanza database di lettura e repliche di lettura dell'istanza database

Una replica di lettura di un'istanza database di un cluster database Multi-AZ è diversa dalle istanze database di lettura del cluster database Multi-AZ per i seguenti motivi:

- A differenza delle repliche di lettura dell'istanza database, le istanze database di lettura fungono da destinazioni del failover automatico.
- Le istanze database di lettura devono confermare una modifica proveniente dall'istanza database di scrittura prima che venga eseguito il commit di tale modifica. Per le repliche di lettura dell'istanza database, tuttavia, gli aggiornamenti vengono copiati in modo asincrono nella replica di lettura senza richiedere la conferma.
- Le istanze database di lettura condividono sempre la stessa classe di istanza, lo stesso tipo di archiviazione e la stessa versione di motore dell'istanza database di scrittura del cluster database Multi-AZ. Le repliche di lettura delle istanze database, tuttavia, non devono necessariamente condividere le stesse configurazioni del cluster di origine.
- Puoi promuovere una replica di lettura dell'istanza database a istanza database autonoma. Non è possibile promuovere un'istanza database di lettura di un cluster database Multi-AZ a istanza autonoma.
- L'endpoint di lettura indirizza solo le istanze database di lettura del cluster database Multi-AZ. Non indirizza mai le richieste a una replica di lettura dell'istanza database.

Per ulteriori informazioni sulle istanze database di lettura e scrittura, consulta the section called “Panoramica dei cluster di database Multi-AZ”.

Considerazioni

Di seguito sono indicati i requisiti da considerare prima di creare una replica di lettura dell'istanza database da un cluster database Multi-AZ:

- Quando crei la replica di lettura dell'istanza database, tale replica deve avere la stessa versione principale del cluster di origine e la stessa versione secondaria o successiva. Dopo averlo creato, puoi facoltativamente aggiornare la replica di lettura a una versione secondaria successiva rispetto a quella usata dal cluster di origine.
- Quando crei la replica di lettura dell'istanza database, lo spazio di archiviazione allocato deve essere uguale allo spazio di archiviazione allocato del cluster database Multi-AZ di origine. Puoi modificare lo spazio di archiviazione allocato dopo aver creato la replica di lettura.
- Per RDS per MySQL, il parametro `gtid-mode` deve essere impostato su `ON` per il cluster database Multi-AZ di origine. Per ulteriori informazioni, consulta the section called “Utilizzo di gruppi di parametri di cluster di database”.
- Una transazione attiva a esecuzione prolungata può rallentare il processo di creazione della replica di lettura. Ti consigliamo di attendere il completamento delle transazioni a esecuzione prolungata prima di creare una replica di lettura.
- Se elimini il cluster database Multi-AZ di origine per una replica di lettura dell'istanza database, tutte le repliche di lettura su cui sta scrivendo vengono promosse a istanze database autonome.

Creazione di una replica di lettura dell'istanza database

È possibile creare una replica di lettura di un'istanza DB da un cluster DB Multi-AZ utilizzando AWS Management Console AWS CLI, o l'API RDS.

> **Note**
>
> Ti consigliamo vivamente di creare tutte le repliche di lettura nello stesso cloud privato virtuale (VPC) utilizzando Amazon VPC come cluster database Multi-AZ di origine.
> Se crei una replica di lettura in un VPC diverso da quello del cluster database Multi-AZ di origine, gli intervalli di routing interdominio senza classi (CIDR) possono sovrapporsi tra la replica e il sistema RDS. La sovrapposizione CIDR rende la replica instabile, influendo negativamente sulle applicazioni che si connettono. Se viene visualizzato un errore durante la creazione della replica di lettura, scegli un gruppo di sottoreti DB di destinazione diverso. Per ulteriori informazioni, consulta the section called “Uso di un'istanza database in un VPC”.

Console

Per creare una replica di lettura di un'istanza database da un cluster database Multi-AZ, completa i seguenti passaggi utilizzando la AWS Management Console.

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, scegli Databases (Database).
3. Scegli il cluster database Multi-AZ da usare come origine della replica di lettura.
4. Per Actions (Operazioni), scegliere Create read replica (Crea replica di lettura).
5. In Origine replica, verifica che sia selezionato il cluster DB Multi-AZ corretto.
6. In Identificatore DB, specifica il nome della replica di lettura.
7. Per le restanti sezioni, specifica le impostazioni dell'istanza database. Per informazioni sull'impostazione, consulta the section called “Impostazioni disponibili”.

   > **Note**
   >
   > Lo spazio di archiviazione allocato per la replica di lettura dell'istanza database deve essere uguale allo spazio di archiviazione allocato per il cluster database Multi-AZ di origine.

8. Scegli Create read replica (Crea replica di lettura).

AWS CLI

Per creare una replica di lettura di un'istanza DB da un cluster DB Multi-AZ, usa il comando. AWS CLI `create-db-instance-read-replica` Per `--source-db-cluster-identifier`, specifica l'identificatore del cluster database Multi-AZ.

PerLinux, omacOS: Unix

```
aws rds create-db-instance-read-replica \
  --db-instance-identifier myreadreplica \
  --source-db-cluster-identifier mymultiazdbcluster
```

Per Windows:

```
aws rds create-db-instance-read-replica ^
```

```
  --db-instance-identifier myreadreplica ^
  --source-db-cluster-identifier mymultiazdbcluster
```

API RDS

Per creare una replica di lettura di un'istanza database da un cluster database Multi-AZ, utilizza l'operazione `CreateDBInstanceReadReplica`.

Promozione della replica di lettura dell'istanza database

Se non hai più bisogno della replica di lettura dell'istanza database, puoi promuoverla in un'istanza database autonoma. Quando promuovi una replica di lettura, l'istanza database viene riavviata prima di diventare disponibile. Per istruzioni, consulta the section called “Promozione di una replica di lettura”.

Se utilizzi la replica di lettura per eseguire la migrazione di un'implementazione Multi-AZ di cluster database a un'implementazione di istanza database Single-AZ o Multi-AZ, assicurati di interrompere tutte le transazioni in fase di scrittura nel cluster database di origine. Attendi quindi il completamento di tutti gli aggiornamenti nella replica di lettura. Gli aggiornamenti del database vengono eseguiti nella replica di lettura dopo essere stati completati in una delle istanze database di lettura del cluster database Multi-AZ. Questo ritardo della replica può variare in modo significativo. Utilizzare il parametro `ReplicaLag` per determinare quando sono stati applicati tutti gli aggiornamenti alla replica di lettura. Per ulteriori informazioni sul ritardo della replica, consulta the section called “Monitoraggio della replica di lettura”.

Dopo aver promosso la replica di lettura, attendi che lo stato dell'istanza database promossa indichi `Available` prima di impostare le applicazioni per l'uso dell'istanza database promossa. Facoltativamente, elimina l'implementazione Multi-AZ del cluster database se non ne hai più bisogno. Per istruzioni, consulta the section called “Per eliminare un cluster di database Multi-AZ”.

Limitazioni alla creazione di una replica di lettura di un'istanza database da un cluster database Multi-AZ

Le seguenti limitazioni si applicano alla creazione di una replica di lettura di un'istanza database da un'implementazione Multi-AZ di un cluster database.

- Non è possibile creare una replica di lettura di un'istanza DB in un cluster DB diverso da Account AWS quello Account AWS che possiede il cluster DB Multi-AZ di origine.
- Non è possibile creare una replica di lettura di un'istanza DB in un cluster DB Multi-AZ diverso Regione AWS da quello di origine.

- Non puoi eseguire il ripristino point-in-time di una replica di lettura di un'istanza database.
- La crittografia di archiviazione deve avere le stesse impostazioni per il cluster database Multi-AZ di origine e la replica di lettura dell'istanza database.
- Se il cluster database Multi-AZ di origine è crittografata, la replica di lettura dell'istanza database deve essere crittografata utilizzando la stessa chiave KMS.
- Per eseguire un aggiornamento della versione secondaria nel cluster database Multi-AZ di origine, è innanzitutto necessario eseguire l'aggiornamento della versione secondaria nella replica di lettura dell'istanza database.
- La replica di lettura dell'istanza database non supporta le repliche di lettura a cascata.
- Per RDS per PostgreSQL, il cluster database Multi-AZ di origine deve eseguire PostgreSQL versione 13.11, 14.8 o 15.2.R2 o successive per creare una replica di lettura dell'istanza database.
- È possibile eseguire un aggiornamento della versione principale nel cluster database Multi-AZ di origine di una replica di lettura dell'istanza database, ma l'esecuzione della replica di lettura si arresta e non può essere riavviata.

Utilizzo della replica logica di PostgreSQL con cluster database multi-AZ

Utilizzando la funzionalità di replica logica di PostgreSQL con il cluster database multi-AZ, puoi replicare e sincronizzare singole tabelle anziché l'intera istanza database. La replica logica utilizza un modello di pubblicazione e sottoscrizione per replicare le modifiche da un'origine in uno o più destinatari. Funziona utilizzando i record di modifica del WAL (write-ahead log) PostgreSQL. Per ulteriori informazioni, consulta the section called “Replica logica”.

Quando crei un nuovo slot di replica logica sull'istanza database di scrittura di un cluster database multi-AZ, lo slot viene copiato in modo asincrono su ogni istanza database di lettura nel cluster. Gli slot delle istanze database di lettura vengono continuamente sincronizzati con quelli dell'istanza database di scrittura.

La replica logica è supportata per i cluster database multi-AZ che eseguono RDS per PostgreSQL versione 14.8-R2 e versioni successive e 15.3-R2 e versioni successive.

> **Note**
>
> Oltre alla funzionalità di replica logica nativa di PostgreSQL, i cluster database multi-AZ che eseguono RDS per PostgreSQL supportano anche l'estensione `pglogical`.

Per ulteriori informazioni sulla funzionalità di replica logica di PostgreSQL, consulta la sezione relativa alla replica logica nella documentazione di PostgreSQL.

Argomenti

Prerequisiti

Per configurare la funzionalità di replica logica di PostgreSQL per i cluster database multi-AZ, è necessario soddisfare i seguenti prerequisiti.

- L'account utente deve essere membro del gruppo `rds_superuser` e disporre dei privilegi `rds_superuser`. Per ulteriori informazioni, consulta the section called “Informazioni su ruoli e autorizzazioni di PostgreSQL”.

- Il cluster database multi-AZ deve essere associato a un gruppo di parametri del cluster database personalizzato in modo da poter configurare i valori dei parametri descritti nella procedura seguente. Per ulteriori informazioni, consulta the section called “Utilizzo di gruppi di parametri di cluster di database”.

Configurazione della replica logica

Per configurare la replica logica per un cluster database multi-AZ, devi abilitare parametri specifici all'interno del gruppo di parametri del cluster database associato, quindi creare slot di replica logica.

> **Note**
>
> A partire dalla versione 16 di PostgreSQL, è possibile utilizzare istanze Reader DB del cluster DB Multi-AZ per la replica logica.

Configurazione della replica logica per un cluster database multi-AZ RDS per PostgreSQL

1. Apri il gruppo di parametri del cluster database personalizzato associato al cluster database multi-AZ RDS per PostgreSQL.
2. Nel campo di ricerca Parametri, individua il parametro statico `rds.logical_replication` e imposta il relativo valore su 1. La modifica di questo parametro può aumentare la generazione di WAL, quindi abilita questo parametro la solo quando utilizzi slot logici.
3. Nell'ambito di questa modifica, configura i seguenti parametri del cluster database.

   - `max_wal_senders`
   - `max_replication_slots`
   - `max_connections`

   A secondo dell'utilizzo previsto, potrebbe anche essere necessario modificare i valori dei seguenti parametri. Tuttavia, in molti casi, i valori predefiniti sono sufficienti.

   - `max_logical_replication_workers`
   - `max_sync_workers_per_subscription`
4. Riavvia il cluster database multi-AZ per rendere effettivi i valori dei parametri. Per istruzioni, consulta the section called “Riavvio di un cluster di database multi-AZ”.

5. Crea uno slot di replica logica sull'istanza database di scrittura del cluster database multi-AZ come illustrato in the section called “Lavorare con gli slot di replica logica”. Questo processo richiede che venga specificato un plug-in di decodifica. Attualmente RDS per PostgreSQL supporta i plugin `test_decoding`, `wal2json` e `pgoutput` forniti con PostgreSQL.

   Lo slot viene copiato in modo asincrono su ogni istanza database di lettura nel cluster.

6. Verifica lo stato dello slot su tutte le istanze database di lettura del cluster database multi-AZ. A tale scopo, verifica la vista `pg_replication_slots` su tutte le istanze database di lettura e assicurati che lo stato `confirmed_flush_lsn` stia progredendo mentre l'applicazione sta utilizzando attivamente le modifiche logiche.

   I seguenti comandi mostrano come controllare lo stato di replica sulle istanze database di lettura.

```
% psql -h test-postgres-instance-2.abcdefabcdef.us-west-2.rds.amazonaws.com

postgres=> select slot_name, slot_type, confirmed_flush_lsn from
 pg_replication_slots;
  slot_name   | slot_type | confirmed_flush_lsn
--------------+-----------+---------------------
 logical_slot | logical   | 32/D0001700
(1 row)

postgres=> select slot_name, slot_type, confirmed_flush_lsn from
 pg_replication_slots;
  slot_name   | slot_type | confirmed_flush_lsn
--------------+-----------+---------------------
 logical_slot | logical   | 32/D8003628
(1 row)

% psql -h test-postgres-instance-3.abcdefabcdef.us-west-2.rds.amazonaws.com

postgres=> select slot_name, slot_type, confirmed_flush_lsn from
 pg_replication_slots;
  slot_name   | slot_type | confirmed_flush_lsn
--------------+-----------+---------------------
 logical_slot | logical   | 32/D0001700
(1 row)

postgres=> select slot_name, slot_type, confirmed_flush_lsn from
 pg_replication_slots;
  slot_name   | slot_type | confirmed_flush_lsn
--------------+-----------+---------------------
```

```
 logical_slot | logical   | 32/D8003628
(1 row)
```

Dopo aver completato le attività di replica, interrompi il processo di replica, elimina gli slot di replica e disattiva la replica logica. Per disattivare la replica logica, modifica il gruppo di parametri del cluster database e reimposta il valore di `rds.logical_replication` su `0`. Riavvia il cluster database per applicare la modifica apportata al parametro.

Limitazioni e consigli

Le seguenti limitazioni e raccomandazioni si applicano all'utilizzo della replica logica con cluster DB Multi-AZ che eseguono PostgreSQL versione 16:

- È possibile utilizzare solo istanze Writer DB per creare o eliminare slot di replica logica. Ad esempio, il `CREATE SUBSCRIPTION` comando deve utilizzare l'endpoint cluster writer nella stringa di connessione host.
- È necessario utilizzare l'endpoint cluster writer durante qualsiasi sincronizzazione o risincronizzazione delle tabelle. Ad esempio, è possibile utilizzare i seguenti comandi per risincronizzare una tabella appena aggiunta:

  ```
  Postgres=>ALTER SUBSCRIPTION subscription-name CONNECTION host=writer-endpoint
  Postgres=>ALTER SUBSCRIPTION subscription-name REFRESH PUBLICATION
  ```
- È necessario attendere il completamento della sincronizzazione della tabella prima di utilizzare le istanze DB del lettore per la replica logica. È possibile utilizzare la tabella del `pg_subscription_rel` catalogo per monitorare la sincronizzazione delle tabelle. La sincronizzazione della tabella è completa quando la `srsubstate` colonna è impostata su ready ()`r`.
- Si consiglia di utilizzare gli endpoint dell'istanza per la connessione di replica logica una volta completata la sincronizzazione iniziale della tabella. Il comando seguente riduce il carico sull'istanza DB di Writer affidando la replica a una delle istanze DB Reader:

  ```
  Postgres=>ALTER SUBSCRITPION subscription-name CONNECTION host=reader-instance-
  endpoint
  ```

  Non è possibile utilizzare lo stesso slot su più di un'istanza DB alla volta. Quando due o più applicazioni replicano le modifiche logiche da diverse istanze DB del cluster, alcune modifiche potrebbero andare perse a causa di un failover del cluster o di un problema di rete. In queste

situazioni, è possibile utilizzare gli endpoint dell'istanza per la replica logica nella stringa di connessione host. L'altra applicazione che utilizza la stessa configurazione mostrerà il seguente messaggio di errore:

```
replication slot slot_name is already active for PID x providing immediate feedback.
```

- Durante l'utilizzo dell'`pglogical`estensione, è possibile utilizzare solo l'endpoint cluster writer. L'estensione presenta limitazioni note che possono creare slot di replica logica inutilizzati durante la sincronizzazione delle tabelle. Gli slot di replica obsoleti riservano i file WAL (write-ahead log) e possono causare problemi di spazio su disco.

Per eliminare un cluster di database Multi-AZ

È possibile eliminare un cluster DB Multi-AZ utilizzando l' AWS Management Console, la o l' AWS CLIAPI RDS. Per eliminare un cluster DB Multi-AZ, devi prima eliminare tutte le sue istanze DB.

Il tempo necessario per eliminare un cluster DB Multi-AZ può variare in base ai seguenti fattori:

- Il periodo di conservazione dei backup (ovvero il numero di backup da eliminare).
- Quanti dati vengono eliminati.
- Se viene scattata un'istantanea finale.

La protezione da eliminazione deve essere disabilitata sul cluster DB Multi-AZ prima di poterla eliminare. Per ulteriori informazioni, consulta the section called “Prerequisiti per l'eliminazione di un'istanza database”. È possibile disabilitare la protezione da eliminazione modificando il cluster DB Multi-AZ. Per ulteriori informazioni, consulta the section called “Modifica di un cluster di database Multi-AZ”.

Console

Per eliminare un cluster database Multi-AZ

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel pannello di navigazione, scegliere Databases (Database), quindi scegliere il cluster di database Multi-AZ che si desidera eliminare.
3. In Actions (Azioni), scegliere Delete (Elimina).
4. Scegliere Create final snapshot? (Crea snapshot finale?) per creare uno snapshot DB Multi-AZ finale per il cluster di database.

   Se si è scelto di creare uno snapshot finale, immettere un nome per Final snapshot name (Nome dello snapshot finale).
5. Per mantenere i backup automatici, scegliere Retain automated backups (Mantieni backup automatici).
6. Immettere **`delete me`** nella casella.
7. Scegliere Delete (Elimina).

AWS CLI

Per eliminare un cluster DB Multi-AZ utilizzando il AWS CLI, chiama il delete-db-clustercomando con le seguenti opzioni:

- `--db-cluster-identifier`
- `--final-db-snapshot-identifier` o `--skip-final-snapshot`

Example Con uno snapshot finale

Per LinuxmacOS, oUnix:

```
aws rds delete-db-cluster \
    --db-cluster-identifier mymultiazdbcluster \
    --final-db-snapshot-identifier mymultiazdbclusterfinalsnapshot
```

Per Windows:

```
aws rds delete-db-cluster ^
    --db-cluster-identifier mymultiazdbcluster ^
    --final-db-snapshot-identifier mymultiazdbclusterfinalsnapshot
```

Example Con uno snapshot finale

Per LinuxmacOS, oUnix:

```
aws rds delete-db-cluster \
    --db-cluster-identifier mymultiazdbcluster \
    --skip-final-snapshot
```

Per Windows:

```
aws rds delete-db-cluster ^
    --db-cluster-identifier mymultiazdbcluster ^
    --skip-final-snapshot
```

API RDS

Per eliminare un cluster di database Multi-AZ tramite l'API Amazon RDS, chiamare l'operazione DeleteDBCluster con i parametri seguenti:

- `DBClusterIdentifier`
- `FinalDBSnapshotIdentifier` o `SkipFinalSnapshot`

Limitazioni dei cluster DB Multi-AZ

Un cluster di database Multi-AZ ha un'istanza database di scrittore e due istanze database di lettore in tre zone di disponibilità separate. I cluster di database multi-AZ offrono elevata disponibilità, maggiore capacità per i carichi di lavoro in lettura e minore latenza rispetto alle implementazioni Multi-AZ. Per ulteriori informazioni sui cluster di database Multi-AZ, consulta Implementazioni cluster di database multi-AZ.

Le seguenti limitazioni si applicano ai cluster DB Multi-AZ.

- I cluster di database Multi-AZ non supportano le seguenti funzionalità:
  - Connessioni IPv6 (modalità dual-stack)
  - Backup automatici tra regioni
  - Autenticazione IAM DB e autenticazione Kerberos
  - Modifica della porta. In alternativa, è possibile ripristinare un cluster di database Multi-AZ a un punto nel tempo e specificare una porta diversa.
  - Gruppi di opzioni
  - P oint-in-time-recovery (PITR) per i cluster eliminati
  - Esportazione dei dati di istantanee del cluster Multi-AZ DB in un bucket S3 o ripristino di un'istantanea del cluster DB Multi-AZ da un bucket S3
  - Scalabilità automatica dello storage impostando lo storage massimo allocato. In alternativa, puoi scalare manualmente l’archiviazione.
  - Arresto e avvio del cluster DB Multi-AZ
  - Copia di uno snapshot di un cluster di database Multi-AZ
  - Crittografia di un cluster di database Multi-AZ non crittografato
- I cluster di database Multi-AZ RDS per MySQL non supportano la replica su un database di destinazione esterno.
- I cluster di database Multi-AZ di RDS per MySQL supportano solo le seguenti procedure archiviate nel sistema:
  - `mysql.rds_rotate_general_log`
  - `mysql.rds_rotate_slow_log`
  - `mysql.rds_show_configuration`
  - `mysql.rds_set_external_master_with_auto_position`

- I cluster DB RDS for PostgreSQL Multi-AZ non supportano le seguenti estensioni: e. `aws_s3` `pg_transport`
- I cluster di database Multi-AZ RDS for PostgreSQL non supportano l'utilizzo di un server DNS personalizzato per l'accesso alla rete in uscita.

Utilizzo dell'estensione del supporto per Amazon RDS

Con l'estensione del supporto di Amazon RDS, puoi continuare a eseguire il tuo database su una versione principale del motore dopo la data di fine del supporto standard per RDS a un costo aggiuntivo. Alla data di fine del supporto standard di RDS , Amazon RDS Amazon registra automaticamente i database in RDS Extended Support. La registrazione automatica a RDS Extended Support non modifica il motore del database e non influisce sull'uptime o sulle prestazioni dell'istanza DB.

Questa offerta a pagamento ti offre più tempo per eseguire l'aggiornamento a una versione del motore principale supportata.

Ad esempio, la data di fine del supporto RDS standard per MySQL versione 5.7 è il 29 febbraio 2024. Tuttavia, non sei pronto per l'aggiornamento manuale a RDS for MySQL versione 8.0 prima di tale data. In questo caso, Amazon RDS registra automaticamente i tuoi database in RDS Extended Support il 29 febbraio 2024 e puoi continuare a eseguire RDS for MySQL versione 5.7. A partire dal 1° marzo 2024, Amazon RDS ti addebita automaticamente l'importo del servizio RDS Extended Support.

RDS Extended Support è disponibile fino a 3 anni dopo la data di fine del supporto standard di RDS per una versione principale del motore MySQL versione 2). Trascorso questo periodo, se non hai aggiornato la versione del motore principale a una versione supportata, Amazon RDS Amazon aggiornerà automaticamente la versione principale del motore. Ti consigliamo di eseguire l'aggiornamento a una versione del motore principale supportata il prima possibile.

Argomenti

Panoramica di Amazon RDS Extended Support

Dopo la data di fine del supporto standard di RDS , Amazon RDS Amazon registrerà automaticamente i database in RDS Extended Support. Amazon RDS aggiorna automaticamente l'istanza DB all'ultima versione secondaria rilasciata prima della fine della data di supporto standard di RDS , se non utilizzi già quella versione. Amazon RDS Aurora effettuerà l'upgrade della versione secondaria solo dopo la fine della data di supporto standard di RDS per la versione principale del motore.

È possibile creare nuovi database con le principali versioni del motore che hanno raggiunto la data di fine del supporto standard di RDS . RDS automaticamente questi nuovi database in RDS Extended Support e ti addebita il costo di questa offerta.

Se esegui l'upgrade a un motore che è ancora coperto dal supporto standard RDS prima della data di fine del supporto standard di RDS , Amazon RDS Amazon Support.

Se il ripristino non riesce, Amazon RDS Aurora registrerà automaticamente il motore a RDS Extended Support con una versione compatibile con lo snapshot.

Puoi terminare l'iscrizione a RDS Extended Support in qualsiasi momento.

Argomenti

Costi di Amazon RDS Extended Support

Saranno addebitati costi per tutti i motori iscritti a RDS Extended Support a partire dal giorno successivo alla data di fine del supporto standard RDS. Per la data di fine del supporto standard RDS, consulta Versioni principali di MySQL supportate il Calendario di rilascio di Amazon RDS for PostgreSQL. I costi di RDS Extended Support si applicano alle istanze di standby nelle implementazioni Multi-AZ.

Il costo aggiuntivo per RDS Extended Support si interrompe automaticamente quando si esegue una delle seguenti azioni:

- Effettua l'upgrade a una versione del motore coperta dal supporto standard.

- Elimina il database su cui è in esecuzione una versione principale dopo la data di fine del supporto standard di RDS .

Gli addebiti verranno riavviati se la versione del motore di destinazione entrerà in RDS Extended Support in futuro.

Ad esempio, RDS per PostgreSQL 11 entrerà in Extended Support il 1° marzo 2024, ma i costi non inizieranno fino al 1° aprile 2024. Ti verranno addebitati solo 30 giorni di Extended Support su RDS per PostgreSQL Aurora PostgreSQL . Continuerai a eseguire RDS per PostgreSQL 12 su questa istanza DB oltre la data di fine del supporto standard RDS del 28 febbraio 2025. Il tuo database sarà nuovamente soggetto ai costi di RDS Extended Support a partire dal 1° marzo 2025.

Per ulteriori informazioni, consulta Prezzi di Amazon RDS per MySQL e Prezzi di Amazon RDS per PostgreSQL.

Evitare i costi per Amazon RDS Extended Support

A tale scopo, utilizza o l'API RDS. AWS CLI

Nel AWS CLI, specificare `open-source-rds-extended-support-disabled` l'`--engine-lifecycle-support`opzione. Nell'API RDS, specificare `open-source-rds-extended-support-disabled` il `LifeCycleSupport` parametro. Per ulteriori informazioni, consulta Creazione di un'istanza DB o di un cluster DB Multi-AZ, di un cluster o Ripristino di un'istanza DB o di un cluster DB Multi-AZ, di un cluster .

Versioni con Amazon RDS Extended Support

RDS Extended Support è disponibile solo per le versioni principali. Non è disponibile per le versioni minori.

RDS Extended Support è disponibile per RDS per MySQL 5.7 e 8.0 e per RDS per PostgreSQL 11 e versioni successive. Per ulteriori informazioni, consulta Versioni principali di MySQL supportate il calendario di rilascio di Amazon RDS for PostgreSQL nelle note di rilascio di Amazon RDS for PostgreSQL.

Denominazione delle versioni di Amazon RDS Extended Support

Amazon RDS Aurora rilascerà nuove versioni minori con correzioni e patch CVE per i motori su RDS Extended Support. Per ulteriori informazioni, consulta Versioni Amazon RDS Extended Support per

RDS per MySQL gli aggiornamenti di Amazon RDS Extended Support per RDS for PostgreSQL nelle note di rilascio di Amazon RDS for PostgreSQL.

I nomi di queste versioni minori saranno nel formato major.minor-RDS.YYYYMMDD.patch.YYYYMMDD, ad esempio 5.7.44-RDS.20240208.R2.20240210 (per RDS per MySQL) 11.22-RDS.20240208.R2.20240210 o (per RDS per PostgreSQL).

importante

Per MySQL, il numero di versione principale è sia il numero intero che la prima parte frazionaria del numero di versione, ad esempio 8.0. Un aggiornamento della versione principale incrementa la parte principale del numero di versione. Ad esempio, un aggiornamento da 5.7.44 a 8.0.33 è un aggiornamento della versione principale, dove 5.7 e 8.0 sono i numeri di versione principali.

Per PostgreSQL, il numero di versione principale è il numero intero, ad esempio 11.

minor-RDS.YYYYMMDD

Per MySQL, il numero di versione secondario è la terza parte del numero di versione, ad esempio, in. 44-RDS.20240208 5.7.44-RDS.20240208

Per PostgeSQL, il numero di versione secondario è la seconda parte del numero di versione, ad esempio, la in. 22-RDS.20240208 11.22-RDS.20240208

La data è quella in cui Amazon RDS ha creato la versione secondaria di Amazon RDS.

patch

La versione patch è quella che segue la data in cui Amazon RDS ha creato la versione secondaria di Amazon RDS, ad esempio R2 in o. 5.7.44-RDS.20240208.R2 11.22-RDS.20240208.R2

Una versione patch di Amazon RDS include importanti correzioni di bug aggiunte a una versione secondaria di Amazon RDS dopo il suo rilascio.

YYYYMMGD

La data è quella in cui Amazon RDS ha creato la versione patch, ad esempio 20240210 in o. 5.7.44-RDS.20240208.R2.20240210 11.22-RDS.20240208.R2.20240210

Una versione datata di Amazon RDS è una patch di sicurezza che include importanti correzioni di sicurezza aggiunte a una versione secondaria dopo il suo rilascio. Non include correzioni che potrebbero modificare il comportamento di un motore.

Amazon RDS Aurora e le responsabilità dei clienti con Amazon RDS Extended Support

Il seguente contenuto descrive le responsabilità di Amazon RDS Aurora e le tue responsabilità con RDS Extended Support.

Argomenti

Responsabilità di Amazon RDS

Dopo la data di fine del supporto standard per RDS , Amazon RDS Amazon fornirà patch, correzioni di bug e aggiornamenti per i motori registrati in RDS Extended Support. Ciò avverrà per un massimo di 3 anni o fino a quando non smetterai di usare i motori, a seconda dell'evento che si verifica per primo.

Le patch riguarderanno i CVE critici e quelli elevati, secondo quanto definito dalle classificazioni di gravità CVSS del National Vulnerability Database (NVD). Per ulteriori informazioni, consulta Vulnerability Metrics (Metriche relative alla vulnerabilità).

Le tue responsabilità

Sei responsabile dell'applicazione delle patch, delle correzioni di bug e degli aggiornamenti forniti per le istanze DB o i cluster DB Multi-AZ, i cluster Aurora DB o i cluster globali . Amazon RDS Aurora si riserva il diritto di modificare, sostituire o ritirare tali patch, correzioni di bug e upgrade in qualsiasi momento. Se è necessaria una patch per risolvere problemi critici di sicurezza o stabilità, Amazon RDS Aurora si riserva il diritto di aggiornare le istanze DB o i cluster DB Multi-AZ, di richiedere l'installazione della patch.

Sei inoltre responsabile dell'aggiornamento del motore a una versione più recente prima della data di fine del servizio RDS di Extended Support. La data di fine del supporto esteso RDS è in genere 3 anni dopo la supporto standard RDS. Per la data di fine dell'Extended Support RDS per la versione principale del motore di database, consulta Versioni principali di MySQL supportate il calendario di rilascio di Amazon RDS for PostgreSQL.

Se l'aggiornamento non riesce, Amazon RDS Aurora si riserva il diritto di eliminare l'istanza DB o il cluster DB Multi-AZ, il cluster Aurora DB che esegue il motore oltre la data di fine del supporto standard di RDS . Tuttavia, prima di farlo, Amazon RDS Aurora conserverà i dati di quel motore.

Creazione di un'istanza DB o di un cluster DB Multi-AZ, un cluster con Amazon RDS Extended Support

Quando crei un'istanza DB o un cluster DB Multi-AZ, un cluster , seleziona Enable RDS Extended Support nella console o utilizza l'opzione Extended Support AWS CLI in o il parametro nell'API RDS.

> **Note**
>
> Se non si specifica l'impostazione RDS Extended Support, per impostazione predefinita RDS è RDS Extended Support. Questo comportamento predefinito mantiene la disponibilità del database oltre la data di fine del supporto standard di RDS .

Argomenti

- Considerazioni per RDS Extended Support
- Crea un'istanza DB o un cluster DB Multi-AZ, un cluster con RDS Extended Support

Considerazioni per RDS Extended Support

Prima di creare un'istanza DB o un cluster DB Multi-AZ, un cluster , considera i seguenti elementi:

- Una volta trascorsa la data di fine del supporto standard di RDS , puoi impedire la creazione di una nuova istanza DB o di un nuovo cluster DB Multi-AZ, un nuovo cluster Aurora DB Support. A tale scopo, utilizza o l'API RDS. AWS CLI Nel AWS CLI, specificare `open-source-rds-extended-support-disabled` l'`--engine-lifecycle-support`opzione. Nell'API RDS, specificare `open-source-rds-extended-support-disabled` il `LifeCycleSupport` parametro. Se si specifica `open-source-rds-extended-support-disabled` e la data di fine del supporto standard di RDS è trascorsa, la creazione di un'istanza DB o di un cluster DB Multi-AZ, un cluster globale avrà sempre esito negativo.
- RDS Extended Support è impostato a livello di cluster. I membri di un cluster avranno sempre la stessa impostazione per RDS Extended Support nella console RDS AWS CLI, `--engine-lifecycle-support` nella e `EngineLifecycleSupport` nell'API RDS.

Per ulteriori informazioni, consulta i Versioni di MySQL calendari di rilascio per Amazon RDS for PostgreSQL.

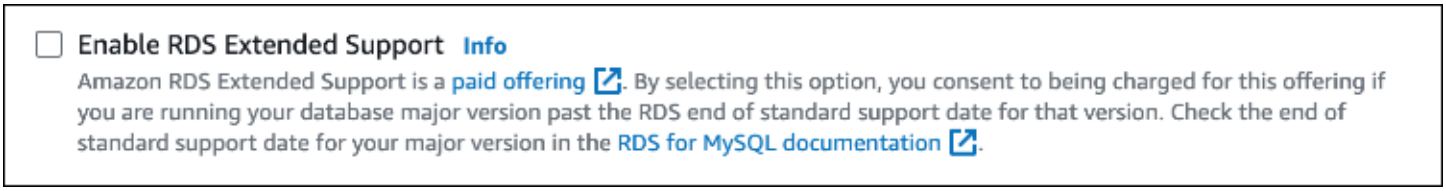
Crea un'istanza DB o un cluster DB Multi-AZ, un cluster con RDS Extended Support

È possibile creare un'istanza DB o un cluster DB Multi-AZ, un cluster versione RDS Extended Support AWS Management Console utilizzando l'API AWS CLI, the o RDS.

Console

Quando DB Multi-AZ, nella sezione Opzioni del motore, seleziona Enable RDS Extended Support.

L'immagine seguente mostra l'impostazione Enable RDS Extended Support:



AWS CLI

Quando si utilizza il AWS CLI comando or create-db-cluster(Multi-AZ DB cluster), selezionare RDS Extended Support specificando `open-source-rds-extended-support` l'opzione. `--engine-lifecycle-support` Per impostazione predefinita, questa opzione è impostata su. `open-source-rds-extended-support`

Per impedire la creazione di una nuova istanza DB o di un cluster DB Multi-AZ, di un nuovo cluster dopo la data di fine del supporto standard di RDS , specificare l'opzione. `open-source-rds-extended-support-disabled` `--engine-lifecycle-support` In questo modo, eviterai i costi associati all'RDS Extended Support.

API RDS

Quando utilizzi l'operazione dell'API Amazon o CreateDBCluster (cluster DB Multi-AZ), seleziona RDS Extended Support impostando il parametro su. `EngineLifecycleSupport` `open-source-rds-extended-support` Questo parametro è impostato su `open-source-rds-extended-support` per impostazione predefinita.

Per impedire la creazione di una nuova istanza DB o di un cluster DB Multi-AZ, di un nuovo cluster dopo la data di fine del supporto standard di RDS , specificare il parametro. `open-source-rds-`

`extended-support-disabled` `EngineLifecycleSupport` In questo modo, eviterai i costi associati all'RDS Extended Support.

Per ulteriori informazioni, consulta i seguenti argomenti:

- Per creare un'istanza database, seguire le istruzioni per il proprio motore database in Creazione di un'istanza database Amazon RDS.
- Per creare un cluster di database Multi-AZ, seguire le istruzioni in Creazione di un cluster di database Multi-AZ.

Visualizzazione della registrazione delle istanze DB o dei cluster DB Multi-AZ, dei cluster Aurora DB o dei cluster

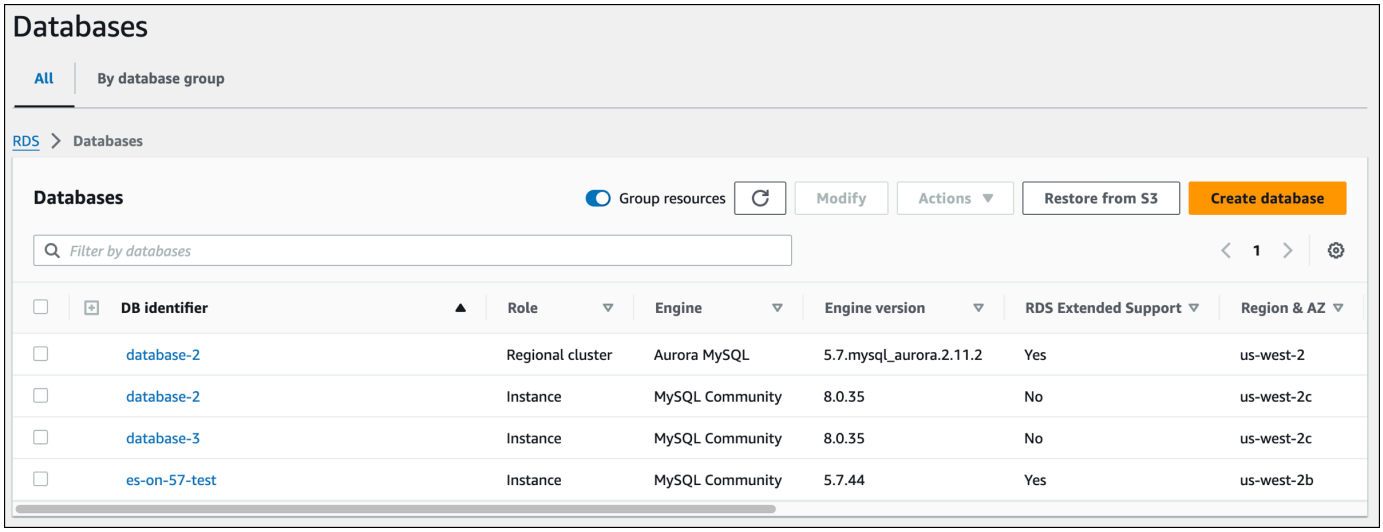
È possibile visualizzare la registrazione delle istanze DB o dei cluster DB Multi-AZ, dei cluster Aurora DB o dei cluster utilizzando il. AWS Management Console

Console

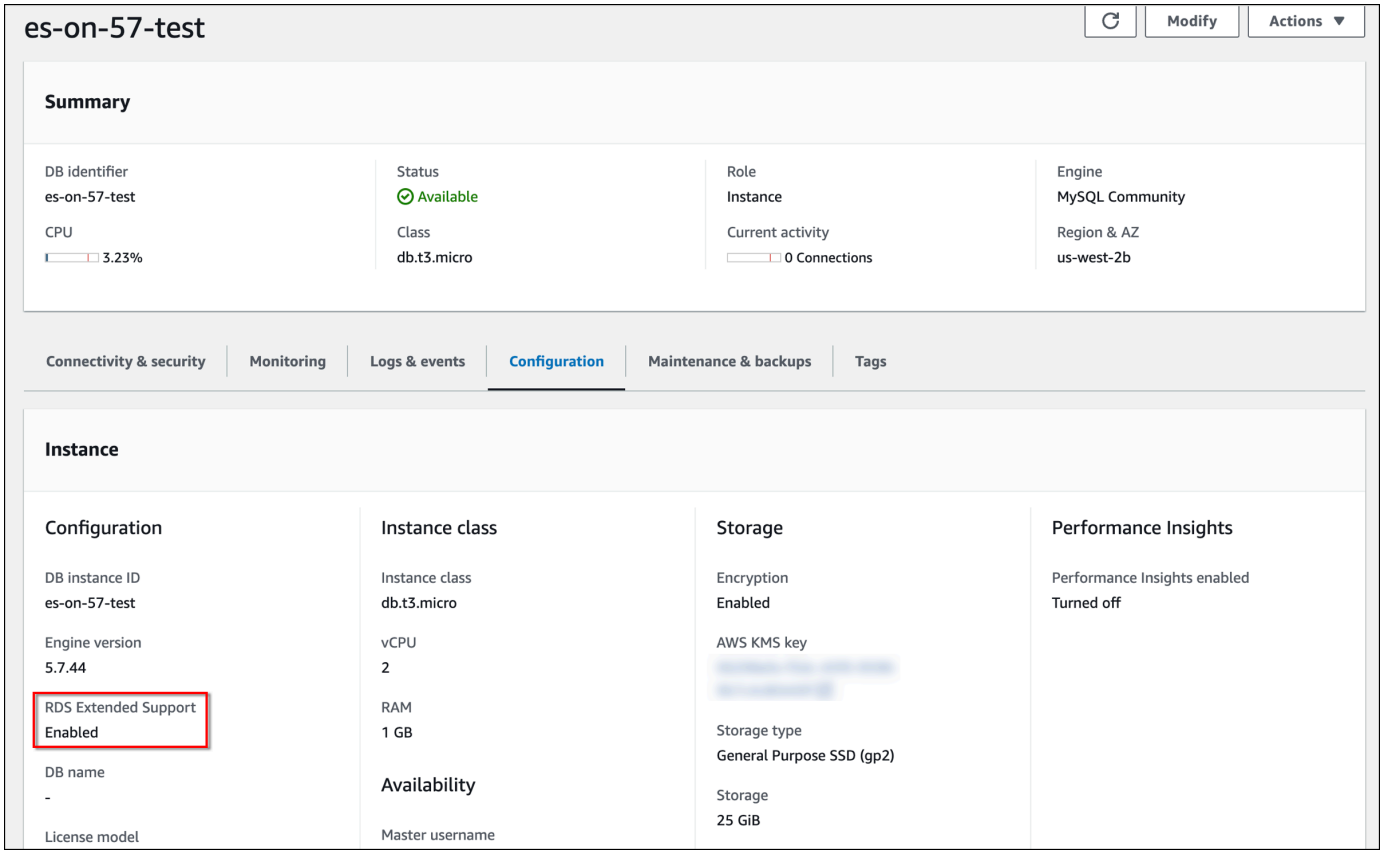
Per visualizzare la registrazione delle istanze DB o dei cluster DB Multi-AZ, dei cluster Aurora DB o dei cluster

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, scegli Databases (Database). Il valore in RDS Extended Support indica se un'istanza DB o un cluster DB Multi-AZ, un cluster è registrato in RDS Extended Support. Se non viene visualizzato alcun valore, RDS Extended Support non è disponibile per il database.

> **Tip**
>
> Se la colonna RDS Extended Support non viene visualizzata, scegli l'icona Preferenze, quindi attiva RDS Extended Support.



3. È inoltre possibile visualizzare la registrazione nella scheda Configurazione per ogni database. Scegli un database sotto l'identificatore DB. Nella scheda Configurazione, guarda in Extended Support per vedere se il database è registrato o meno.



Ripristino di un'istanza DB o di un cluster DB Multi-AZ, di un cluster Amazon RDS Extended Support

Quando ripristini un'istanza DB o un cluster DB Multi-AZ, un cluster , seleziona Enable RDS Extended Support nella console o utilizza l'opzione Extended Support AWS CLI in o il parametro nell'API RDS.

> **Note**
>
> Se non si specifica l'impostazione RDS Extended Support, per impostazione predefinita RDS è RDS Extended Support. Questo comportamento predefinito mantiene la disponibilità del database oltre la data di fine del supporto standard di RDS .

Argomenti

Considerazioni per RDS Extended Support

Prima di ripristinare un'istanza DB o un cluster DB Multi-AZ, un cluster , considera i seguenti elementi:

- Una volta trascorsa la data di fine del supporto standard di RDS , se desideri ripristinare un'istanza DB o un cluster DB Multi-AZ, un cluster Amazon S3, puoi farlo solo utilizzando o l'API RDS. AWS CLI Utilizza l'`--engine-lifecycle-support`opzione nel AWS CLI comando restore-db-cluster-from-s3 o il `EngineLifecycleSupport` parametro nell'operazione dell'API RestoreDB S3 RDS. ClusterFrom
- Se desideri impedire a RDS di ripristinare i database alle versioni di RDS Extended Support, `open-source-rds-extended-support-disabled` specifica in o AWS CLI nell'API RDS. In questo modo, eviterai i costi associati all'RDS Extended Support.

  Se specifichi questa impostazione, Amazon RDS Aurora aggiornerà automaticamente il database ripristinato a una versione principale più recente e supportata. Se l'upgrade non supera i controlli pre-aggiornamento, Amazon RDS Amazon tornerà in modo sicuro alla versione del motore RDS Extended Support. Questo database rimarrà in modalità RDS Extended Support e Amazon RDS

Amazon ti addebiterà il costo del supporto RDS Extended Support fino all'aggiornamento manuale del database.

Ad esempio, se ripristini uno snapshot MySQL 5.7 senza utilizzare RDS Extended Support, Amazon RDS tenterà di aggiornare automaticamente il database a MySQL 8.0. Se questo aggiornamento non riesce a causa di un problema che devi risolvere, Amazon RDS ripristinerà il database a MySQL 5.7. Amazon RDS manterrà il database su RDS Extended Support fino a quando non sarà possibile risolvere il problema. Ad esempio, un aggiornamento potrebbe non riuscire a causa di spazio di archiviazione insufficiente. Dopo aver risolto il problema, è necessario avviare l'aggiornamento. Dopo il primo tentativo di aggiornamento del database, Amazon RDS non tenterà più di aggiornarlo.

- RDS Extended Support è impostato a livello di cluster. I membri di un cluster avranno sempre la stessa impostazione per RDS Extended Support nella console RDS AWS CLI, `--engine-lifecycle-support` nella e `EngineLifecycleSupport` nell'API RDS.

Per ulteriori informazioni, consulta i Versioni di MySQL calendari di rilascio per Amazon RDS for PostgreSQL.

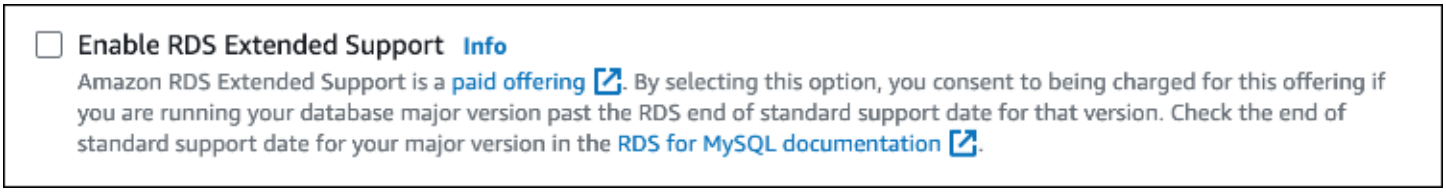
Ripristina un'istanza DB o un cluster DB Multi-AZ, un cluster DB con RDS Extended Support

È possibile ripristinare un'istanza DB o un cluster DB Multi-AZ, un cluster versione RDS Extended Support AWS Management Console utilizzando l'API AWS CLI, the o RDS.

Console

Quando ripristini DB Multi-AZ, seleziona Enable RDS Extended Support nella sezione delle opzioni del motore.

L'immagine seguente mostra l'impostazione Enable RDS Extended Support:



AWS CLI

Quando si utilizza il restore-db-cluster-fromcomando o -snapshot AWS CLI , selezionare RDS Extended Support specificando l'opzione. `open-source-rds-extended-support` `--engine-lifecycle-support`

Se desideri evitare i costi associati a RDS Extended Support, imposta l'`--engine-lifecycle-support`opzione su. `open-source-rds-extended-support-disabled` Per impostazione predefinita, questa opzione è impostata su. `open-source-rds-extended-support`

È inoltre possibile specificare questo valore utilizzando i seguenti AWS CLI comandi:

- restore-db-cluster-from-s3
- restore-db-cluster-to-point-in-time
- restore-db-instance-from-s3
- restore-db-instance-to-point-in-time

API RDS

Quando si utilizza l'operazione dbSnapshot o ClusterFromSnapshot RestoreDB ClusterFromSnapshot RDS API, selezionare RDS Extended Support impostando il parametro su. `EngineLifecycleSupport` `open-source-rds-extended-support`

Per evitare i costi associati a RDS Extended Support, imposta il `EngineLifecycleSupport` parametro su. `open-source-rds-extended-support-disabled` Questo parametro è impostato su `open-source-rds-extended-support` per impostazione predefinita.

È inoltre possibile specificare questo valore utilizzando le seguenti operazioni dell'API RDS:

- Ripristina DB ClusterFrom S3
- Restore DB ClusterToPointInTime
- Ripristina DB S3 InstanceFrom
- Restore DB InstanceToPointInTime

Per ulteriori informazioni sul ripristino di un'istanza DB o di un cluster DB Multi-AZ, segui le istruzioni per il tuo motore DB in. Ripristino da uno snapshot database

Utilizzo delle implementazioni blu/verde Amazon RDS per gli aggiornamenti del database

Un'implementazione blu/verde copia un ambiente di database di produzione in un ambiente di gestione temporanea separato e sincronizzato. Utilizzando le implementazioni blu/verde Amazon RDS, puoi apportare modifiche al database nell'ambiente di gestione temporanea senza influire sull'ambiente di produzione. Ad esempio, è possibile aggiornare la versione principale o secondaria del motore di database, modificare i parametri del database o apportare modifiche allo schema nell'ambiente di gestione temporanea. Quando sei pronto, puoi promuovere l'ambiente di staging come nuovo ambiente di database di produzione, con tempi di inattività in genere inferiori a un minuto.

> **Note**
>
> Attualmente, le implementazioni Blue/Green sono supportate solo per RDS per MariaDB, RDS per MySQL e RDS per PostgreSQL. Per la disponibilità di Amazon Aurora, consulta Utilizzo delle implementazioni blu/verde Amazon RDS per gli aggiornamenti del database nella Guida per l'utente di Amazon Aurora.

Argomenti

Panoramica delle implementazioni blu/verde Amazon RDS

Con le implementazioni blu/verde di Amazon RDS puoi apportare e testare le modifiche del database prima di implementarle in un ambiente di produzione. Un'implementazione blu/verde crea un ambiente di gestione temporanea che copia l'ambiente di produzione. In un'implementazione blu/verde, l'ambiente blu è l'ambiente di produzione corrente. L'ambiente verde è l'ambiente di gestione temporanea. L'ambiente di gestione temporanea rimane sincronizzato con l'ambiente di produzione corrente utilizzando la replica logica.

È possibile apportare modifiche alle istanze database RDS nell'ambiente verde senza influire sui carichi di lavoro di produzione. Ad esempio, è possibile aggiornare la versione principale o secondaria del motore di database, aggiornare la configurazione del file system sottostante o modificare i parametri di database nell'ambiente di gestione temporanea. È possibile testare le modifiche nell'ambiente verde. Quando sei pronto, puoi passare agli ambienti per promuovere l'ambiente verde nel nuovo ambiente di produzione. Lo switchover richiede in genere meno di un minuto senza perdita di dati e senza la necessità di modificare l'applicazione.

Poiché è una copia della topologia dell'ambiente di produzione, l'ambiente verde include le funzionalità utilizzate dall'istanza database. Queste funzionalità comprendono le repliche di lettura, la configurazione dell'archiviazione, gli snapshot del database, i backup automatici, approfondimenti sulle prestazioni e il monitoraggio avanzato. Se l'istanza database blu è un'implementazione di istanza database multi-AZ, anche l'istanza database verde è un'implementazione di istanza database multi-AZ.

> **Note**
>
> Attualmente, le implementazioni blu/verde sono supportate solo per RDS per MariaDB, RDS per MySQL e RDS per PostgreSQL. Per la disponibilità di Amazon Aurora, consulta Using Amazon RDS Blue/Green Deployments per gli aggiornamenti del database nella Amazon Aurora User Guide.

Argomenti

- Considerazioni sulle implementazioni blu/verde
- Best practice per le implementazioni blu/verde
- Limitazioni per le implementazioni blu/verde

Disponibilità di regioni e versioni

Il supporto varia a seconda delle versioni specifiche di ciascun motore di database e a seconda delle Regioni AWS. Per ulteriori informazioni, consulta the section called “Distribuzioni blu/verdi”.

Vantaggi dell'utilizzo delle implementazioni blu/verde Amazon RDS

Con le implementazioni blu/verde Amazon RDS puoi rimanere aggiornato sulle patch di sicurezza, migliorare le prestazioni del database e adottare nuove funzionalità del database con tempi di inattività brevi e prevedibili. Le implementazioni blu/verde riducono i rischi e i tempi di inattività per gli aggiornamenti del database, ad esempio gli aggiornamenti della versione principale o secondaria del motore.

Le implementazioni blu/verde offrono i seguenti vantaggi:

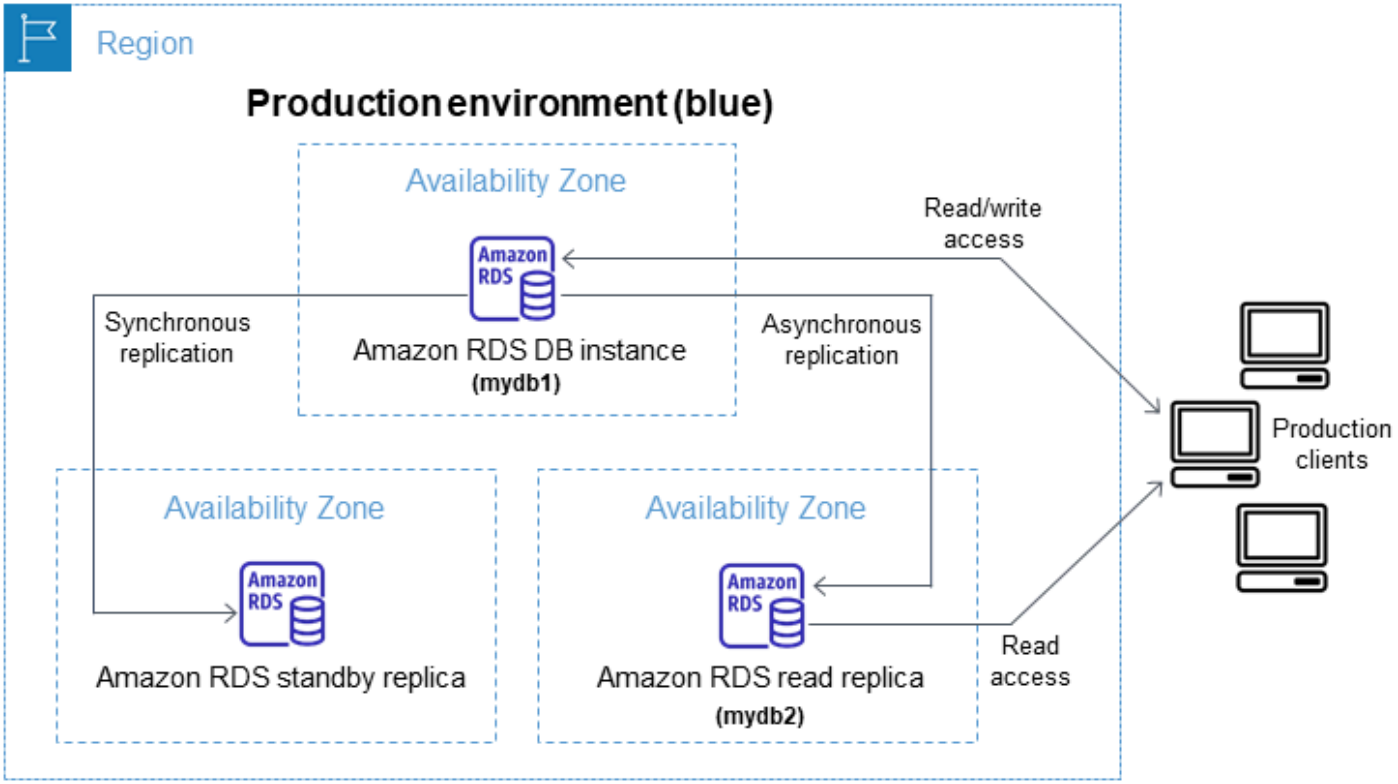
- Crea facilmente un ambiente di gestione temporanea pronto per la produzione.
- Replica automaticamente le modifiche del database dall'ambiente di produzione all'ambiente di gestione temporanea.
- Esegui il test delle modifiche del database in un ambiente di gestione temporanea sicuro, senza influire sull'ambiente di produzione.
- Rimani aggiornato con le patch del database e gli aggiornamenti di sistema.
- Implementa ed esegui il test delle nuove funzionalità del database.
- Esegui lo switchover dell'ambiente di gestione temporanea in un nuovo ambiente di produzione senza modificare l'applicazione.
- Esegui lo switchover in sicurezza usando i guardrail di switchover integrati.
- Elimina la perdita di dati durante lo switchover.
- Esegui lo switchover rapidamente, in genere in meno di un minuto a seconda del carico di lavoro.

Flusso di lavoro di un'implementazione blu/verde

Completa i seguenti passaggi principali quando utilizzi un'implementazione blu/verde per gli aggiornamenti del database.

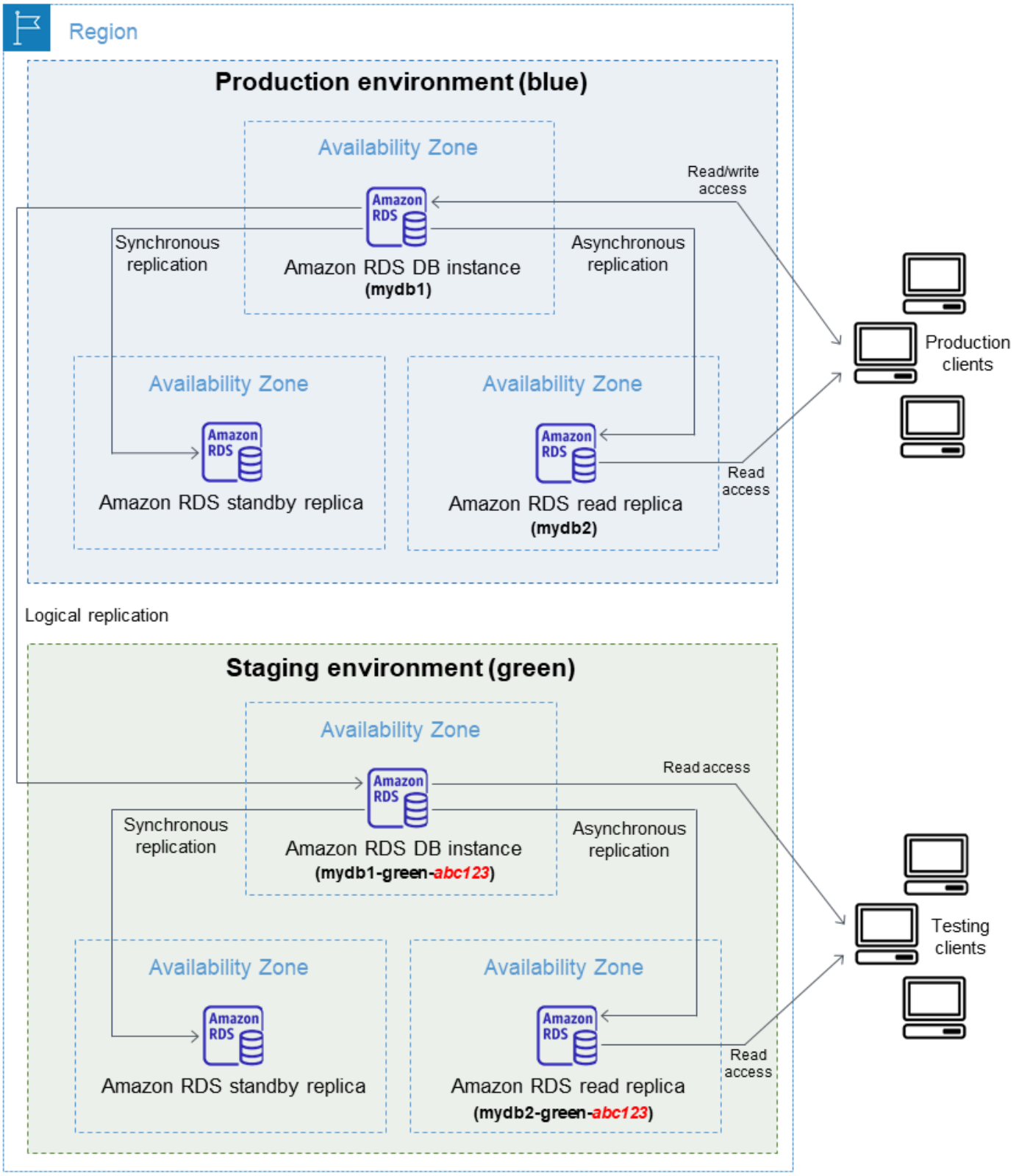
1. Identifica un ambiente di produzione che richieda aggiornamenti.

   Ad esempio, l'ambiente di produzione in questa immagine ha un'implementazione di istanza database multi-AZ (mydb1) e una replica di lettura (mydb2).



2. Crea l'implementazione blu/verde. Per istruzioni, consulta Creazione di un'implementazione blu/verde.

   L'immagine seguente mostra un esempio di implementazione blu/verde dell'ambiente di produzione del passaggio 1. Durante la creazione dell'implementazione blu/verde, RDS copia la topologia e la configurazione complete dell'istanza database primaria per creare l'ambiente verde. I nomi delle istanze database copiati vengono aggiunti con `-green-`*`random-characters`*. L'ambiente di gestione temporanea nell'immagine contiene un'implementazione di istanza database multi-AZ (*`mydb1-green-abc123`*) e una replica di lettura (*`mydb2-green-abc123`*).



Quando crei l'implementazione blu/verde, puoi aggiornare la versione del motore di database e specificare un gruppo di parametri database diverso per le istanze database dell'ambiente verde.

RDS configura anche la replica logica dall'istanza database primaria dell'ambiente blu all'istanza database primaria dell'ambiente verde.

Dopo aver creato l'implementazione blu/verde, l'istanza database dell'ambiente verde è di sola lettura per impostazione predefinita.

3. Se necessario, apporta ulteriori modifiche all'ambiente di gestione temporanea.

   Ad esempio, è possibile apportare modifiche allo schema del database o modificare la classe dell'istanza database utilizzata da una o più istanze database nell'ambiente verde.

   Per ulteriori informazioni sulla modifica di un'istanza database, consulta Modifica di un'istanza database Amazon RDS.

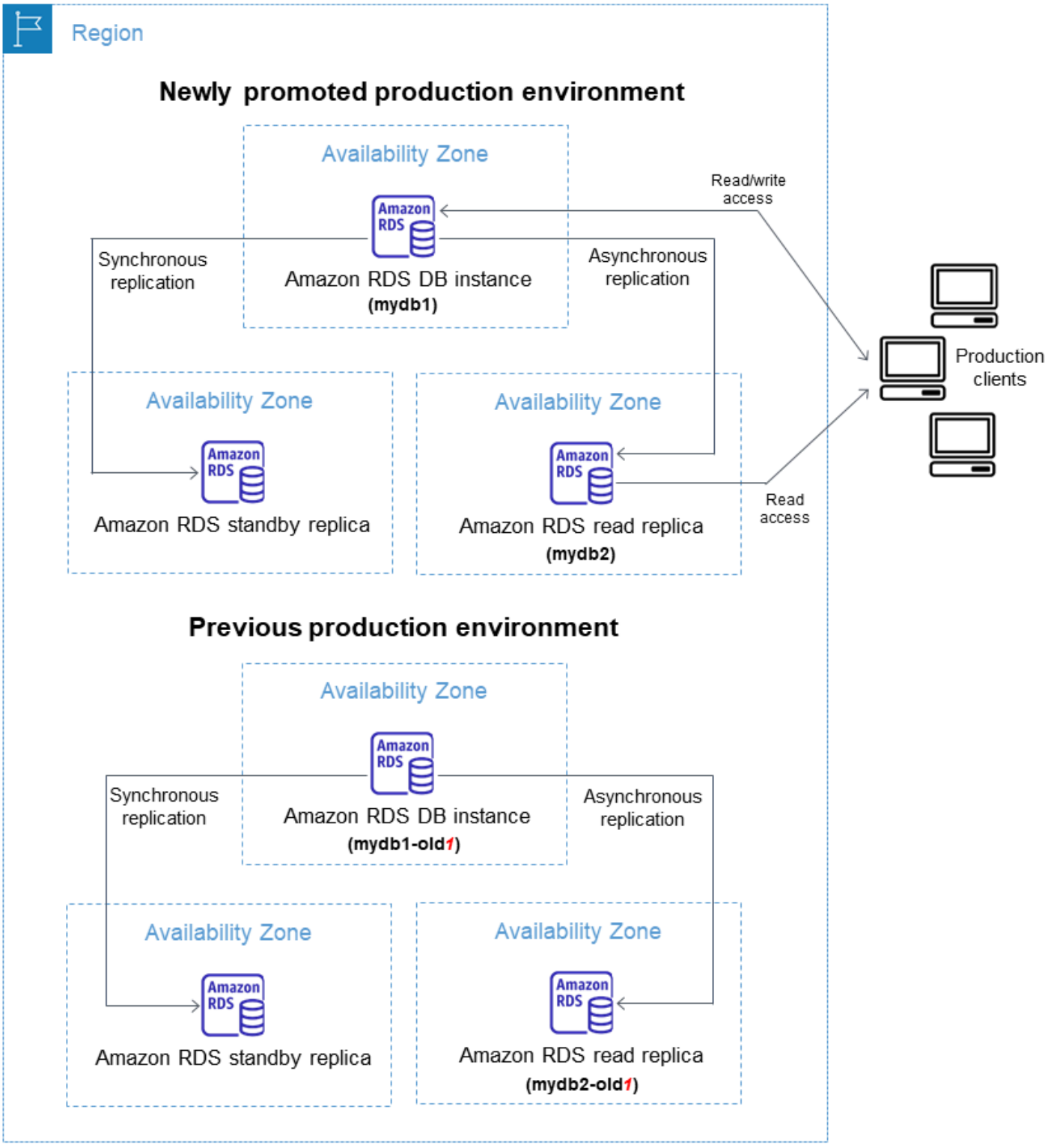
4. Esegui il test dell'ambiente di gestione temporanea.

   Durante i test, ti consigliamo di mantenere i database in un ambiente verde di sola lettura. Abilita le operazioni di scrittura sull'ambiente verde con cautela perché possono causare conflitti di replica. Possono inoltre generare dati non previsti nei database di produzione dopo lo switchover. Per abilitare le operazioni di scrittura per RDS for MySQL, imposta `read_only` il parametro su, quindi riavvia l'`0`istanza DB. Per RDS per PostgreSQL, imposta il parametro su `default_transaction_read_only` a livello di sessione. `off`

5. Quando sei pronto, esegui lo switchover in modo che l'ambiente di gestione temporanea diventi il nuovo ambiente di produzione. Per istruzioni, consulta Switchover di un'implementazione blu/verde.

   Lo switchover comporta tempi di inattività. I tempi di inattività sono in genere inferiori al minuto, ma possono essere più lunghi a seconda del carico di lavoro.

   L'immagine seguente mostra le istanze database dopo lo switchover.



Dopo lo switchover, le istanze database che si trovavano nell'ambiente verde diventano le nuove istanze database di produzione. I nomi e gli endpoint dell'ambiente di produzione corrente vengono assegnati all'ambiente di produzione appena promosso e non richiedono modifiche

all'applicazione. Di conseguenza, il traffico di produzione ora viene indirizzato al nuovo ambiente di produzione. Le istanze database nell'ambiente blu precedente vengono rinominate aggiungendo `-old`*n* al nome corrente, dove *n* è un numero. Ad esempio, supponi che il nome dell'istanza database nell'ambiente blu sia `mydb1`. Dopo lo switchover, il nome dell'istanza database diventa `mydb1-old1`.

Nell'esempio dell'immagine, durante lo switchover si verificano le seguenti modifiche:

- L'implementazione dell'istanza database multi-AZ dell'ambiente verde denominata `mydb1-green-abc123` diventa l'implementazione dell'istanza database multi-AZ di produzione denominata `mydb1`.
- La replica di lettura dell'ambiente verde denominata `mydb2-green-abc123` diventa la replica di lettura di produzione `mydb2`.
- L'implementazione dell'istanza database multi-AZ denominata `mydb1` diventa `mydb1-old1`.
- La replica di lettura dell'ambiente blu denominata `mydb2` diventa `mydb2-old1`.

6. Se non hai più bisogno di un'implementazione blu/verde, puoi eliminarla. Per istruzioni, consulta Eliminazione di un'implementazione blu/verde.

   Dopo lo switchover, l'ambiente di produzione precedente non viene eliminato, quindi è possibile utilizzarlo per i test di regressione, se necessario.

Autorizzazione di accesso alle operazioni dell'implementazione blu/verde

Gli utenti devono disporre delle autorizzazioni necessarie per eseguire operazioni relative alle implementazioni blu/verde. Puoi creare le policy IAM che concedono a utenti e ruoli l'autorizzazione per eseguire operazioni API specifiche sulle risorse indicate di cui hanno bisogno. Puoi quindi collegare tali policy ai ruoli o ai set di autorizzazioni IAM che richiedono le autorizzazioni. Per ulteriori informazioni, consulta Gestione accessi e identità per Amazon RDS.

L'utente che crea un'implementazione blu/verde deve disporre delle autorizzazioni per eseguire le seguenti operazioni RDS:

- `rds:AddTagsToResource`
- `rds:CreateDBInstanceReadReplica`

L'utente che esegue lo switchover a un'implementazione blu/verde deve disporre delle autorizzazioni per eseguire le seguenti operazioni RDS:

- `rds:ModifyDBInstance`
- `rds:PromoteReadReplica`

L'utente che elimina un'implementazione blu/verde deve disporre delle autorizzazioni per eseguire le seguenti operazioni :

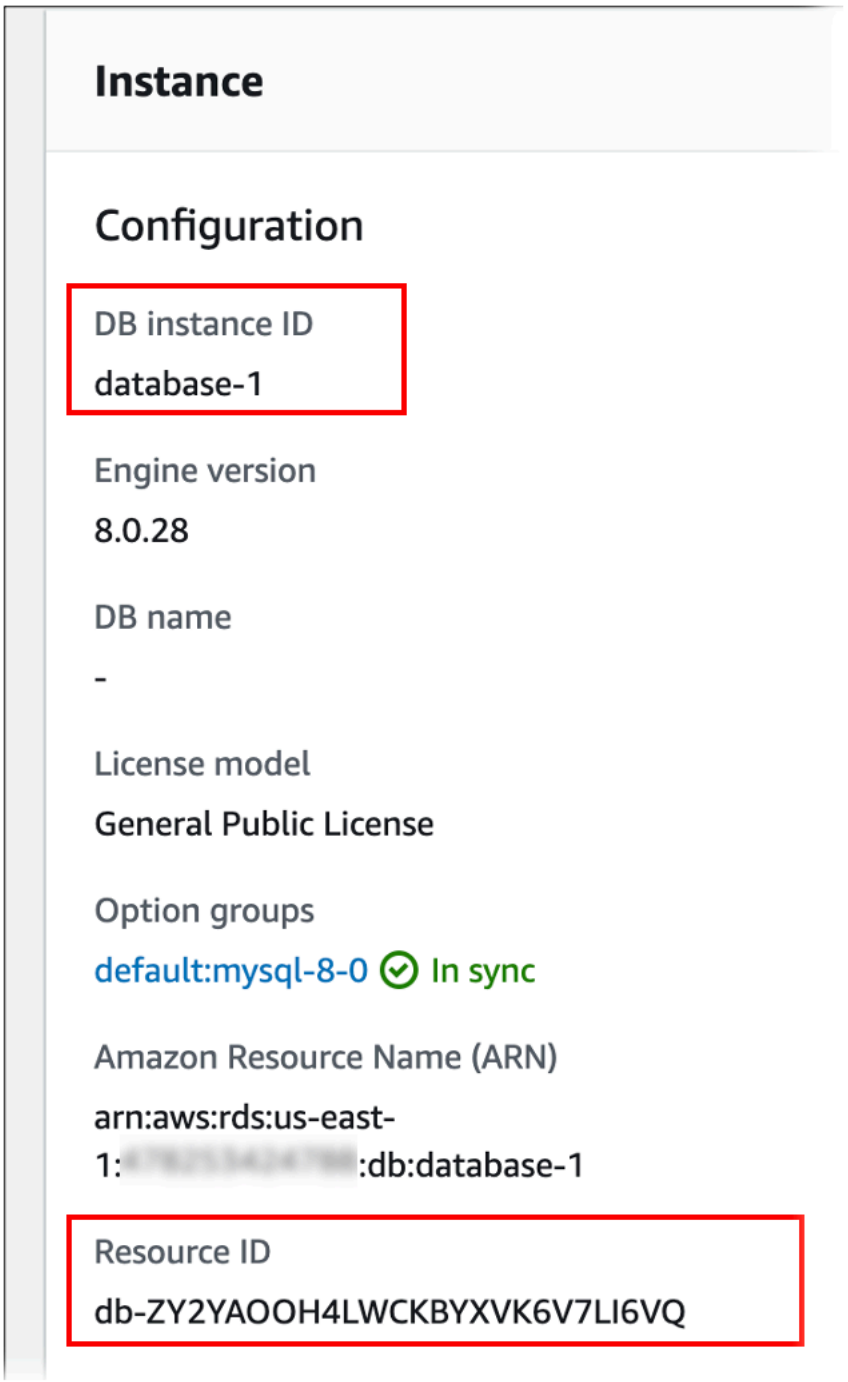
- `rds:DeleteDBInstance`

Amazon RDS il provisioning e modifica le risorse nell'ambiente di staging per tuo conto. Queste risorse includono istanze DB che utilizzano una convenzione di denominazione definita internamente. Pertanto, le policy IAM allegate non possono contenere modelli di nomi di risorse parziali come. `my-db-prefix-*` Sono supportati solo i caratteri jolly (*). In generale, consigliamo di utilizzare i tag delle risorse e altri attributi supportati per controllare l'accesso a queste risorse, anziché i caratteri jolly. Per ulteriori informazioni, consulta Azioni, risorse e chiavi di condizione per Amazon RDS.

Considerazioni sulle implementazioni blu/verde

Amazon RDS traccia le risorse nelle implementazioni blu/verde con `DbiResourceId` di ciascuna risorsa. Questo ID di risorsa è un Regione AWS identificatore univoco e immutabile per la risorsa.

L'ID della risorsa è separato dall'ID dell'istanza database:



Il nome (ID istanza) di una risorsa viene modificato quando effettui lo switchover a un'implementazione blu/verde, ma ogni risorsa mantiene lo stesso ID risorsa. Ad esempio, un identificatore di istanza database potrebbe essere `mydb` nell'ambiente blu. Dopo lo switchover, la stessa istanza database potrebbe essere rinominata in `mydb-old1`. Tuttavia, l'ID risorsa dell'istanza database non viene modificato durante lo switchover. Pertanto, quando le risorse verdi vengono

promosse come nuove risorse di produzione, i relativi ID risorsa non corrispondono agli ID delle risorse blu che erano precedentemente in produzione.

Dopo lo switchover a un'implementazione blu/verde, è consigliabile aggiornare gli ID risorsa con quelli delle risorse di produzione appena promosse per funzionalità e servizi integrati utilizzati con le risorse di produzione. In particolare, considera i seguenti aggiornamenti:

- Se applichi il filtro utilizzando l'API RDS e gli ID risorsa, modifica gli ID risorsa utilizzati nel filtro dopo lo switchover.
- Se lo utilizzi CloudTrail per il controllo delle risorse, imposta i consumatori di in modo che tengano traccia dei nuovi ID delle risorse CloudTrail dopo lo switchover. Per ulteriori informazioni, consulta Monitoraggio delle chamate API di Amazon RDS in AWS CloudTrail.
- Se utilizzi l'API di Approfondimenti sulle prestazioni, modifica gli ID risorsa nelle chiamate all'API dopo lo switchover. Per ulteriori informazioni, consulta Monitoraggio del carico DB con Performance Insights su Amazon RDS.

  Dopo lo switchover è possibile monitorare un database con lo stesso nome, ma non contiene i dati precedenti allo lo switchover.
- Se utilizzi gli ID risorsa nelle policy IAM, assicurati di aggiungere gli ID delle risorse appena promosse quando necessario. Per ulteriori informazioni, consulta Gestione accessi e identità per Amazon RDS.
- Se hai ruoli IAM associati all'istanza DB del , assicurati di riassociarli dopo lo switchover. I ruoli allegati non vengono copiati automaticamente nell'ambiente verde.
- Se si esegue l'autenticazione nell'istanza database utilizzando l'autenticazione del database IAM, assicurarsi che la policy IAM utilizzata per l'accesso al database contenga sia i database blu che quelli verdi elencati sotto l'elemento `Resource` della policy. Ciò è necessario per connettersi al database verde dopo il passaggio. Per ulteriori informazioni, consulta the section called “Creazione e utilizzo di una policy IAM per l'accesso al database IAM”.
- Se lo utilizzi AWS Backup per gestire i backup automatici delle risorse in una distribuzione blu/verde, modifica gli ID delle risorse utilizzati da dopo lo switchover. AWS Backup Per ulteriori informazioni, consulta Utilizzo AWS Backup per gestire i backup automatici.
- Se desideri ripristinare uno snapshot di database manuale o automatico per un'istanza database che faceva parte di un'implementazione blu/verde, assicurati di ripristinare lo snapshot di database corretto esaminando l'ora in cui è stato creato. Per ulteriori informazioni, consulta Ripristino da uno snapshot database.

- Se desideri descrivere un backup automatico di un'istanza database dell'ambiente blu precedente o ripristinarlo a un determinato momento, utilizza l'ID risorsa per l'operazione.

  Poiché il nome dell'istanza database viene modificato durante lo switchover, non è possibile utilizzare il nome precedente per le operazioni `DescribeDBInstanceAutomatedBackups` o `RestoreDBInstanceToPointInTime`.

  Per ulteriori informazioni, consulta Ripristino a un'ora specifica per un'istanza database.
- Quando si aggiunge una replica di lettura a un'istanza database nell'ambiente verde di un'implementazione blu/verde, la nuova replica di lettura non sostituisce una replica di lettura dell'ambiente blu al momento dello switchover. Tuttavia, la nuova replica di lettura viene mantenuta nel nuovo ambiente di produzione dopo lo switchover.
- Quando si elimina un'istanza database nell'ambiente verde di un'implementazione blu/verde, non è possibile creare una nuova istanza database per sostituirla nell'implementazione blu/verde.

  Se crei una nuova istanza database con lo stesso nome e nome della risorsa Amazon (ARN) dell'istanza database eliminata, ha un `DbiResourceId` diverso e quindi non fa parte dell'ambiente verde.

  Se elimini un'istanza database nell'ambiente verde, si verifica il seguente comportamento:

  - Se l'istanza database con lo stesso nome esiste nell'ambiente blu, non verrà eseguito lo switchover all'istanza database dell'ambiente verde. Questa istanza database non verrà rinominata aggiungendo `-old`*n* al suo nome.
  - Qualsiasi applicazione che punti all'istanza database nell'ambiente blu continua a utilizzare la stessa istanza database dopo lo switchover.

  Lo stesso comportamento si applica alle istanze database e alle repliche di lettura.

Best practice per le implementazioni blu/verde

Di seguito sono elencate le best practice per le implementazioni blu/verde:

Best practice generali

- Esegui accuratamente il test delle istanze database nell'ambiente verde prima di effettuare lo switchover.

- Mantieni i tuoi database nell'ambiente verde di sola lettura. Si consiglia di abilitare le operazioni di scrittura nell'ambiente verde con cautela perché possono causare conflitti di replica nell'ambiente verde. Possono inoltre generare dati non previsti nei database di produzione dopo lo switchover.
- Quando si utilizza un'implementazione blu/verde per implementare le modifiche dello schema, applica solo modifiche compatibili con la replica.

  Ad esempio, è possibile aggiungere nuove colonne alla fine di una tabella senza interrompere la replica dalla distribuzione blu a quella verde. Tuttavia, le modifiche dello schema, come la ridenominazione delle colonne o delle tabelle, interrompono la replica nell'implementazione verde.

  Per ulteriori informazioni sulle modifiche compatibili con la replica, consulta Replication with Differing Table Definitions on Source and Replica nella documentazione MySQL e Restrictions nella documentazione della replica logica PostgreSQL.
- Dopo aver creato l'implementazione blu/verde, gestisci il caricamento lento, se necessario. Assicurati che il caricamento dei dati sia completato prima di effettuare lo switchover. Per ulteriori informazioni, consulta Gestione del caricamento lento quando si crea un'implementazione blu/verde.
- Quando si effettua lo switchover in una implementazione blu/verde, attenersi alle best practice relative allo switchover. Per ulteriori informazioni, consulta the section called “Best practice per lo switchover”.

Best practice di RDS per MySQL

- Evita di utilizzare motori di archiviazione non transazionali, come MyISAM, che non sono ottimizzati per la replica.
- Ottimizza le repliche di lettura per la replica di log binari.

  Ad esempio, se la versione del motore di database lo supporta, prendi in considerazione l'utilizzo della replica GTID, della replica parallela e della replica protetta da arresto anomalo nell'ambiente di produzione prima di implementare l'implementazione blu/verde. Queste opzioni promuovono la coerenza e la durabilità dei dati prima dello switchover all'implementazione blu/verde. Per ulteriori informazioni sulla replica GTID per le repliche di lettura, consulta Utilizzo della replica basata su GTID.

Best practice di RDS per PostgreSQL

- Se il database dispone di memoria disponibile sufficiente, aumentate il valore del parametro `logical_decoding_work_mem` DB nell'ambiente blu. In questo modo si riduce la decodifica su disco e si utilizza invece la memoria. È possibile monitorare la memoria liberabile con la `FreeableMemory` CloudWatch metrica. Per ulteriori informazioni, consulta the section called “Parametri a CloudWatch livello di istanza Amazon per Amazon RDS”.
- Aggiorna tutte le estensioni di PostgreSQL all'ultima versione prima di creare un'implementazione blu/verde. Per ulteriori informazioni, consulta the section called “Aggiornamento estensioni PostgreSQL”.
- Se utilizzi l'estensione `aws_s3`, assicurati di concedere all'istanza database l'accesso ad Amazon S3 tramite un ruolo IAM dopo la creazione dell'ambiente verde. In tal modo i comandi di importazione ed esportazione continuano a funzionare dopo lo switchover. Per istruzioni, consulta the section called “Configurazione dell'accesso a un bucket Amazon S3”.
- Se specifichi una versione del motore superiore per l'ambiente verde, esegui l'`ANALYZE`operazione su tutti i database per aggiornare la tabella. `pg_statistic` Le statistiche di Optimizer non vengono trasferite durante l'aggiornamento di una versione principale, quindi è necessario rigenerare tutte le statistiche per evitare problemi di prestazioni. Per ulteriori procedure consigliate durante gli aggiornamenti delle versioni principali, consulta. the section called “Come eseguire l'aggiornamento a una versione principale”
- Evita di configurare i trigger `ENABLE ALWAYS` se `ENABLE REPLICA` o se il trigger viene utilizzato sulla fonte per manipolare i dati. In caso contrario, il sistema di replica propaga le modifiche ed esegue il trigger, che porta alla duplicazione.
- Le transazioni di lunga durata possono causare un notevole ritardo nella replica. Per ridurre il ritardo di replica, prova a fare quanto segue:
  - Riduci le transazioni di lunga durata che possono essere ritardate fino a quando l'ambiente verde non raggiunge il livello dell'ambiente blu.
  - Avvia un'operazione manuale di congelamento sottovuoto su tavoli occupati prima di creare la distribuzione blu/verde.
  - Per PostgreSQL versione 12 e successive, disabilita `index_cleanup` il parametro su tabelle di grandi dimensioni o occupate per aumentare la frequenza di manutenzione normale sui database blu. Per ulteriori informazioni, consulta the section called “Vacuum di una tabella il più rapidamente possibile”.
- Una replica lenta può causare il riavvio frequente di mittenti e destinatari, il che ritarda la sincronizzazione. Per assicurarvi che rimangano attivi, disattivate i timeout impostando il

`wal_sender_timeout` parametro su `0` nell'ambiente blu e il parametro su nell'ambiente verde. `wal_receiver_timeout` `0`

- Per evitare che i segmenti write-ahead log (WAL) vengano rimossi dall'ambiente blu, imposta il `wal_keep_segments` parametro su 15625 per PostgreSQL versione 13 e precedenti. Per la versione 14 e successive, imposta il `wal_keep_size` parametro su 1 TiB, se c'è abbastanza spazio di archiviazione libero.

Limitazioni per le implementazioni blu/verde

Le seguenti limitazioni si applicano alle implementazioni blu/verde.

Argomenti

- Limitazioni generali per le implementazioni blu/verde
- Limitazioni dell'estensione PostgreSQL per le distribuzioni blu/green
- Limitazioni per le modifiche nelle implementazioni blu/verde
- Limitazioni della replica logica di PostgreSQL per le implementazioni blu/verde

Limitazioni generali per le implementazioni blu/verde

Le seguenti limitazioni generali si applicano alle implementazioni blu/verde:

- A causa di un bug di community, le versioni da 8.0.11 a 8.0.13 di MySQL non sono supportate per le implementazioni blu/verde.
- Le seguenti versioni di RDS per PostgreSQL sono supportate come versioni di origine e di destinazione dell'aggiornamento: 11.21 e versioni successive, 12.16 e versioni successive, 13.12 e versioni successive, 14.9 e versioni successive e 15.4 e versioni successive. Per le versioni precedenti, puoi eseguire un aggiornamento della versione secondaria a una supportata.
- Le distribuzioni blu/verde non supportano la gestione delle password degli utenti principali con. AWS Secrets Manager
- Per RDS per PostgreSQL, le tabelle non registrate non vengono replicate nell'ambiente verde.
- Per , il di istanze DB con ambiente blu non può essere una sorgente logica autogestita (editore) o una replica (sottoscrittore).
- Durante il passaggio, gli ambienti blu e verdi non possono avere integrazioni Zero-ETL con Amazon Redshift. Occorre prima eliminare l'integrazione ed eseguire il passaggio, quindi ricreare l'integrazione.

- Il pianificatore eventi (parametro `event_scheduler`) deve essere disabilitato nell'ambiente verde quando si crea un'implementazione blu/verde. Ciò impedisce che si generino eventi nell'ambiente verde e conseguentemente incongruenze.
- Le distribuzioni blu/green non supportano il driver AWS JDBC per MySQL. Per ulteriori informazioni, consulta Limitazioni note su. GitHub
- Le implementazioni blu/verde non sono supportate per le seguenti funzionalità:
  - Server proxy per Amazon RDS
  - Repliche di lettura a cascata
  - Repliche di lettura tra regioni diverse
  - AWS CloudFormation
  - Implementazioni di cluster DB Multi-AZ

    Le implementazioni blu/verde sono supportate per le implementazioni dell'istanza database Multi-AZ. Per ulteriori informazioni sulle implementazioni Multi-AZ, consulta Configurazione e gestione di un'implementazione multi-AZ.

Limitazioni dell'estensione PostgreSQL per le distribuzioni blu/green

Alle estensioni PostgreSQL si applicano le seguenti limitazioni:

- L'estensione `pg_partman` deve essere disabilitata nell'ambiente blu quando si crea un'implementazione blu/verde. L'estensione esegue operazioni DDL come `CREATE TABLE` che interrompono la replica logica dall'ambiente blu all'ambiente verde.
- L'estensione `pg_cron` deve rimanere disabilitata su tutti i database verdi dopo la creazione dell'implementazione blu/verde. L'estensione dispone di worker in background che vengono eseguiti come superutente e aggirano l'impostazione di sola lettura dell'ambiente verde, il che potrebbe causare conflitti di replica.
- Se l'istanza database blu è configurata come server esterno di un'estensione FDW (Foreign Data Wrapper), è necessario utilizzare il nome dell'endpoint dell'istanza anziché gli indirizzi IP. Ciò consente alla configurazione di rimanere funzionale dopo lo switchover.
- Le estensioni `pglogical` e `pg_active` devono essere disabilitate nell'ambiente blu quando si crea un'implementazione blu/verde. Dopo aver promosso l'ambiente verde come nuovo ambiente di produzione, puoi abilitare nuovamente le estensioni. Inoltre, il database blu non può essere un abbonato logico di un'istanza esterna.

- Se utilizzi l'`pgAudit`estensione, deve rimanere nelle librerie condivise (`shared_preload_libraries`) nei gruppi di parametri DB personalizzati sia per le istanze DB blu che per quelle verdi. Per ulteriori informazioni, consulta the section called “Configurazione dell'estensione pgAudit”.

Limitazioni per le modifiche nelle implementazioni blu/verde

Di seguito sono elencate le limitazioni per le modifiche di un'implementazione blu/verde:

- Non è possibile modificare un'istanza decrittografato in un'istanza crittografato.
- Non è possibile modificare un'istanza crittografato in un'istanza decrittografato.
- Non è possibile modificare un'istanza dell'ambiente blu con una versione successiva del motore rispetto all'instanza dell'ambiente verde.
- Le risorse nell'ambiente blu e nell'ambiente verde devono trovarsi nello stesso Account AWS.
- Per RDS per MySQL, se il database di origine è associato a un gruppo di opzioni personalizzato, non è possibile specificare un aggiornamento della versione principale quando si crea l'implementazione blu/verde.

  In tal caso, è possibile creare un'implementazione blu/verde senza specificare un aggiornamento della versione principale. Quindi, puoi aggiornare il database nell'ambiente verde. Per ulteriori informazioni, consulta Aggiornamento della versione del motore di un'istanza database.

Limitazioni della replica logica di PostgreSQL per le implementazioni blu/verde

Le implementazioni blu/verde utilizzano la replica logica per mantenere l'ambiente di gestione temporanea sincronizzato con l'ambiente di produzione. PostgreSQL presenta alcune restrizioni relative alla replica logica, che si traducono in limitazioni durante la creazione di implementazioni blu/verdi per le istanze database RDS per PostgreSQL.

La tabella seguente descrive le limitazioni della replica logica che si applicano alle implementazioni blu/verde per RDS per PostgreSQL.

Limitazione	Spiegazione
Le istruzioni DDL (Data Definitio n Language),	Se Amazon RDS rileva una modifica DDL nell'ambiente blu, i database verdi entrano nello stato di replica degradata.

Limitazione	Spiegazione
come `CREATE TABLE` e `CREATE SCHEMA`, non vengono replicate dall'ambiente blu nell'ambiente verde.	Ricevi un evento che ti avvisa che le modifiche DDL nell'ambiente blu non possono essere replicate nell'ambiente verde. È necessario eliminare l'implementazione blu/verde e tutti i database verdi, quindi ricrearla. In caso contrario, non sarà possibile eseguire lo switchover dall'implementazione blu/verde.
Le operazioni `NEXTVAL` sugli oggetti della sequenza non sono sincronizzate tra l'ambiente blu e l'ambiente verde.	Durante lo switchover, Amazon RDS incrementa i valori di sequenza nell'ambiente verde in modo che corrispondano a quelli nell'ambiente blu. Se hai migliaia di sequenze, lo switchover può subire un ritardo.
La creazione o la modifica di oggetti di grandi dimensioni nell'ambiente blu non viene replicata nell'ambiente verde.	Se Amazon RDS rileva la creazione o la modifica di oggetti di grandi dimensioni nell'ambiente blu archiviati nella tabella di sistema `pg_largeobject` , i database verdi entrano nello stato di replica degradata. RDS genera un evento che notifica che le modifiche di oggetti di grandi dimensioni nell'ambiente blu non possono essere replicate nell'ambiente verde. È necessario eliminare l'implementazione blu/verde e tutti i database verdi, quindi ricrearla. In caso contrario, non sarà possibile eseguire lo switchover dall'implementazione blu/verde.
Le viste materializzate non vengono aggiornate automaticamente nell'ambiente verde.	L'aggiornamento delle viste materializzate nell'ambiente blu non le aggiorna nell'ambiente verde. Dopo lo switchover, puoi pianificare un aggiornamento delle viste materializzate.

Limitazione	Spiegazione
Le operazioni UPDATE e DELETE non sono consentite nelle tabelle che non dispongono di una chiave primaria.	Prima di creare un'implementazione blu/verde, assicurati che tutte le tabelle nell'istanza database abbiano una chiave primaria.

Per ulteriori informazioni, consulta Restrictions nella documentazione della replica logica di PostgreSQL.

Creazione di un'implementazione blu/verde

Quando si crea un'implementazione blu/verde, si specifica l'istanza database di origine da copiare nell'implementazione. L'istanza scelta è l'istanza database di produzione e diventa l'istanza database primaria nell'ambiente blu. Questa istanza database viene copiata nell'ambiente verde e RDS configura la replica dall'istanza database dell'ambiente blu all'istanza database dell'ambiente verde.

RDS copia la topologia dell'ambiente blu in un'area di gestione temporanea, insieme alle funzionalità configurate. Quando l'istanza database blu ha delle repliche di lettura, queste vengono copiate come repliche di lettura dell'istanza database verde nell'implementazione. Se l'istanza database blu è un'implementazione di istanza database multi-AZ, l'istanza database verde viene creata come un'implementazione di istanza database multi-AZ.

Argomenti

- Preparazione di una implementazione blu/verde
- Specifica delle modifiche durante la creazione di un'implementazione blu/verde
- Gestione del caricamento lento quando si crea un'implementazione blu/verde
- Creazione di un'implementazione blu/verde

Preparazione di una implementazione blu/verde

Esistono alcuni passaggi da eseguire prima di creare una distribuzione blu/verde, a seconda del motore su cui è in esecuzione l'istanza database del .

Argomenti

- Preparazione di un'istanza DB RDS for MySQL per una distribuzione blu/verde
- Preparazione di un'istanza database RDS per PostgreSQL per un'implementazione blu/verde

Preparazione di un'istanza DB RDS for MySQL per una distribuzione blu/verde

Prima di creare una distribuzione blu/verde per un'istanza DB RDS for MySQL, è necessario abilitare i backup automatici. Per istruzioni, consulta the section called “Abilitazione dei backup automatici”.

Preparazione di un'istanza database RDS per PostgreSQL per un'implementazione blu/verde

Prima di creare un'implementazione blu/verde per un'istanza database RDS per PostgreSQL, effettua quanto segue:

- Associa l'istanza a un gruppo di parametri di database personalizzato con la replica logica (`rds.logical_replication`) attivata. La replica logica è necessaria per la replica dall'ambiente blu nell'ambiente verde. Per istruzioni, consulta the section called “Modifica di parametri in un gruppo di parametri del database”.

  Poiché le implementazioni blu/verdi richiedono almeno un lavoratore in background per database, assicuratevi di ottimizzare le seguenti impostazioni di configurazione in base al carico di lavoro. Per istruzioni su come ottimizzare ogni impostazione, consulta Impostazioni di configurazione nella documentazione di PostgreSQL.

  - `max_replication_slots`
  - `max_wal_senders`
  - `max_logical_replication_workers`
  - `max_worker_processes`

  Dopo aver abilitato la replica logica e impostato tutte le opzioni di configurazione, assicurati di riavviare l'istanza database per rendere effettive le modifiche apportate. Affinché la creazione di implementazioni blu/verde abbia esito positivo, l'istanza database deve essere sincronizzata con il

gruppo di parametri di database. Per ulteriori informazioni, consulta the section called “Riavvio di un'istanza database”.

- Assicurati che l'istanza database stia eseguendo una versione di RDS per PostgreSQL compatibile con le implementazioni blu/verde di RDS. Per l'elenco delle versioni compatibili, consulta the section called “Distribuzioni blu/verdi”.
- Verifica che l'istanza database non sia l'origine o la destinazione della replica esterna. Per ulteriori informazioni, consulta the section called “Limitazioni generali”.
- Assicurati che tutte le tabelle dell'istanza database abbiano una chiave primaria. La replica logica di PostgreSQL non consente operazioni `UPDATE` o `DELETE` su tabelle che non dispongono di una chiave primaria.
- Se utilizzi i trigger, assicurati che non interferiscano con la creazione, l'aggiornamento e l'eliminazione di `pg_catalog.pg_publication` `pg_catalog.pg_replication_slots` oggetti i cui nomi iniziano con «`pg_catalog.pg_subscription`rds».

Specifica delle modifiche durante la creazione di un'implementazione blu/verde

È possibile apportare le seguenti modifiche all'istanza database nell'ambiente verde quando si crea l'implementazione blu/verde:

È possibile apportare altre modifiche all'istanza database nell'ambiente verde dopo l'implementazione. Ad esempio, è possibile apportare modifiche allo schema del database o modificare la classe dell'istanza database utilizzata da una o più istanze database nell'ambiente verde.

Per ulteriori informazioni sulla modifica di un'istanza database, consulta Modifica di un'istanza database Amazon RDS.

Specifica di una versione successiva del motore

È possibile specificare una versione superiore del motore se si desidera testare un aggiornamento del motore di database. Al momento dello switchover, il database viene aggiornato alla versione principale o secondaria specificata del motore di database.

Specifica di un gruppo di parametri di database

È possibile verificare in che modo le modifiche ai parametri influiscono sulle istanze database nell'ambiente verde o specificare un gruppo di parametri per una nuova versione principale del motore di database in caso di aggiornamento.

Se si specifica un gruppo di parametri database diverso, il gruppo specificato viene associato a tutte le istanze database nell'ambiente verde. Se non si specifica un gruppo di parametri diverso, ogni istanza database nell'ambiente verde viene associata al gruppo di parametri della corrispondente istanza database blu.

Abilitazione di Scritture ottimizzate per RDS

È possibile utilizzare le implementazioni blu/verde per eseguire l'aggiornamento a una classe di istanza database che supporti Scritture ottimizzate per RDS. È possibile abilitare Scritture ottimizzate per RDS solo su un database creato con una classe di istanza database supportata. Pertanto, questa opzione crea un database verde con una classe di istanza database supportata che consente di attivare Scritture ottimizzate per RDS sull'istanza database verde.

Se si esegue l'aggiornamento da una classe di istanza database che non supporta Scritture ottimizzate per RDS a una che lo supporta, è necessario anche aggiornare la configurazione di archiviazione dell'istanza database verde. Per ulteriori informazioni, consulta the section called “Aggiornamento della configurazione di archiviazione”.

È possibile aggiornare solo la classe dell'istanza database verde primaria. Per impostazione predefinita, le repliche di lettura nell'ambiente verde ereditano le impostazioni dell'istanza database dall'ambiente blu. Dopo aver creato l'ambiente verde, è necessario modificare manualmente la classe di istanza database delle repliche di lettura nell'ambiente verde.

A seconda della versione del motore e della classe dell'istanza database blu, alcuni aggiornamenti della classe di istanza non sono supportati. Per altre informazioni sulle classi di istanza database, consulta the section called “Classi di istanze database”.

Aggiornamento della configurazione di archiviazione

Se il database blu non utilizza la configurazione di archiviazione più recente, RDS può migrare l'istanza database verde dalla configurazione di archiviazione precedente (file system a 32 bit) alla configurazione preferita. Puoi utilizzare le implementazioni blu/verde RDS per superare le limitazioni di dimensionamento relative all'archiviazione e alle dimensioni dei file per i file system a 32 bit precedenti. Inoltre, questa impostazione modifica la configurazione di archiviazione per renderla

compatibile con Scritture ottimizzate per RDS se la classe di istanza database specificata supporta questa funzionalità.

> **Note**
>
> L'aggiornamento della configurazione di archiviazione è un'operazione che richiede un elevato livello di I/O e comporta tempi di creazione lunghi per le implementazioni blu/verdi. Il processo di aggiornamento di archiviazione è più rapido se l'istanza database blu utilizza l'archiviazione su SSD con capacità di IOPS allocata (io1) e se è stato eseguito il provisioning dell'ambiente verde con un'istanza di dimensioni almeno 4 volte superiori. Gli aggiornamenti dell'archiviazione su volumi SSD per scopi generici (gp2) possono far esaurire il saldo dei crediti di I/O; se ciò si verifica, il processo di aggiornamento diventa più lungo. Per ulteriori informazioni, consulta the section called “Storage delle istanze database”.
> Durante il processo di aggiornamento dell'archiviazione, il motore di database non è disponibile. Se il consumo di archiviazione sull'istanza database blu è maggiore o uguale al 90% della dimensione dell'archiviazione allocata, quest'ultima verrà aumentata del 10% durante il processo di aggiornamento dell'archiviazione per l'istanza verde.

Questa opzione è disponibile solo se il database blu non utilizza la configurazione di archiviazione più recente o se stai modificando la classe dell'istanza database nell'ambito della stessa richiesta.

Gestione del caricamento lento quando si crea un'implementazione blu/verde

Quando crei un'implementazione blu/verde, Amazon RDS crea l'istanza database primaria nell'ambiente verde eseguendo il ripristino da uno snapshot di database. Dopo la creazione, l'istanza database verde continua a caricare i dati in background, operazione nota come caricamento lento. Se l'istanza database ha le repliche di lettura, anche queste vengono create da snapshot di database e sono soggette al caricamento lento.

Se accedi a dati che non sono ancora stati caricati, l’istanza del cluster di database scarica immediatamente i dati richiesti da Amazon S3 e continua a caricare il resto dei dati in background. Per ulteriori informazioni, consulta Snapshot Amazon EBS.

Per ridurre gli effetti del caricamento lento sulle tabelle a cui è necessario accedere rapidamente, è possibile eseguire operazioni che comportano scansioni di tabelle complete, ad esempio `SELECT *`. Questa operazione consente ad Amazon RDS di scaricare tutti i dati della tabella di backup da S3.

Se un'applicazione tenta di accedere a dati non caricati, si può riscontrare una latenza maggiore del normale durante il caricamento dei dati. Questa maggiore latenza dovuta al caricamento lento potrebbe portare a prestazioni scadenti per i carichi di lavoro sensibili alla latenza.

> **Important**
>
> Se passi a un'implementazione blu/verde prima che il caricamento dei dati sia completato, l'applicazione potrebbe riscontrare problemi di prestazioni dovuti all'elevata latenza.

Creazione di un'implementazione blu/verde

Puoi creare una distribuzione blu/verde utilizzando l'API AWS Management Console, the AWS CLI o RDS.

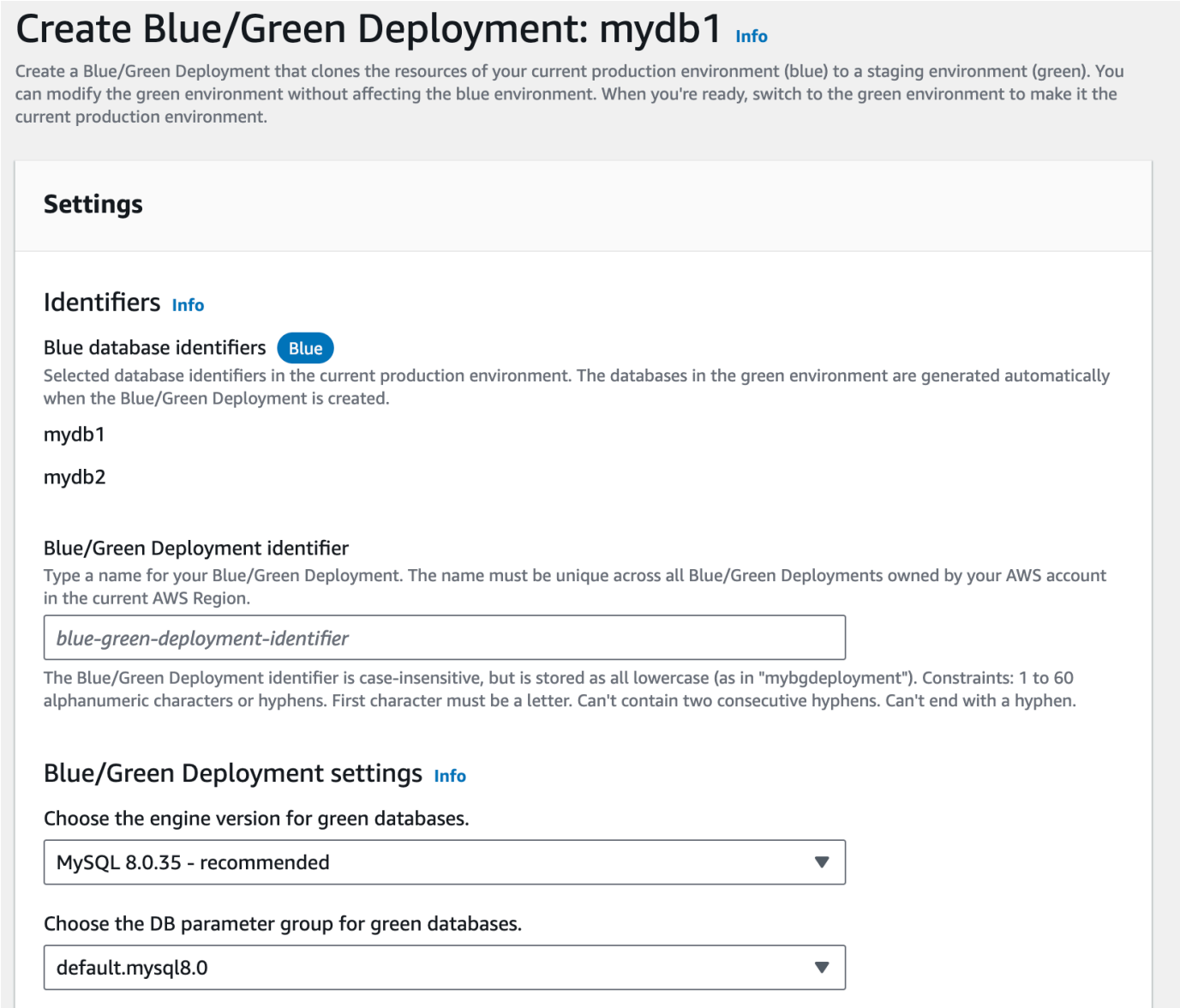
Console

Per creare un'implementazione blu/verde

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, scegli Databases (Database) quindi seleziona l'istanza da copiare nell'ambiente verde.
3. Scegli Azioni, crea una distribuzione blue/verde.

   Se scegli un'istanza database RDS per PostgreSQL, esamina e verifica i limiti della replica logica. Per ulteriori informazioni, consulta the section called “Limitazioni della replica logica di PostgreSQL”.

   Viene visualizzata la pagina Create Blue/Green Deployment (Crea implementazione blu/verde).



4. Controlla gli identificatori blu del database. Assicurati che corrispondano alle istanze DB che ti aspetti nell'ambiente blu. In caso contrario, scegli Cancel (Annulla).
5. Per Blue/Green Deployment identifier (Identificatore implementazione blu/verde), immetti un nome per l'implementazione blu/verde.
6. (Facoltativo) Per Blue/Green Deployment settings (Impostazioni di implementazione blu/verde), specifica le impostazioni per l'ambiente verde:

   - Scegli una versione del motore di database se desideri testare un aggiornamento della versione del motore di database.
   - Scegli un gruppo di parametri database da associare alle istanze database dell'ambiente verde.

È possibile apportare altre modifiche ai database nell'ambiente verde dopo che è stato implementato.

7. (Facoltativo) Per Scritture ottimizzate per RDS abilita questa funzionalità aggiornando la classe dell'istanza database verde primaria. Per ulteriori informazioni, consulta the section called “Abilitazione di Scritture ottimizzate per RDS”.

   Se stai passando da una classe di istanza database che non supporta Scritture ottimizzate a una che lo supporta, devi anche eseguire un aggiornamento della configurazione dell'archiviazione. Per i dettagli consulta la fase successiva.

8. (Facoltativo) Per Aggiornamento della configurazione dell'archiviazione scegli se aggiornare la configurazione del file system di archiviazione. Se abiliti questa opzione, RDS esegue la migrazione dell'istanza database verde dal file system di archiviazione precedente alla configurazione preferita. Per ulteriori informazioni, consulta the section called “Aggiornamento del file system di archiviazione”.

   Questa opzione è disponibile solo se il database blu non utilizza la configurazione di archiviazione più recente o se stai abilitando Scritture ottimizzate per RDS all'interno della stessa richiesta.

9. Scegli Crea ambiente di staging.

AWS CLI

Per creare una distribuzione blu/verde utilizzando il AWS CLI, utilizzate il create-blue-green-deploymentcomando con le seguenti opzioni:

- `--blue-green-deployment-name`: specifica il nome dell'implementazione blu/verde.
- `--source`: specifica il nome della risorsa Amazon (ARN) dell'istanza da copiare.
- `--target-engine-version`: specifica una versione del motore se vuoi testare un aggiornamento della versione del motore di database in un ambiente verde. Questa opzione aggiorna le istanze nell'ambiente verde alla versione del motore di database specificata.

  Se non specificata, ogni istanza database nell'ambiente verde viene creata con la stessa versione del motore dell'istanza database corrispondente nell'ambiente blu.
- `--target-db-parameter-group-name`: specifica un gruppo di parametri database che desideri associare alle istanze database nell'ambiente verde.

- `--target-db-instance-class`: specifica una classe di istanza database che supporti Scritture ottimizzate per RDS. Questa opzione abilita Scritture ottimizzate per RDS sull'istanza database verde primaria. Per ulteriori informazioni, consulta the section called “Abilitazione di Scritture ottimizzate per RDS”.
- `--upgrade-target-storage-config`: specifica se aggiornare la configurazione del file system di archiviazione sul database verde. È possibile abilitare questa opzione solo se il valore dell'opzione `is-storage-config-upgrade-available` è `true` per l'istanza database o se si modifica il valore dell'opzione `target-db-instance-class` nella stessa richiesta. Per ulteriori informazioni, consulta the section called “Aggiornamento del file system di archiviazione”.

Example

PerLinux, omacOS: Unix

```
aws rds create-blue-green-deployment \
    --blue-green-deployment-name my-blue-green-deployment \
    --source arn:aws:rds:us-east-2:123456789012:db:mydb1 \
    --target-engine-version 8.0.31 \
    --target-db-parameter-group-name mydbparametergroup \
    --target-db-instance-class db.m5.8xlarge \
    --upgrade-target-storage-config
```

Per Windows:

```
aws rds create-blue-green-deployment ^
    --blue-green-deployment-name my-blue-green-deployment ^
    --source arn:aws:rds:us-east-2:123456789012:db:mydb1 ^
    --target-engine-version 8.0.31 ^
    --target-db-parameter-group-name mydbparametergroup ^
    --target-db-instance-class db.m5.8xlarge ^
    --upgrade-target-storage-config
```

API RDS

Per creare un'implementazione blu/verde con l'API Amazon RDS, utilizza l'operazione `CreateBlueGreenDeployment` con i seguenti parametri:

- `BlueGreenDeploymentName`: specifica il nome dell'implementazione blu/verde.
- `Source`: specifica il nome della risorsa Amazon (ARN) dell'istanza da copiare nell'ambiente verde.

- `TargetEngineVersion`: specifica una versione del motore se vuoi testare un aggiornamento della versione del motore di database in un ambiente verde. Questa opzione aggiorna le istanze nell'ambiente verde alla versione del motore di database specificata.

  Se non specificata, ogni istanza database nell'ambiente verde viene creata con la stessa versione del motore dell'istanza database corrispondente nell'ambiente blu.
- `TargetDBParameterGroupName`: specifica un gruppo di parametri database che desideri associare alle istanze database nell'ambiente verde.
- `TargetDBInstanceClass`: specifica una classe di istanza database che supporti Scritture ottimizzate per RDS. Questa opzione abilita Scritture ottimizzate per RDS sull'istanza database verde primaria. Per ulteriori informazioni, consulta the section called “Abilitazione di Scritture ottimizzate per RDS”.
- `UpgradeTargetStorageConfig`: specifica se aggiornare la configurazione del file system di archiviazione sul database verde. È possibile abilitare questa opzione solo se il valore dell'opzione `is-storage-config-upgrade-available` è `true` per l'istanza database o se si modifica il valore dell'opzione `target-db-instance-class` nella stessa richiesta. Per ulteriori informazioni, consulta the section called “Aggiornamento del file system di archiviazione”.

Visualizzazione di un'implementazione blu/verde

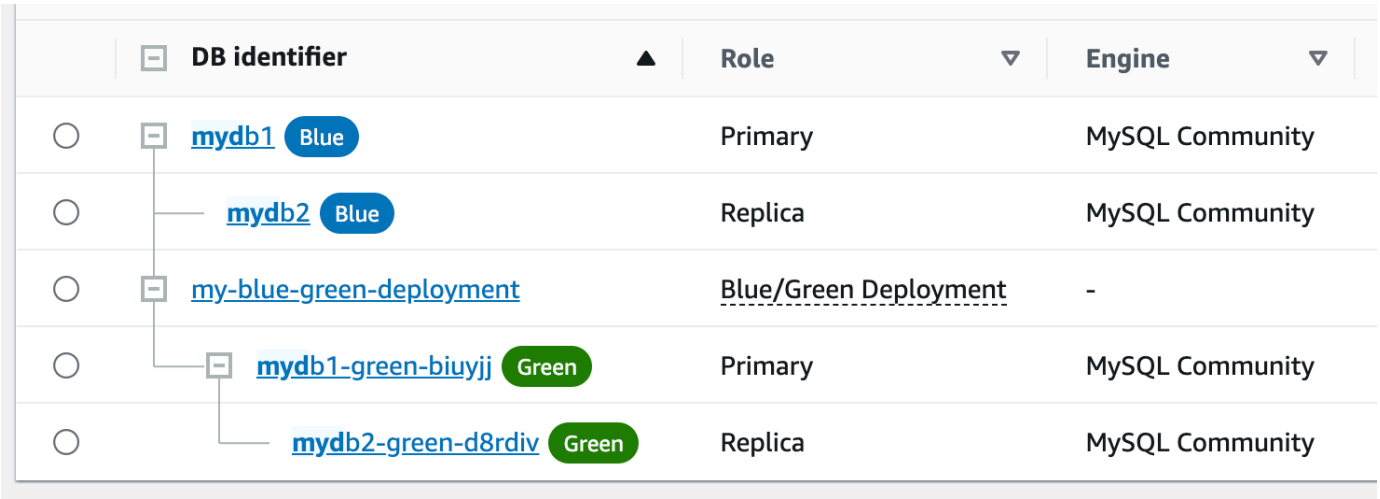
È possibile visualizzare i dettagli di un'implementazione blu/verde utilizzando la AWS Management Console, AWS CLI o l'API RDS.

È anche possibile visualizzare e sottoscrivere gli eventi per informazioni su un'implementazione blu/verde. Per ulteriori informazioni, consulta Eventi di implementazione blu/verde.

Console

Per visualizzare i dettagli di un'implementazione blu/verde

1. Accedi alla AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, scegli Databases (Database), quindi trova l'implementazione blu/verde nell'elenco.

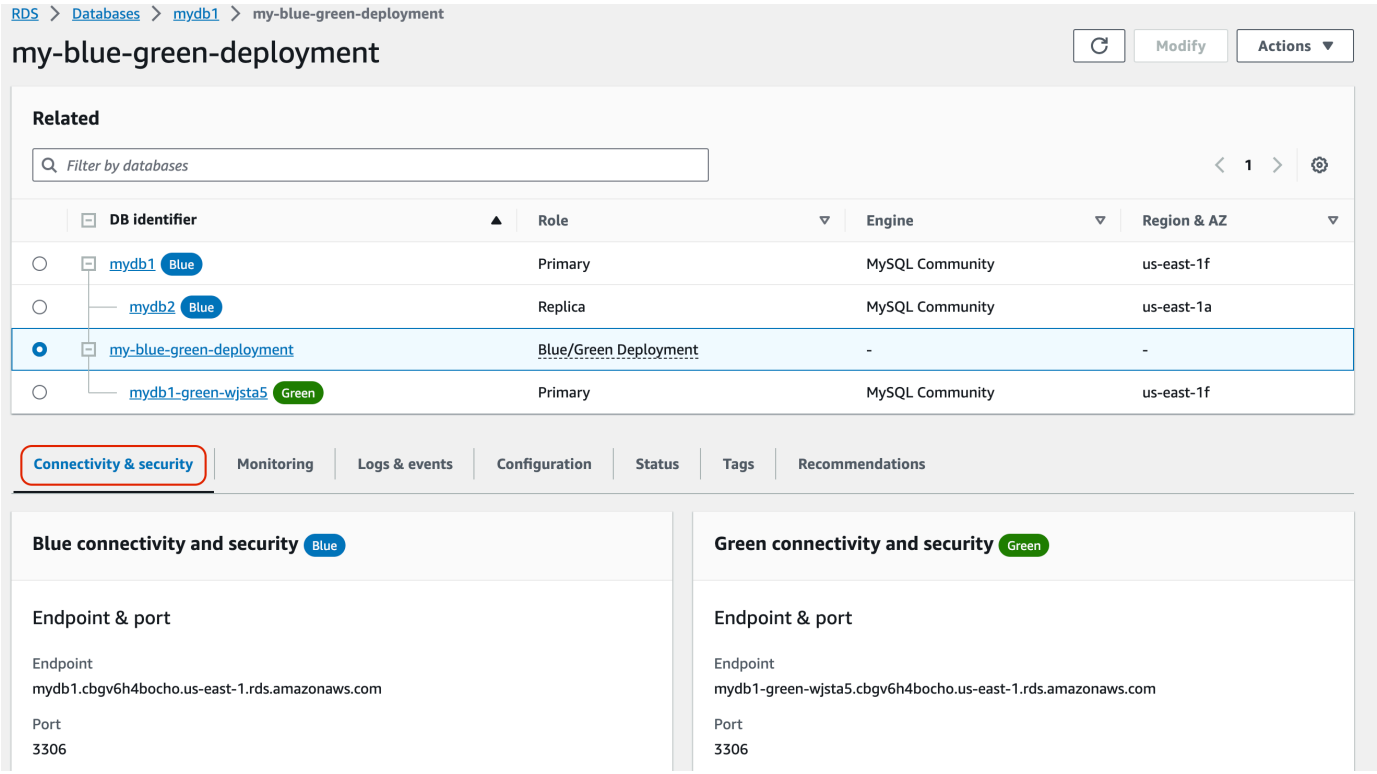


Il valore Role (Ruolo) per l'implementazione blu/verde è Blue/Green Deployment (Implementazione blu/verde).

3. Scegli il nome dell'implementazione blu/verde di cui desideri visualizzarne i dettagli.

   Ogni scheda ha una sezione per l'implementazione blu e una sezione per l'implementazione verde. Ad esempio, nella scheda Configurazione, la versione del motore DB potrebbe essere diversa nell'ambiente blu e nell'ambiente verde se si sta aggiornando la versione del motore DB nell'ambiente verde.

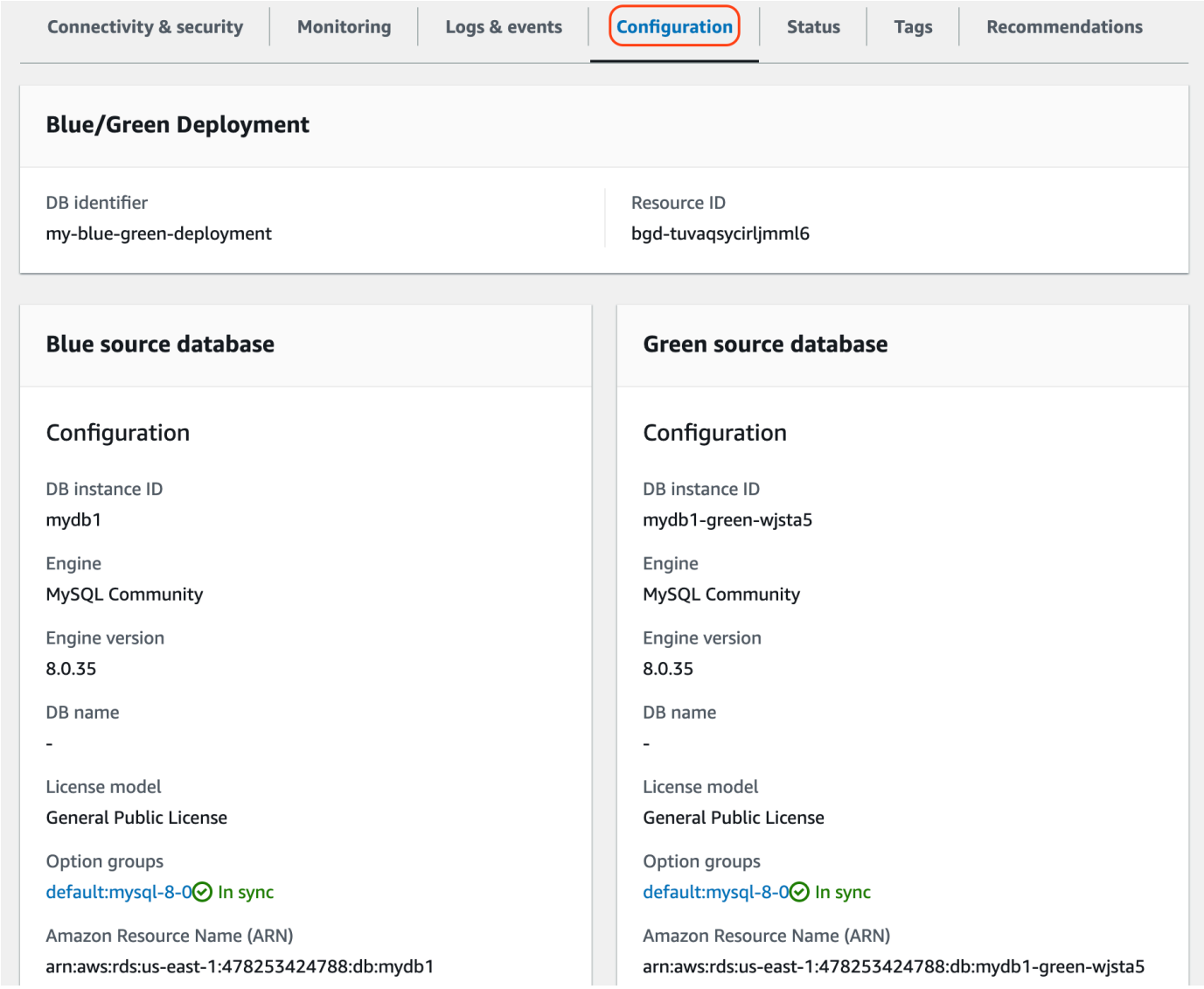
   L'immagine seguente mostra un esempio della scheda Connettività e sicurezza:



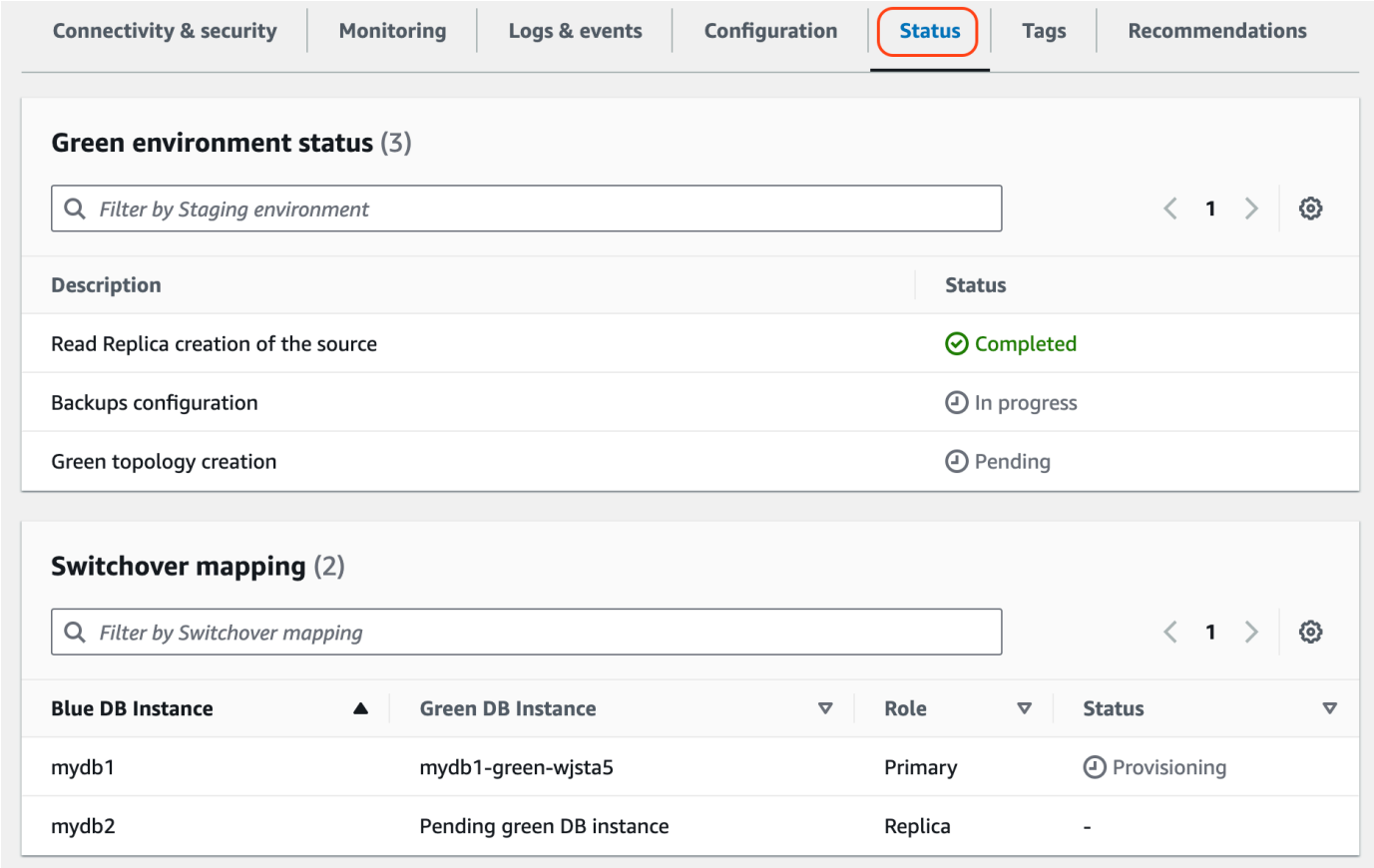
La scheda Connettività e sicurezza include anche una sezione denominata Replica che mostra lo stato attuale della replica logica e il ritardo della replica tra gli ambienti blu e verde. Se lo stato di replica è `Replicating`, l'implementazione blu/verde viene replicata correttamente.

Per le implementazioni blu/verde RDS per PostgreSQL, lo stato di replica può cambiare in `Replication degraded` se si apportano modifiche DDL non supportate o a oggetti di grandi dimensioni nell'ambiente blu. Per ulteriori informazioni, consulta the section called “Limitazioni della replica logica di PostgreSQL”.

L'immagine seguente mostra un esempio della scheda Configurazione:



L'immagine seguente mostra un esempio della scheda Stato:



AWS CLI

Per visualizzare i dettagli su una distribuzione blu/verde utilizzando ilAWS CLI, utilizzare il describe-blue-green-deploymentscomando.

Example Visualizzazione dei dettagli di un'implementazione blu/verde filtrando per nome

Quando si utilizza il describe-blue-green-deploymentscomando, è possibile filtrare in base a. `--blue-green-deployment-name` L'esempio seguente mostra i dettagli per un'implementazione blu/verde denominata *`my-blue-green-deployment`*.

```
aws rds describe-blue-green-deployments --filters Name=blue-green-deployment-
name,Values=my-blue-green-deployment
```

Example Visualizzazione dei dettagli di un'implementazione blu/verde specificando l'identificatore

Quando si utilizza il describe-blue-green-deploymentscomando, è possibile specificare il`--blue-green-deployment-identifier`. L'esempio seguente mostra i dettagli per un'implementazione blu/verde con l'identificatore *`bgd-1234567890abcdef`*.

```
aws rds describe-blue-green-deployments --blue-green-deployment-
identifier bgd-1234567890abcdef
```

API RDS

Per visualizzare i dettagli su un'implementazione blu/verde utilizzando l'API Amazon RDS, utilizza l'operazione `DescribeBlueGreenDeployments` e specifica `BlueGreenDeploymentIdentifier`.

Switchover di un'implementazione blu/verde

Lo switchover rende l'ambiente verde il nuovo ambiente di produzione. Se l'istanza database verde include repliche di lettura, anche queste vengono promosse. Prima dello switchover, il traffico di produzione viene indirizzato all'istanza database e alle repliche di lettura nell'ambiente blu. Dopo lo switchover, il traffico di produzione viene indirizzato all'istanza database e alle repliche di lettura nell'ambiente verde.

Argomenti

Timeout dello switchover

È possibile specificare un timeout per lo switchover compreso tra 30 secondi e 3.600 secondi (un'ora). Se lo switchover richiede più tempo della durata specificata, viene eseguito il rollback di tutte le modifiche e non viene apportata alcuna modifica agli ambienti. L'impostazione predefinita del timeout è 300 secondi (cinque minuti).

Guardrail dello switchover

Quando avvii uno switchover, Amazon RDS esegue alcuni controlli di base per verificare la preparazione degli ambienti blu e verdi per lo switchover. Questi controlli sono noti come guardrail dello switchover e impediscono lo switchover se gli ambienti non sono pronti per farlo. Pertanto, evitano tempi di inattività più lunghi del previsto e impediscono la perdita di dati tra gli ambienti blu e quelli verdi che potrebbe verificarsi se lo switchover venisse avviato.

Amazon RDS esegue i seguenti controlli guardrail sull'ambiente verde:

- Integrità della replica: verifica se lo stato della replica dell'istanza database primaria verde è integro. L'istanza database primaria verde è una replica dell'istanza database primaria blu.
- Ritardo della replica: verifica se il ritardo della replica dell'istanza database primaria verde rientra nei limiti consentiti per lo switchover. I limiti consentiti si basano sul periodo di timeout specificato. Il ritardo della replica indica il ritardo dell'istanza database primaria verde rispetto all'istanza database primaria blu. Per ulteriori informazioni, consulta the section called “Diagnosi e risoluzione del ritardo tra repliche di lettura” per RDS per MySQL e the section called “Monitoraggio e ottimizzazione del processo di replica” per RDS per PostgreSQL.
- Scritture attive: assicura che non vi siano scritture attive nell'istanza database primaria verde.

Amazon RDS esegue i seguenti controlli guardrail sull'ambiente blu:

- Replica esterna: per PostgreSQL RDS per PostgreSQL, assicura che l'ambiente blu non sia una fonte logica autogestita (editore) o una replica (sottoscrittore). In tal caso, si consiglia di eliminare gli slot e gli abbonamenti di replica autogestiti su tutti i database nell'ambiente blu, procedere con il passaggio al digitale e quindi ricrearli per riprendere la replica. Per , assicurati che il database blu non sia una replica binlog esterna.
- Scritture attive di lunga durata: assicura che non vi siano scritture attive di lunga durata nell'istanza database primaria blu perché possono aumentare il ritardo della replica.
- Istruzioni DDL di lunga durata: assicura che non vi siano istruzioni DLL di lunga durata nell'istanza database primaria blu perché possono aumentare il ritardo della replica.
- Modifiche PostgreSQL non supportate: per le istanze database RDS per PostgreSQL, assicura che non siano state eseguite modifiche DDL, aggiunte o modifiche di oggetti di grandi dimensioni nell'ambiente blu. Per ulteriori informazioni, consulta the section called “Limitazioni della replica logica di PostgreSQL”.

Se Amazon RDS rileva modifiche PostgreSQL non supportate, modifica lo stato di replica su `Replication degraded` e ti avvisa che lo switchover non è disponibile per l'implementazione blu/verde. Per procedere con lo switchover, ti consigliamo di eliminare e ricreare l'implementazione blu/verde e tutti i database verdi. A tale scopo, scegli Operazioni, Elimina con database verdi.

Azioni dello switchover

Quando si effettua lo switchover per un'implementazione blu/verde, RDS esegue le seguenti azioni:

1. Esegue controlli guardrail per verificare se gli ambienti blu e verdi sono pronti per lo switchover.
2. Interrompe le nuove operazioni di scrittura nell'istanza database primaria in entrambi gli ambienti.
3. Elimina le connessioni alle istanze database in entrambi gli ambienti e non consente nuove connessioni.
4. Attende che la replica recuperi l'ambiente verde in modo che sia sincronizzato con l'ambiente blu.
5. Rinomina le istanze database in entrambi gli ambienti.

   RDS rinomina le istanze database nell'ambiente verde in modo che corrispondano alle istanze database nell'ambiente blu. Ad esempio, supponi che il nome dell'istanza database nell'ambiente blu sia `mydb`. Supponi anche che il nome dell'istanza database corrispondente nell'ambiente verde sia `mydb-green-abc123`. Durante lo switchover, il nome dell'istanza database nell'ambiente verde viene modificato in `mydb`.

   RDS rinomina le istanze database nell'ambiente blu aggiungendo `-old`*n* al nome corrente, dove *n* è un numero. Ad esempio, supponi che il nome dell'istanza database nell'ambiente blu sia `mydb`. Dopo lo switchover, il nome dell'istanza database diventa `mydb-old1`.

   RDS rinomina anche gli endpoint nell'ambiente verde in modo che corrispondano agli endpoint nell'ambiente blu, per non apportare modifiche all'applicazione.
6. Consente le connessioni ai database in entrambi gli ambienti.
7. Consente le operazioni di scrittura nell'istanza database primaria nel nuovo ambiente di produzione.

   Dopo lo switchover, il precedente principale di produzione consente operazioni di lettura solo fino a quando non si imposta il parametro e si riavvia l'istanza DB. `read_only 0`

Puoi monitorare lo stato di uno switchover utilizzando Amazon. EventBridge Per ulteriori informazioni, consulta the section called “Eventi di implementazione blu/verde”.

Se configurati nell'ambiente blu, i tag vengono spostati nel nuovo ambiente di produzione durante lo switchover. Anche l'ambiente di produzione precedente mantiene questi tag. Per ulteriori informazioni sui tag, consulta Tagging delle risorse Amazon RDS.

Se lo switchover inizia e poi si interrompe prima del termine per un qualsiasi motivo, viene eseguito il rollback di tutte le modifiche e non viene apportata alcuna modifica agli ambienti.

Best practice per lo switchover

Prima di effettuare lo switchover, ti consigliamo vivamente di seguire le best practice completando le seguenti attività:

- Esegui accuratamente il test delle risorse nell'ambiente verde. Assicurati che funzionino correttamente ed efficacemente.
- Monitora le CloudWatch metriche Amazon pertinenti. Per ulteriori informazioni, consulta the section called “Verifica CloudWatch delle metriche prima del passaggio al digitale”.
- Identifica il momento migliore per lo switchover.

  Durante lo switchover, le scritture dei database vengono interrotte in entrambi gli ambienti. Identifica il momento in cui il traffico è più basso nell'ambiente di produzione. Le transazioni di lunga durata, come le DDL attive, possono aumentare i tempi dello switchover, con conseguenti tempi di inattività più lunghi per i carichi di lavoro di produzione.

  Se è presente un numero elevato di connessioni a istanze database, valutare la possibilità di ridurre tale numero manualmente alla quantità minima necessaria per l'applicazione prima di effettuare lo switchover all'implementazione blu/verde. A tale scopo, creare uno script che monitora lo stato dell'implementazione blu/verde e inizia a rimuovere le connessioni quando rileva che lo stato è cambiato in `SWITCHOVER_IN_PROGRESS`.
- Assicurati che le istanze database siano nello stato `Available` in entrambi gli ambienti.
- Assicurati che l'istanza database primaria nell'ambiente verde sia nello stato integro e in grado di replicare.
- Assicurati che le configurazioni di rete e client non aumentino il Time-To-Live (TTL) della cache DNS di oltre cinque secondi, ovvero l'impostazione predefinita per le zone DNS RDS. Altrimenti, le applicazioni continueranno a inviare traffico di scrittura all'ambiente blu dopo lo switchover.

- Assicurati che il caricamento dei dati sia completato prima di effettuare lo switchover. Per ulteriori informazioni, consulta the section called “Gestione del caricamento lento”.
- Per come segue:
  - Esamina i limiti della replica logica e intraprendi le azioni necessarie prima del passaggio. Per ulteriori informazioni, consulta the section called “Limitazioni della replica logica di PostgreSQL”.
  - Eseguire l'operazione `ANALYZE` per aggiornare la tabella `pg_statistics`. Ciò riduce il rischio di problemi di prestazioni dopo il passaggio al digitale.

> **Note**
>
> Durante uno switchover non è possibile modificare le istanze database incluse nello switchover.

Verifica CloudWatch delle metriche prima del passaggio al digitale

Prima di passare a una distribuzione blu/verde, ti consigliamo di controllare i valori delle seguenti metriche all'interno di Amazon. CloudWatch

- `ReplicaLag`: utilizza questo parametro per identificare l'attuale ritardo di replica nell'ambiente verde. Per ridurre i tempi di inattività, assicurati che il valore sia prossimo allo zero prima di effettuare lo switchover.
- `DatabaseConnections`: utilizza questo parametro per stimare il livello di attività dell'implementazione blu/verde e assicurarti che il valore sia a un livello accettabile per la tua implementazione prima dello switchover. Se Approfondimenti sulle prestazioni è attivato, `DBLoad` è un parametro più accurato.

Per ulteriori informazioni su questi parametri, consulta the section called “CloudWatch metriche per RDS”.

Switchover di un'implementazione blu/verde

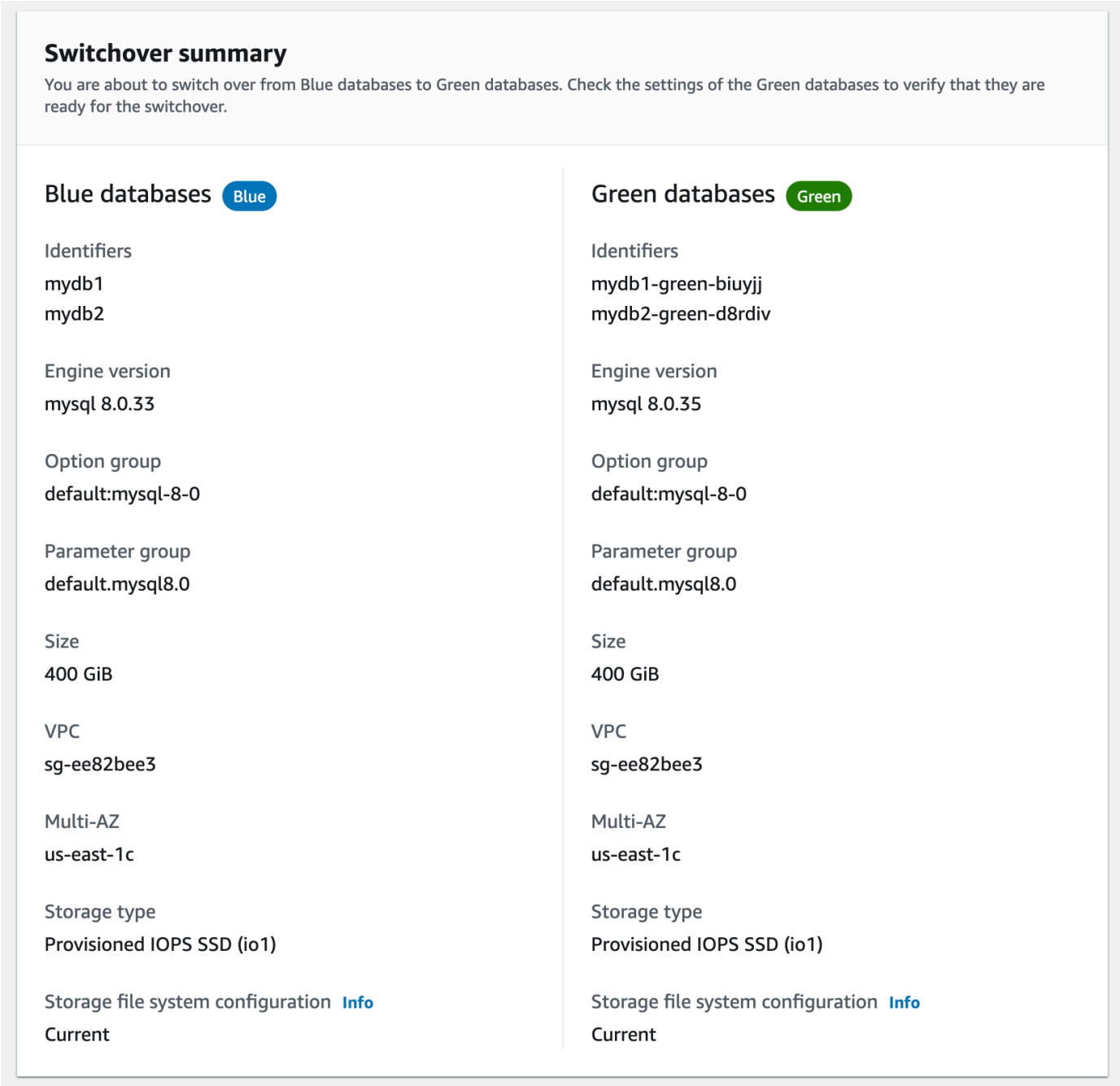
È possibile passare da una distribuzione blu/verde utilizzando l'API AWS Management Console AWS CLI, the o RDS.

Console

Per eseguire lo switchover di un'implementazione blu/verde

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, scegli Databases (Database) e seleziona l'implementazione blu/verde di cui eseguire lo switchover.
3. In Actions (Operazioni), scegli Switch over (Esegui switchover).

   Viene visualizzata la pagina Switch over (Switchover).



4. Nella pagina Switch over (Switchover), consulta il riepilogo dello switchover. Assicurati che le risorse in entrambi gli ambienti corrispondano a quelle previste. In caso contrario, scegli Cancel (Annulla).
5. In Impostazioni di timeout, inserisci il limite di tempo per lo switchover.
6. Se sull'istanza è in esecuzione RDS per PostgreSQL, esamina e verifica i suggerimenti prima dello switchover. Per ulteriori informazioni, consulta the section called “Limitazioni della replica logica di PostgreSQL”.

7. Seleziona Switch over (Switchover).

AWS CLI

Per passare da una distribuzione blu/verde utilizzando il AWS CLI, usa il switchover-blue-green-deploymentcomando con le seguenti opzioni:

- `--blue-green-deployment-identifier`— Specificare l'ID della risorsa della distribuzione blu/verde.
- `--switchover-timeout`: specifica il limite di tempo per lo switchover, in secondi. Il valore predefinito è 300.

Example Switchover di un'implementazione blu/verde

PerLinux, omacOS: Unix

```
aws rds switchover-blue-green-deployment \
    --blue-green-deployment-identifier bgd-1234567890abcdef \
    --switchover-timeout 600
```

Per Windows:

```
aws rds switchover-blue-green-deployment ^
    --blue-green-deployment-identifier bgd-1234567890abcdef ^
    --switchover-timeout 600
```

API RDS

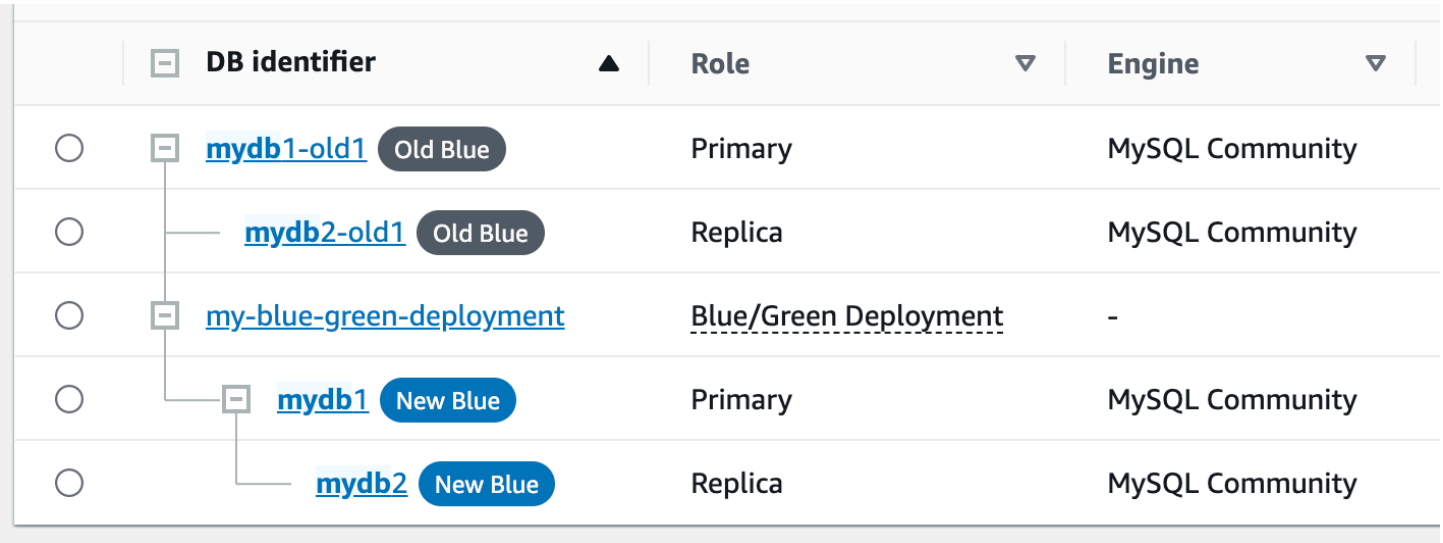
Per eseguire lo switchover di una implementazione blu/verde utilizzando l'API Amazon RDS, usa l'operazione `SwitchoverBlueGreenDeployment` con i seguenti parametri:

- `BlueGreenDeploymentIdentifier`— Specificare l'ID della risorsa della distribuzione blu/verde.
- `SwitchoverTimeout`: specifica il limite di tempo per lo switchover, in secondi. Il valore predefinito è 300.

Dopo lo switchover

Dopo uno switchover, le istanze database vengono mantenute nell'ambiente blu precedente. A queste risorse si applicano i costi standard. La replica tra gli ambienti blu e verde vengono arrestati.

RDS rinomina le istanze database nell'ambiente blu aggiungendo `-old`*n* al nome corrente, dove *n* è un numero. Le istanze database sono di sola lettura finché non si imposta il parametro `read_only` su `0`.



Aggiornamento del nodo principale per i consumatori

Dopo aver cambiato una distribuzione RDS per MariaDB o RDS per MySQL MySQL blu/verde, se il cluster DB di blu aveva repliche esterne o utenti di log binari prima del passaggio, è necessario aggiornare il relativo nodo principale dopo il passaggio per mantenere la continuità della replica.

Dopo il passaggio, l'istanza DB che si trovava precedentemente nell'ambiente verde emette un evento che contiene il nome del file di registro principale e la posizione del registro principale. Per esempio:

```
aws rds describe-events --output json --source-type db-instance --source-identifier db-instance-identifier

{
    "Events": [
...
        {
            "SourceIdentifier": "db-instance-identifier",
```

```
            "SourceType": "db-instance",
            "Message": "Binary log coordinates in green environment after switchover:
             file mysql-bin-changelog.000003 and position 804",
            "EventCategories": [],
            "Date": "2023-11-10T01:33:41.911Z",
            "SourceArn": "arn:aws:rds:us-east-1:123456789012:db:db-instance-identifier"
        }
    ]
}
```

Innanzitutto, assicurati che il consumatore o la replica abbiano applicato tutti i log binari del vecchio ambiente blu. Quindi, utilizza le coordinate del registro binario fornite per riprendere l'applicazione sui consumatori. Ad esempio, se stai eseguendo una replica MySQL su EC2, puoi usare il comando: `CHANGE MASTER TO`

```
CHANGE MASTER TO MASTER_HOST='{new-writer-endpoint}', MASTER_LOG_FILE='mysql-bin-
changelog.000003', MASTER_LOG_POS=804;
```

> **Note**
>
> Se il consumatore è un'altra istanza RDS per MariaDB o RDS per MariaDB DB, puoi eseguire le seguenti stored procedure nell'ordine:,, e. the section called “mysql.rds_stop_replication” the section called “mysql.rds_reset_external_master” the section called “mysql.rds_set_external_master” the section called “mysql.rds_start_replication”

Eliminazione di un'implementazione blu/verde

È possibile eliminare l'implementazione blu/verde prima o dopo lo switchover.

Quando elimini un'implementazione blu/verde prima dello switchover, Amazon RDS elimina facoltativamente le istanze database nell'ambiente verde:

- Se scegli di eliminare le istanze database nell'ambiente verde (`--delete-target`), per tali istanze la protezione dall'eliminazione deve essere disattivata.
- Se non elimini le istanze database nell'ambiente verde (`--no-delete-target`), le istanze vengono mantenute ma non fanno più parte di un'implementazione blu/verde. La replica continua tra gli ambienti.

L'opzione per eliminare i database verdi non è disponibile nella console dopo lo switchover. Quando si eliminano le distribuzioni blu/verdi utilizzando il AWS CLI, non è possibile specificare l'`--delete-target`opzione se lo stato di distribuzione è. `SWITCHOVER_COMPLETED`

> **Important**
>
> L'eliminazione dell'implementazione blu/verde non influisce sull'ambiente blu.

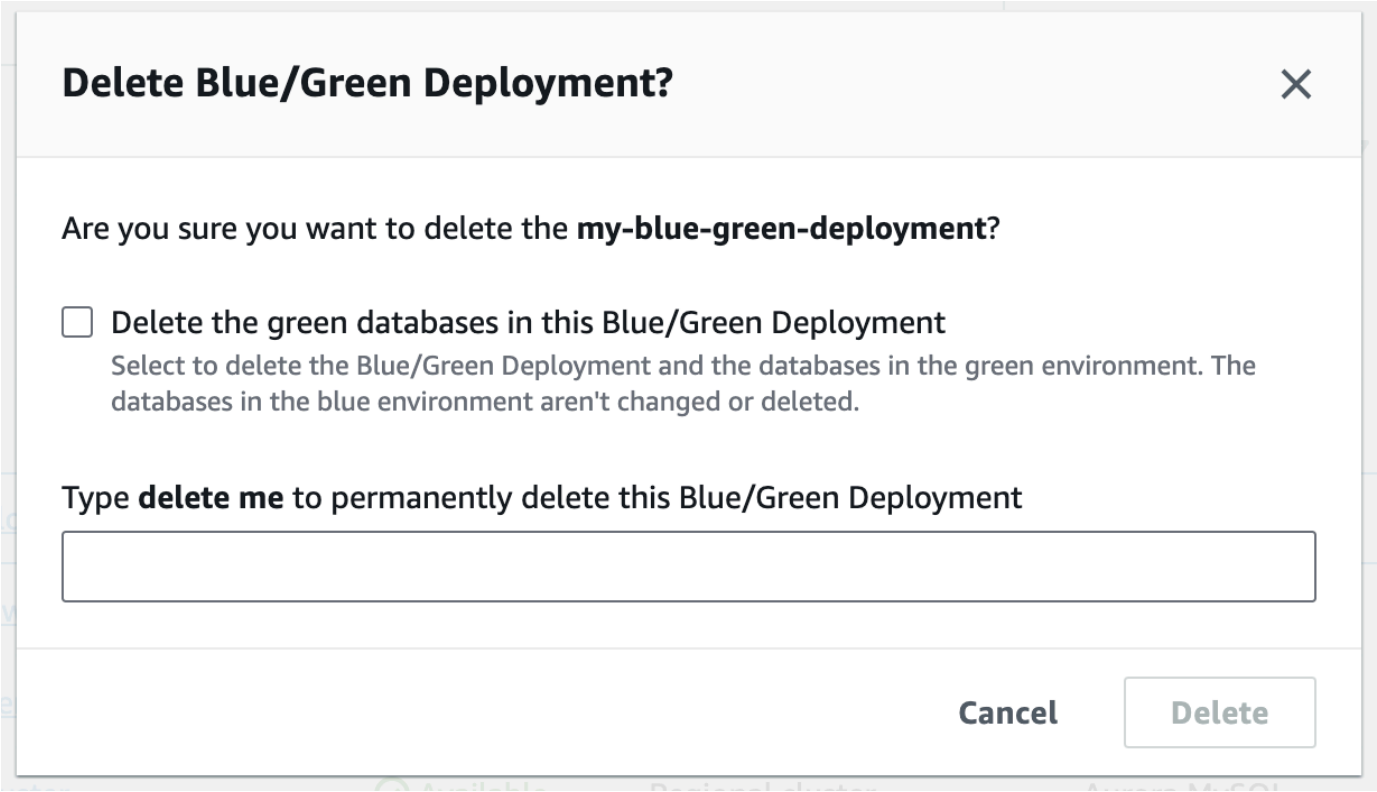
È possibile eliminare una distribuzione blu/verde utilizzando l'API AWS Management Console, the AWS CLI o RDS.

Console

Per eliminare un'implementazione blu/verde

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, scegli Databases (Database) e seleziona l'implementazione blu/verde da eliminare.
3. In Actions (Azioni), scegliere Delete (Elimina).

   Viene visualizzata una finestra Delete Blue/Green Deployment? (Eliminare l'implementazione blu/verde?).



Per eliminare i database verdi, seleziona Delete the green databases in this Blue/Green Deployment (Elimina i database verdi in questa implementazione blu/verde).

4. Immettere **`delete me`** nella casella.
5. Scegliere Delete (Elimina).

AWS CLI

Per eliminare una distribuzione blu/verde utilizzando il AWS CLI, usa il delete-blue-green-deploymentcomando con le seguenti opzioni:

- `--blue-green-deployment-identifier`— L'ID della risorsa della distribuzione blu/verde da eliminare.
- `--delete-target`: specifica che le istanze nell'ambiente verde vengono eliminate. Non è possibile specificare questa opzione se lo stato dell'implementazione blu/verde è `SWITCHOVER_COMPLETED`.
- `--no-delete-target`: specifica che le istanze nell'ambiente verde vengono mantenute.

Example Eliminazione di un'implementazione blu/verde e delle istanze nell'ambiente verde

PerLinux, omacOS: Unix

```
aws rds delete-blue-green-deployment \
    --blue-green-deployment-identifier bgd-1234567890abcdef \
    --delete-target
```

Per Windows:

```
aws rds delete-blue-green-deployment ^
    --blue-green-deployment-identifier bgd-1234567890abcdef ^
    --delete-target
```

Example Eliminazione di un'implementazione blu/verde, mantenendo le istanze nell'ambiente verde

Per LinuxmacOS, oUnix:

```
aws rds delete-blue-green-deployment \
    --blue-green-deployment-identifier bgd-1234567890abcdef \
    --no-delete-target
```

Per Windows:

```
aws rds delete-blue-green-deployment ^
    --blue-green-deployment-identifier bgd-1234567890abcdef ^
    --no-delete-target
```

API RDS

Per eliminare un'implementazione blu/verde con l'API Amazon RDS, utilizza l'operazione `DeleteBlueGreenDeployment` con i seguenti parametri:

- `BlueGreenDeploymentIdentifier`— L'ID della risorsa della distribuzione blu/verde da eliminare.
- `DeleteTarget`: specifica `TRUE` per eliminare le istanze nell'ambiente verde o `FALSE` per mantenerle. Non può essere `TRUE` se lo stato dell'implementazione blu/verde è `SWITCHOVER_COMPLETED`.

Backup, ripristino ed esportazione dei dati

Questa sezione mostra come eseguire il backup, il ripristino e l'esportazione dei dati da un'istanza database Amazon RDS o da un cluster DB Multi-AZ.

Argomenti

Introduzione ai backup

Amazon RDS crea e salva i backup automatici dell'istanza database o del cluster database Multi-AZ durante la finestra di backup dell'istanza database. RDS crea uno snapshot dei volumi di storage dell'istanza database, eseguendo il backup dell'intera istanza database anziché dei singoli database. RDS salva i backup automatici dell'istanza database in base al periodo di retention dei backup specificato. Se necessario, è possibile ripristinare l'istanza database a uno specifico momento durante il periodo di conservazione dei backup.

I backup automatici seguono queste regole:

- La tua istanza database deve essere nello stato `available` per effettuare i backup automatici. I backup automatici non si verificano mentre l'istanza database è in uno stato diverso da `available`, ad esempio, `storage_full`.
- I backup automatici non si verificano quando una copia dello snapshot di database viene eseguita nella stessa Regione AWS per lo stesso database.

Puoi inoltre eseguire il backup dell'istanza database manualmente mediante la creazione di una snapshot DB. Per ulteriori informazioni sulla creazione di uno snapshot di database, consulta Creazione di uno snapshot DB per un'istanza DB Single-AZ.

Il primo snapshot di un'istanza database contiene i dati dell'intero database. Gli snapshot successivi dello stesso database sono incrementali, ovvero vengono salvati solo i dati che sono cambiati dal salvataggio dell'ultimo snapshot.

Puoi copiare le snapshot DB automatiche e manuali e condividere le snapshot DB manuali. Per ulteriori informazioni sulla copia di una snapshot DB, consulta Copia di una snapshot DB.. Per ulteriori informazioni sulla condivisione di una snapshot DB, consulta Condivisione di uno snapshot del database.

Storage di backup

Lo storage di backup Amazon RDS per ciascuna regione Regione AWS è composto da backup automatici e snapshot DB manuali per quella regione. Lo spazio totale di storage di backup è uguale alla somma dello storage di tutti i backup nella regione. Il trasferimento di una snapshot DB in un'altra regione aumenta lo storage di backup nella regione di destinazione. I backup sono archiviati in Amazon S3.

Per ulteriori informazioni sui costi di storage dei backup, consulta Prezzi di Amazon RDS.

Se si sceglie di mantenere i backup automatici quando si elimina un'istanza database, i backup automatici vengono salvati per tutto il periodo di conservazione. Se non scegli Retain automated backups (Mantieni backup automatici) quando elimini un'istanza database, tutti i backup automatici vengono eliminati con l'istanza database. Dopo che sono stati eliminati, i backup automatici non possono essere ripristinati. Se scegli di fare in modo che Amazon RDS crei una snapshot DB finale prima di eliminare l'istanza database, puoi utilizzarla per ripristinare l'istanza database. In alternativa, puoi utilizzare uno snapshot manuale creato in precedenza. Gli snapshot manuali non vengono eliminati. Puoi avere un massimo di 100 snapshot manuali per regione.

Gestione dei backup automatici

Questa sezione mostra come gestire i backup automatici per istanze DB e cluster di database.

Argomenti

Finestra di backup

I backup automatici vengono effettuati quotidianamente durante la finestra di backup scelta. Se il backup richiede più tempo rispetto alla finestra di backup prevista, l'esecuzione continua dopo il termine della finestra finché non viene completata. La finestra di backup non può sovrapporsi con la finestra di manutenzione settimanale per l'istanza database o il cluster di database multi-AZ.

Durante la finestra di backup automatico, le operazioni I/O di storage potrebbero essere sospese brevemente durante l'inizializzazione del processo di backup (in genere per alcuni secondi). Potresti rilevare un aumento della latenza per alcuni minuti durante i backup per le implementazioni Multi-AZ. Per MariaDB, MySQL, Oracle e PostgreSQL l'attività di I/O non viene sospesa nel database principale durante il backup delle implementazioni multi-AZ, perché il backup viene acquisito durante la fase di standby. Per SQL Server, l'attività di I/O viene sospesa brevemente durante il backup delle implementazioni single-AZ e multi-AZ, perché il backup viene acquisito durante il backup principale. Per Db2, l'attività di I/O viene inoltre sospesa brevemente durante il backup, anche se il backup viene eseguito dalla modalità di standby.

I backup automatici possono occasionalmente essere saltati se l'istanza database o il cluster di database ha un carico di lavoro pesante nel momento in cui deve essere avviato un backup. Se un

backup viene saltato, è comunque possibile eseguire un backup point-in-time-recovery (PITR) e il backup viene comunque tentato nella finestra di backup successiva. Per ulteriori informazioni su PITR, consulta Ripristino a un'ora specifica per un'istanza database.

Se non specifichi una finestra di backup al momento della creazione dell'istanza database o del cluster di database multi-AZ, Amazon RDS assegna una finestra di backup predefinita di 30 minuti. Questa finestra viene selezionata a caso da un intervallo di tempo di 8 ore per ciascuna. Regione AWS La tabella seguente elenca i blocchi temporali per ciascuno Regione AWS a cui sono assegnate le finestre di backup predefinite.

Nome della regione	Regione	Periodo di tempo
US East (Ohio)	us-east-2	03:00 - 11:00 UTC
US East (N. Virginia)	us-east-1	03:00 - 11:00 UTC
US West (N. Californi a)	us-west-1	06:00 - 14:00 UTC
US West (Oregon)	us-west-2	06:00 - 14:00 UTC
Africa (Cape Town)	af-south-1	03:00 - 11:00 UTC
Asia Pacific (Hong Kong)	ap-east-1	06:00 - 14:00 UTC
Asia Pacific (Hyderabad)	ap-south-2	06:30 - 14:30 UTC
Asia Pacifico (Giacarta)	ap-southeast-3	08:00–16:00 UTC
Asia Pacifico (Melbourne)	ap-southeast-4	11:00 - 19:00 UTC
Asia Pacific (Mumbai)	ap-south-1	16:30 - 00:30 UTC
Asia Pacific (Osaka)	ap-northeast-3	00:00 - 08:00 UTC
Asia Pacific (Seoul)	ap-northeast-2	13:00 - 21:00 UTC

Nome della regione	Regione	Periodo di tempo
Asia Pacific (Singapore)	ap-southeast-1	14:00 - 22:00 UTC
Asia Pacific (Sydney)	ap-southeast-2	12:00 - 20:00 UTC
Asia Pacific (Tokyo)	ap-northeast-1	13:00 - 21:00 UTC
Canada (Central)	ca-central-1	03:00 - 11:00 UTC
Canada occidentale (Calgary)	ca-west-1	18:00 - 02:00 UTC
China (Beijing)	cn-north-1	06:00 - 14:00 UTC
China (Ningxia)	cn-northwest-1	06:00 - 14:00 UTC
Europe (Frankfurt)	eu-central-1	20:00 - 04:00 UTC
Europe (Ireland)	eu-west-1	22:00 - 06:00 UTC
Europe (London)	eu-west-2	22:00 - 06:00 UTC
Europa (Milano)	eu-south-1	02:00 - 10:00 UTC
Europe (Paris)	eu-west-3	07:29 - 14:29 UTC
Europa (Spagna)	eu-south-2	02:00 - 10:00 UTC
Europe (Stockholm)	eu-north-1	23:00 - 07:00 UTC
Europa (Zurigo)	eu-central-2	02:00 - 10:00 UTC
Israele (Tel Aviv)	il-central-1	03:00 - 11:00 UTC
Medio Oriente (Bahrein)	me-south-1	06:00 - 14:00 UTC
Medio Oriente (Emirati Arabi Uniti)	me-central-1	05:00–13:00 UTC

Nome della regione	Regione	Periodo di tempo
Sud America (São Paulo)	sa-east-1	23:00 - 07:00 UTC
AWS GovCloud (Stati Uniti orientali)	us-gov-east-1	17:00 - 01:00 UTC
AWS GovCloud (Stati Uniti occidentali)	us-gov-west-1	06:00 - 14:00 UTC

Backup retention period (Periodo di retention dei backup)

Puoi impostare il periodo di retention dei backup al momento della creazione di un'istanza database o un cluster di database multi-AZ. Se crei un'istanza DB utilizzando l'API Amazon RDS o il AWS CLI e se non imposti il periodo di conservazione del backup, il periodo di conservazione del backup predefinito è di un giorno. Se crei un'istanza DB utilizzando la console, il periodo di conservazione dei backup predefinito è di sette giorni.

Puoi modificare il periodo di conservazione dei backup dopo la creazione di un'istanza database o un cluster di database. È possibile impostare il periodo di conservazione dei backup per un'istanza database su un valore compreso tra 0 e 35 giorni. Impostando il periodo di retention dei backup su 0, i backup automatici vengono disabilitati. Per un cluster DB Multi-AZ, è possibile impostare il periodo di conservazione dei backup tra 1 e 35 giorni. I limiti degli snapshot manuali (100 per ogni regione) non si applicano ai backup automatici.

I backup automatici non vengono creati mentre un'istanza database o un cluster di database viene arrestato. I backup possono essere conservati più a lungo del periodo di conservazione del backup se un'istanza database è stata arrestata. RDS non include il tempo trascorso nello stato `stopped` quando viene calcolata la finestra di conservazione del backup.

> **Important**
>
> Si verifica un'interruzione se si modifica il periodo di conservazione dei backup di un'istanza DB da 0 a un valore diverso da zero o da un valore diverso da zero a 0.

Abilitazione dei backup automatici

Se i backup automatici non sono abilitati per l'istanza database, puoi abilitarli in qualsiasi momento. Per abilitare i backup automatici, impostare il periodo di conservazione dei backup su un valore diverso da zero positivo. Quando i backup automatici vengono attivati, l'istanza database viene portata offline e un backup viene creato immediatamente.

> **Note**
>
> Se gestisci i backup in AWS Backup, non puoi abilitare i backup automatici. Per ulteriori informazioni, consulta Utilizzo AWS Backup per gestire i backup automatici.

Console

Per abilitare immediatamente i backup automatici

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel pannello di navigazione, scegli Database, quindi scegli l'istanza database o il cluster di database multi-AZ che vuoi modificare.
3. Scegli Modifica.
4. In Periodo di retention dei backup, scegli un valore positivo diverso da zero, ad esempio 3 giorni.
5. Scegli Continue (Continua).
6. Scegliere Apply immediately (Applica immediatamente).
7. Scegli Modifica istanza database o Modifica cluster per salvare le modifiche e abilitare i backup automatici.

AWS CLI

Per abilitare i backup automatici, usa il comando AWS CLI `modify-db-instance`or `modify-db-cluster`.

Includere i seguenti parametri:

- `--db-instance-identifier` (o `--db-cluster-identifier` per un cluster di database multi-AZ)

- `--backup-retention-period`
- `--apply-immediately` o `--no-apply-immediately`

In questo esempio vengono abilitati i backup automatici impostando il periodo di conservazione dei backup su tre giorni. Le modifiche vengono applicate immediatamente.

Example

Per LinuxmacOS, oUnix:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance  \
    --backup-retention-period 3 \
    --apply-immediately
```

Per Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance  ^
    --backup-retention-period 3 ^
    --apply-immediately
```

API RDS

Per abilitare i backup automatici, utilizza l'operazione API RDS `ModifyDBInstance` o `ModifyDBCluster` con i seguenti parametri obbligatori:

- `DBInstanceIdentifier` o `DBClusterIdentifier`
- `BackupRetentionPeriod`

Visualizzazione dei backup automatici

Per visualizzare i backup automatici conservati, scegli Backup automatici nel pannello di navigazione, quindi scegli Mantenuti. Per visualizzare singoli snapshot associati a un backup automatico mantenuto, scegli Snapshot nel pannello di navigazione. In alternativa, puoi descrivere singoli snapshot associati a un backup automatico mantenuto. Da qui, puoi ripristinare un'istanza database direttamente da uno di tali snapshot.

Per descrivere i backup automatici per le istanze DB esistenti utilizzando il AWS CLI, utilizza uno dei seguenti comandi:

```
aws rds describe-db-instance-automated-backups --db-instance-
identifier DBInstanceIdentifier
```

oppure

```
aws rds describe-db-instance-automated-backups --dbi-resource-id DbiResourceId
```

Per descrivere i backup automatici mantenuti tramite l'API RDS, chiama l'operazione `DescribeDBInstanceAutomatedBackups` con uno dei seguenti parametri:

- `DBInstanceIdentifier`
- `DbiResourceId`

Mantenimento dei backup automatici

> **Note**
>
> È possibile mantenere solo i backup automatici delle istanza database, non dei cluster di database multi-AZ.

Puoi scegliere di mantenere i backup automatici quando elimini un'istanza DB. I backup automatici possono essere mantenuti per il numero di giorni configurato per il periodo di conservazione dei backup sull'istanza database al momento della sua eliminazione.

I backup automatici mantenuti contengono snapshot di sistema e log delle transazioni di un'istanza database. Includono anche proprietà dell'istanza database come archiviazione allocata e classe di istanza database, che sono richieste per eseguire il ripristino in un'istanza attiva.

I backup automatici conservati e le istantanee manuali comportano costi di fatturazione fino alla loro eliminazione. Per ulteriori informazioni, consulta Costi di retention.

È possibile conservare i backup automatici per le istanze RDS che eseguono i motori Db2, MariaDB, MySQL, PostgreSQL, Oracle e Microsoft SQL Server.

È possibile ripristinare o rimuovere i backup automatici conservati utilizzando l'API RDS e. AWS Management Console AWS CLI

Argomenti

Periodo di conservazione

Gli snapshot di sistema e i log delle transazioni di un backup automatico mantenuto scadono allo stesso modo dell'istanza database di origine. Poiché non esistono nuovi snapshot o log creati per questa istanza, i backup automatici mantenuti scadono alla fine completamente. Di fatto, continuano a esistere fino al termine del loro ultimo snapshot di sistema, in base alle impostazioni del periodo di retention per l'istanza di origine al momento dell'eliminazione. I backup automatici mantenuti vengono rimossi dal sistema dopo che il loro ultimo snapshot di sistema è scaduto.

Puoi rimuovere un backup automatico mantenuto nello stesso modo con cui elimini un'istanza database. Puoi rimuovere backup automatici mantenuti utilizzando la console o l'operazione API RDS `DeleteDBInstanceAutomatedBackup`.

Gli snapshot finali sono indipendenti dai backup automatici mantenuti. Ti suggeriamo di acquisire uno snapshot finale anche se mantieni i backup automatici in quanto prima o poi scadono. Lo snapshot finale non scade.

Visualizzazione dei backup conservati

Per visualizzare i backup automatici conservati, scegli Backup automatici nel pannello di navigazione, quindi scegli Mantenuti. Per visualizzare singoli snapshot associati a un backup automatico mantenuto, scegli Snapshot nel pannello di navigazione. In alternativa, puoi descrivere singoli snapshot associati a un backup automatico mantenuto. Da qui, puoi ripristinare un'istanza database direttamente da uno di tali snapshot.

Per descrivere i backup automatici conservati utilizzando il AWS CLI, utilizza il comando seguente:

```
aws rds describe-db-instance-automated-backups --dbi-resource-id DbiResourceId
```

Per descrivere i backup automatici mantenuti tramite l'API RDS, chiama l'operazione `DescribeDBInstanceAutomatedBackups` con il parametro `DbiResourceId`.

Ripristino

Per informazioni sul ripristino di istanze database dai backup automatici, consulta Ripristino a un'ora specifica per un'istanza database.

Costi di retention

Il costo di un backup automatico mantenuto è il costo di storage totale degli snapshot di sistema ad esso associati. Non sono previsti costi aggiuntivi per i log delle transazioni o i metadati dell'istanza. Tutte le altre regole di prezzo per i backup si applicano alle istanze ripristinabili.

Ad esempio, supponiamo che lo storage allocato totale di istanze in esecuzione sia 100 GB. Supponiamo anche di avere 50 GB di snapshot manuali più 75 GB di snapshot di sistema associate a un backup automatico mantenuto. In questo caso, vengono addebitati solo i 25 GB aggiuntivi di storage di backup, come riportato di seguito: (50 GB + 75 GB) – 100 GB = 25 GB.

Limitazioni

Le seguenti limitazioni si applicano ai backup automatici mantenuti:

- Il numero massimo di backup automatici conservati in una AWS regione è 40. Non è incluso nella quota di istanze database. Possono esserci contemporaneamente 40 istanze database in esecuzione e 40 backup automatici mantenuti aggiuntivi.
- I backup automatici mantenuti non contengono informazioni relative ai parametri o ai gruppi di opzioni.
- È possibile ripristinare un'istanza eliminata a un punto temporale che si trova all'interno del periodo di conservazione al momento dell'eliminazione.
- Non è possibile modificare un backup automatico conservato. Questo perché è costituito da backup di sistema, log delle transazioni e proprietà dell'istanza database esistenti al momento in cui è stata eliminata l'istanza di origine.

Eliminazione dei backup automatici mantenuti

Puoi eliminare i backup automatici mantenuti quando non servono più.

Console

Per eliminare i backup automatici mantenuti

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, selezionare Automated backups (Backup automatici).
3. Nella scheda Conservato scegli il backup automatico conservato che desideri eliminare.
4. In Actions (Azioni), selezionare Delete (Elimina).
5. Nella pagina di conferma, immetti **`delete me`** e seleziona Elimina.

AWS CLI

Puoi eliminare un backup automatico conservato utilizzando il AWS CLI comando delete-db-instance-automated-backup con la seguente opzione:

- `--dbi-resource-id` – L'identificatore della risorsa per il cluster database.

  È possibile trovare l'identificatore di risorsa per l'istanza DB di origine di un backup automatizzato mantenuto eseguendo il comando -backups. AWS CLI describe-db-instance-automated

Example

Il seguente esempio elimina il backup automatico mantenuto con l'identificatore della risorsa di istanza DB source `db-123ABCEXAMPLE`.

PerLinux, o: macOS Unix

```
aws rds delete-db-instance-automated-backup \
    --dbi-resource-id db-123ABCEXAMPLE
```

Per Windows:

```
aws rds delete-db-instance-automated-backup ^
    --dbi-resource-id db-123ABCEXAMPLE
```

API RDS

Puoi eliminare un backup automatico mantenuto utilizzando l'operazione API Amazon RDS DeleteDB InstanceAutomatedBackup con il seguente parametro:

- `DbiResourceId` – L'identificatore della risorsa per il cluster database.

  Puoi trovare l'identificatore di risorsa per l'istanza DB di origine di un backup automatizzato mantenuto utilizzando l'operazione API Amazon RDS DescribeDB. InstanceAutomatedBackups

Disabilitazione dei backup automatici

In alcuni casi, potrebbe essere necessario disabilitare temporaneamente i backup automatici; ad esempio, durante il caricamento di grandi quantità di dati.

> **Important**
>
> Sconsigliamo vivamente di disabilitare i backup automatici perché disabilita il ripristino. point-in-time La disabilitazione dei backup automatici per un'istanza database o un cluster di database multi-AZ elimina tutti i backup automatici esistenti per il database. Se disattivi e poi riattivi i backup automatici, potrai poi ripristinarli solo dal momento in cui sono stati riattivati.

Console

Per disabilitare immediatamente i backup automatici

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel pannello di navigazione, scegli Database, quindi scegli l'istanza database o il cluster di database multi-AZ che vuoi modificare.
3. Scegli Modifica.
4. Per Periodo di retention dei backup, seleziona 0 giorni.
5. Scegli Continue (Continua).
6. Scegliere Apply immediately (Applica immediatamente).
7. Scegli Modifica istanza database o Modifica cluster per salvare le modifiche e disabilitare i backup automatici.

AWS CLI

Per disabilitare immediatamente i backup automatici, usa il modify-db-clustercomando modify-db-instanceor e imposta il periodo di conservazione dei backup su 0 con. `--apply-immediately`

Example

L'esempio seguente disabilita immediatamente i backup automatici su un cluster di database multi-AZ.

Per LinuxmacOS, oUnix:

```
aws rds modify-db-cluster \
    --db-cluster-identifier mydbcluster \
    --backup-retention-period 0 \
    --apply-immediately
```

Per Windows:

```
aws rds modify-db-cluster ^
    --db-cluster-identifier mydbcluster ^
    --backup-retention-period 0 ^
    --apply-immediately
```

Per sapere quando viene applicata la modifica, chiama `describe-db-instances` per l'istanza database o `describe-db-clusters` per un cluster di database multi-AZ finché il valore del periodo di conservazione dei backup è 0 e lo stato di `mydbcluster` è disponibile.

```
aws rds describe-db-clusters --db-cluster-identifier mydcluster
```

API RDS

Per disabilitare immediatamente i backup automatici, chiama l'operazione ModifyDBInstance o ModifyDBCluster con i seguenti parametri:

- `DBInstanceIdentifier = mydbinstance` (o `DBClusterIdentifier = mydbcluster`)
- `BackupRetentionPeriod = 0`

Example

```
https://rds.amazonaws.com/
```

```
    ?Action=ModifyDBInstance
    &DBInstanceIdentifier=mydbinstance
    &BackupRetentionPeriod=0
    &SignatureVersion=2
    &SignatureMethod=HmacSHA256
    &Timestamp=2009-10-14T17%3A48%3A21.746Z
    &AWSAccessKeyId=<&AWS; Access Key ID>
    &Signature=<Signature>
```

Backup automatici con motori di storage MySQL non supportati

Per il motore di database MySQL, i backup automatici sono supportati solo per il motore di storage InnoDB. L'utilizzo di queste caratteristiche con altri motori di archiviazione MySQL, incluso MyISAM, può causare un comportamento inaffidabile durante il ripristino dai backup. Nello specifico, poiché i motori di storage come MyISAM non supportano il ripristino da arresto anomalo affidabile, è possibile che le tabelle vengano danneggiate in caso di arresto anomalo. Per questo, ti consigliamo di utilizzare il motore di storage InnoDB.

- Per convertire le tabelle MyISAM esistenti in tabelle InnoDB, è possibile utilizzare il comando `ALTER TABLE`, ad esempio: `ALTER TABLE table_name ENGINE=innodb, ALGORITHM=COPY;`
- Se si sceglie di utilizzare MyISAM, è possibile tentare di eseguire manualmente il ripristino delle tabelle danneggiate dopo un arresto anomalo utilizzando il comando `REPAIR`. Per ulteriori informazioni, consulta REPAIR TABLE Statement nella documentazione MySQL. Come specificato nella documentazione MySQL, è tuttavia molto probabile che non sia possibile recuperare tutti i dati.
- Per acquisire uno snapshot delle tabelle MyISAM prima del ripristino, procedere nel seguente modo:

  1. Arrestare ogni attività sulle tabelle MyISAM (ovvero, chiudere tutte le sessioni).

     È possibile chiudere tutte le sessioni chiamando il comando mysql.rds_kill per ogni processo restituito dal comando `SHOW FULL PROCESSLIST`.

  2. Bloccare e svuotare ciascuna tabella MyISAM. Ad esempio, i seguenti comandi bloccano e svuotano le due tabelle denominate `myisam_table1` e `myisam_table2`:

     ```
     mysql> FLUSH TABLES myisam_table, myisam_table2 WITH READ LOCK;
     ```

3. Crea uno snapshot dell'istanza database o del cluster di database multi-AZ. Quando la snapshot è completata, rilasciare i blocchi e riprendere l'attività sulle tabelle MyISAM. È possibile utilizzare il comando seguente per rilasciare i blocchi sulle tabelle:

```
mysql> UNLOCK TABLES;
```

Queste fasi forzano MyISAM a svuotare i dati archiviati in memoria sul disco, garantendo un avvio pulito quando si esegue il ripristino da uno snapshot DB. Per ulteriori informazioni sulla creazione di una snapshot DB, consulta Creazione di uno snapshot DB per un'istanza DB Single-AZ.

Backup automatici con motori di storage MariaDB non supportati

Per il motore di database MariaDB, i backup automatici sono supportati solo per il motore di archiviazione InnoDB. L'utilizzo di queste caratteristiche con altri motori di archiviazione MariaDB, incluso Aria, può causare un comportamento inaffidabile durante il ripristino dai backup. Sebbene Aria sia un'alternativa resistente agli arresti anomali a MyISAM, è possibile che le tabelle vengano comunque danneggiate in caso di arresto anomalo. Per questo, ti consigliamo di utilizzare il motore di storage InnoDB.

- Per convertire le tabelle Aria esistenti in tabelle InnoDB, è possibile utilizzare il comando `ALTER TABLE`. Ad esempio: `ALTER TABLE table_name ENGINE=innodb, ALGORITHM=COPY;`
- Se si sceglie di utilizzare Aria, è possibile tentare di eseguire manualmente il ripristino delle tabelle danneggiate dopo un arresto anomalo utilizzando il comando `REPAIR TABLE`. Per ulteriori informazioni sugli spazi, consulta http://mariadb.com/kb/en/mariadb/repair-table/.
- Per acquisire uno snapshot delle tabelle Aria prima del ripristino, procedere nel seguente modo:
  1. Arrestare ogni attività sulle tabelle Aria (ovvero, chiudere tutte le sessioni).
  2. Bloccare e svuotare ciascuna tabella Aria.
  3. Crea uno snapshot dell'istanza database o del cluster di database multi-AZ. Quando la snapshot è completata, rilasciare i blocchi e riprendere l'attività sulle tabelle Aria. Queste fasi forzano Aria a svuotare i dati archiviati in memoria sul disco, garantendo un avvio pulito quando si esegue il ripristino da una snapshot DB.

Replica dei backup automatici su un altro Regione AWS

Per una maggiore capacità di disaster recovery, puoi configurare l'istanza del database Amazon RDS per replicare istantanee e log delle transazioni verso una destinazione Regione AWS a tua scelta. Se per un'istanza database è configurata la replica di backup, RDS avvia una copia tra regioni di tutte le snapshot e i log delle transazioni non appena questi sono pronti nell'istanza database.

Al trasferimento dei dati vengono applicati addebiti per la copia della snapshot DB. Dopo aver copiato la snapshot DB, vengono applicati addebiti standard allo storage nella regione di destinazione. Per maggiori dettagli, consulta Prezzi di RDS.

Per un esempio di utilizzo della replica di backup, consulta il talk tecnico AWS online Managed Disaster Recovery with Amazon RDS for Oracle Cross-Region Automated Backups.

> **Note**
>
> La replica di backup automatizzata non è supportata per i cluster DB Multi-AZ.

Argomenti

- Disponibilità di regioni e versioni
- Supporto di origine e destinazione Regione AWS
- Abilitazione dei backup automatici tra regioni
- Ricerca di informazioni sui backup replicati
- Ripristino a un'ora specificata da un backup replicato
- Arresto della replica di backup automatici
- Eliminazione dei backup replicati

Disponibilità di regioni e versioni

Il supporto varia a seconda delle versioni specifiche di ciascun motore di database e a seconda delle Regioni AWS. Per ulteriori informazioni sulla disponibilità di versioni e regioni con backup automatici tra regioni, consulta Regioni e motori DB supportati per backup automatici tra regioni in Amazon RDS.

Supporto di origine e destinazione Regione AWS

La replica di Backup è supportata tra i seguenti Regioni AWS.

Regione di origine	Regioni di destinazione disponibili
Asia Pacific (Mumbai)	Asia Pacific (Singapore) Stati Uniti orientali (Virginia settentrionale), Stati Uniti orientali (Ohio), Stati Uniti occidentali (Oregon)
Asia Pacific (Osaka)	Asia Pacific (Tokyo)
Asia Pacific (Seoul)	Asia Pacifico (Singapore), Asia Pacifico (Tokyo) Stati Uniti orientali (Virginia settentrionale), Stati Uniti orientali (Ohio), Stati Uniti occidentali (Oregon)
Asia Pacific (Singapore)	Asia Pacifico (Mumbai), Asia Pacifico (Seoul), Asia Pacifico (Sydney), Asia Pacifico (Tokyo) Stati Uniti orientali (Virginia settentrionale), Stati Uniti orientali (Ohio), Stati Uniti occidentali (Oregon)
Asia Pacific (Sydney)	Asia Pacific (Singapore) Stati Uniti orientali (Virginia settentrionale), Stati Uniti occidentali (California settentrionale), Stati Uniti occidentali (Oregon)
Asia Pacific (Tokyo)	Asia Pacifico (Osaka), Asia Pacifico (Seoul), Asia Pacifico (Singapore) Stati Uniti orientali (Virginia settentrionale), Stati Uniti orientali (Ohio), Stati Uniti occidentali (Oregon)
Canada (Central)	Europe (Ireland) Stati Uniti orientali (Virginia settentrionale), Stati Uniti occidentali (California settentrionale), Stati Uniti occidentali (Oregon)
Cina (Pechino)	Cina (Ningxia)

Regione di origine	Regioni di destinazione disponibili
Cina (Ningxia)	Cina (Pechino)
Europa (Francoforte)	UE (Irlanda), UE (Londra), UE (Parigi), UE (Stoccolma) Stati Uniti orientali (Virginia settentrionale), Stati Uniti orientali (Ohio), Stati Uniti occidentali (Oregon)
Europe (Ireland)	Canada (Central) UE (Francoforte), UE (Londra), UE (Parigi), UE (Stoccolma) Stati Uniti orientali (Virginia settentrionale), Stati Uniti occidentali (California settentrionale), Stati Uniti occidentali (Oregon)
Europe (London)	UE (Francoforte), UE (Irlanda), UE (Parigi), UE (Stoccolma) US East (N. Virginia)
Europe (Paris)	UE (Francoforte), UE (Irlanda), UE (Londra), UE (Stoccolma) US East (N. Virginia)
Europe (Stockholm)	UE (Francoforte), UE (Irlanda), UE (Londra), UE (Parigi) US East (N. Virginia)
South America (São Paulo)	Stati Uniti orientali (Virginia settentrionale), Stati Uniti orientali (Ohio)
AWS GovCloud (Stati Uniti orientali)	AWS GovCloud (Stati Uniti occidentali)
AWS GovCloud (Stati Uniti occidentali)	AWS GovCloud (Stati Uniti orientali)

Regione di origine	Regioni di destinazione disponibili
Stati Uniti orientali (Virginia settentrionale)	Asia Pacifico (Mumbai), Asia Pacifico (Seoul), Asia Pacifico (Singapore), Asia Pacifico (Sydney), Asia Pacifico (Tokyo) Canada (Central) UE (Francoforte), UE (Irlanda), UE (Londra), UE (Parigi), UE (Stoccolma) South America (São Paulo) Stati Uniti orientali (Ohio), Stati Uniti occidentali (California settentrionale), Stati Uniti occidentali (Oregon)
US East (Ohio)	Asia Pacifico (Mumbai), Asia Pacifico (Seoul), Asia Pacifico (Singapore), Asia Pacifico (Tokyo) Canada (Central) UE (Francoforte), UE (Irlanda) South America (São Paulo) Stati Uniti orientali (Virginia settentrionale), Stati Uniti occidentali (California settentrionale), Stati Uniti occidentali (Oregon)
US West (N. California)	Asia Pacific (Sydney) Canada (Central) Europe (Ireland) Stati Uniti orientali (Virginia settentrionale), Stati Uniti orientali (Ohio), Stati Uniti occidentali (Oregon)

Regione di origine	Regioni di destinazione disponibili
US West (Oregon)	Asia Pacifico (Mumbai), Asia Pacifico (Seoul), Asia Pacifico (Singapore), Asia Pacifico (Sydney), Asia Pacifico (Tokyo) Canada (Central) UE (Francoforte), UE (Irlanda) Stati Uniti orientali (Virginia settentrionale), Stati Uniti orientali (Ohio), Stati Uniti occidentali (California settentrionale)

Puoi anche usare il `describe-source-regions` AWS CLI comando per scoprire quali Regioni AWS possono replicarsi tra loro. Per ulteriori informazioni, consulta Ricerca di informazioni sui backup replicati.

Abilitazione dei backup automatici tra regioni

Puoi abilitare la replica di backup su istanze database nuove o esistenti utilizzando la console Amazon RDS. È inoltre possibile utilizzare il `start-db-instance-automated-backups-replication` AWS CLI comando o l'operazione API `StartDBInstanceAutomatedBackupsReplication` RDS. È possibile replicare fino a 20 backup su ciascuna destinazione Regione AWS per ciascuna. Account AWS

> **Note**
>
> Per poter replicare i backup automatici, assicurati di attivarli. Per ulteriori informazioni, consulta Abilitazione dei backup automatici.

Console

Puoi abilitare la replica di backup per un'istanza database nuova o esistente:

- Per una nuova istanza database, abilitarla all'avvio dell'istanza. Per ulteriori informazioni, consulta Impostazioni per istanze database.
- Per un'istanza database esistente, completa la procedura descritta di seguito.

Per abilitare la replica di backup per un'istanza database esistente

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, selezionare Automated backups (Backup automatici).
3. Nella scheda Regione corrente seleziona l'istanza database per cui desideri abilitare la replica di backup.
4. Per Azioni, seleziona Gestisci replica tra regioni.
5. In Replica di backup seleziona Abilita replica in un'altra Regione AWS.
6. Seleziona la regione di destinazione.
7. Seleziona il periodo di conservazione del backup replicato.
8. Se hai abilitato la crittografia sull'istanza DB di origine, scegli l'ARN AWS KMS keyper crittografare i backup o inserisci una chiave ARN.
9. Scegliere Save (Salva).

Nella regione di origine, i backup replicati sono riportati nella scheda Regione corrente della pagina Backup automatici . Nella regione di destinazione, i backup replicati sono riportati nella scheda Backup replicati della pagina Backup automatici .

AWS CLI

Abilita la replica del backup utilizzando il comando. `start-db-instance-automated-backups-replication` AWS CLI

L'esempio seguente di CLI replica i backup automatici da un'istanza database in Stati Uniti occidentali (Oregon) a della regione Stati Uniti orientali (Virginia settentrionale),. Inoltre, crittografa i backup replicati, utilizzando una AWS KMS key nella regione di destinazione.

Per abilitare la replica di backup

- Eseguire uno dei seguenti comandi.

  PerLinux, o: macOS Unix

```
aws rds start-db-instance-automated-backups-replication \
--region us-east-1 \
--source-db-instance-arn "arn:aws:rds:us-west-2:123456789012:db:mydatabase" \
--kms-key-id "arn:aws:kms:us-east-1:123456789012:key/AKIAIOSFODNN7EXAMPLE" \
```

```
--backup-retention-period 7
```

Per Windows:

```
aws rds start-db-instance-automated-backups-replication ^
--region us-east-1 ^
--source-db-instance-arn "arn:aws:rds:us-west-2:123456789012:db:mydatabase" ^
--kms-key-id "arn:aws:kms:us-east-1:123456789012:key/AKIAIOSFODNN7EXAMPLE" ^
--backup-retention-period 7
```

L'`--source-region`opzione è necessaria quando si crittografano i backup tra le regioni AWS GovCloud (Stati Uniti orientali) e AWS GovCloud (Stati Uniti occidentali). Per `--source-region`, specifica la Regione AWS dell'istanza database di origine.

Se non si specifica `--source-region`, assicurati di specificare un valore per `--pre-signed-url`. Un URL prefirmato è un URL che contiene una richiesta firmata Signature Versione 4 per il comando `start-db-instance-automated-backups-replication` chiamato nella Regione AWS di origine. Per ulteriori informazioni sull'*`pre-signed-url`*opzione, consulta start-db-instance-automated-backups-replication nel Command Reference.AWS CLI

API RDS

Abilita la replica di backup utilizzando la funzionalità dell'API RDS `StartDBInstanceAutomatedBackupsReplication` con i seguenti parametri:

- `Region`
- `SourceDBInstanceArn`
- `BackupRetentionPeriod`
- `KmsKeyId` (facoltativo)
- `PreSignedUrl` (obbligatorio se si utilizza `KmsKeyId`)

> **Note**
>
> Se i backup vanno crittografati, è necessario includere anche un URL prefirmato. Per ulteriori informazioni sugli URL prefirmati, consulta Richieste di autenticazione: utilizzo di parametri di

query (AWS Signature Version 4) nella Documentazione di riferimento delle API di Amazon Simple Storage Service e Processo di firma Signature Version 4 in Riferimenti generali AWS .

Ricerca di informazioni sui backup replicati

Per visualizzare le informazioni sui backup replicati puoi utilizzare i seguenti comandi della CLI:

- `describe-source-regions`
- `describe-db-instances`
- `describe-db-instance-automated-backups`

L'`describe-source-regions`esempio seguente elenca la fonte Regioni AWS da cui è possibile replicare i backup automatici nella regione di destinazione degli Stati Uniti occidentali (Oregon).

Per visualizzare le informazioni sulle regioni di origine

- Eseguire il comando riportato qui di seguito.

```
aws rds describe-source-regions --region us-west-2
```

L'output mostra che i backup possono essere replicati da US East (N. Virginia), ma non da Stati Uniti orientali (Ohio) o Stati Uniti occidentali (California settentrionale) in Stati Uniti occidentali (Oregon).

```
{
    "SourceRegions": [
        ...
        {
            "RegionName": "us-east-1",
            "Endpoint": "https://rds.us-east-1.amazonaws.com",
            "Status": "available",
            "SupportsDBInstanceAutomatedBackupsReplication": true
        },
        {
            "RegionName": "us-east-2",
            "Endpoint": "https://rds.us-east-2.amazonaws.com",
            "Status": "available",
            "SupportsDBInstanceAutomatedBackupsReplication": false
        },
```

```
            "RegionName": "us-west-1",
            "Endpoint": "https://rds.us-west-1.amazonaws.com",
            "Status": "available",
            "SupportsDBInstanceAutomatedBackupsReplication": false
        }
    ]
}
```

Nell'esempio `describe-db-instances` seguente vengono illustrati i backup automatici per un'istanza database.

Per visualizzare i backup replicati per un'istanza DB

- Eseguire uno dei seguenti comandi.

  PerLinux, o: macOS Unix

  ```
  aws rds describe-db-instances \
  --db-instance-identifier mydatabase
  ```

  Per Windows:

  ```
  aws rds describe-db-instances ^
  --db-instance-identifier mydatabase
  ```

L'output include i backup replicati.

```
{
    "DBInstances": [
        {
            "StorageEncrypted": false,
            "Endpoint": {
                "HostedZoneId": "Z1PVIF0B656C1W",
                "Port": 1521,
            ...

            "BackupRetentionPeriod": 7,
            "DBInstanceAutomatedBackupsReplications":
 [{"DBInstanceAutomatedBackupsArn": "arn:aws:rds:us-east-1:123456789012:auto-backup:ab-
L2IJCEXJP7XQ7HOJ4SIEXAMPLE"}]
        }
```

```
    ]
}
```

Nell'esempio `describe-db-instance-automated-backups` seguente vengono illustrati i backup automatici per un'istanza database.

Per visualizzare i backup automatici per un'istanza database

- Eseguire uno dei seguenti comandi.

  Per LinuxmacOS, oUnix:

  ```
  aws rds describe-db-instance-automated-backups \
  --db-instance-identifier mydatabase
  ```

  Per Windows:

  ```
  aws rds describe-db-instance-automated-backups ^
  --db-instance-identifier mydatabase
  ```

L'output mostra l'istanza database di origine e i backup automatici in Stati Uniti occidentali (Oregon), con i backup replicati in US East (N. Virginia).

```
{
    "DBInstanceAutomatedBackups": [
        {
            "DBInstanceArn": "arn:aws:rds:us-west-2:868710585169:db:mydatabase",
            "DbiResourceId": "db-L2IJCEXJP7XQ7HOJ4SIEXAMPLE",
            "DBInstanceAutomatedBackupsArn": "arn:aws:rds:us-west-2:123456789012:auto-
backup:ab-L2IJCEXJP7XQ7HOJ4SIEXAMPLE",
            "BackupRetentionPeriod": 7,
            "DBInstanceAutomatedBackupsReplications":
 [{"DBInstanceAutomatedBackupsArn": "arn:aws:rds:us-east-1:123456789012:auto-backup:ab-
L2IJCEXJP7XQ7HOJ4SIEXAMPLE"}]
            "Region": "us-west-2",
            "DBInstanceIdentifier": "mydatabase",
            "RestoreWindow": {
                "EarliestTime": "2020-10-26T01:09:07Z",
                "LatestTime": "2020-10-31T19:09:53Z",
            }
            ...
```

```
        }
    ]
}
```

Nell'esempio `describe-db-instance-automated-backups` seguente viene utilizzata l'opzione `--db-instance-automated-backups-arn` per visualizzare i backup replicati nella regione di destinazione.

Per visualizzare i backup replicati

- Eseguire uno dei seguenti comandi.

  Per LinuxmacOS, oUnix:

  ```
  aws rds describe-db-instance-automated-backups \
  --db-instance-automated-backups-arn "arn:aws:rds:us-east-1:123456789012:auto-
  backup:ab-L2IJCEXJP7XQ7HOJ4SIEXAMPLE"
  ```

  Per Windows:

  ```
  aws rds describe-db-instance-automated-backups ^
  --db-instance-automated-backups-arn "arn:aws:rds:us-east-1:123456789012:auto-
  backup:ab-L2IJCEXJP7XQ7HOJ4SIEXAMPLE"
  ```

L'output mostra l'istanza database di origine in Stati Uniti occidentali (Oregon), con backup replicati in US East (N. Virginia).

```
{
    "DBInstanceAutomatedBackups": [
        {
            "DBInstanceArn": "arn:aws:rds:us-west-2:868710585169:db:mydatabase",
            "DbiResourceId": "db-L2IJCEXJP7XQ7HOJ4SIEXAMPLE",
            "DBInstanceAutomatedBackupsArn": "arn:aws:rds:us-east-1:123456789012:auto-
backup:ab-L2IJCEXJP7XQ7HOJ4SIEXAMPLE",
            "Region": "us-west-2",
            "DBInstanceIdentifier": "mydatabase",
            "RestoreWindow": {
                "EarliestTime": "2020-10-26T01:09:07Z",
                "LatestTime": "2020-10-31T19:01:23Z"
            },
```

```
            "AllocatedStorage": 50,
            "BackupRetentionPeriod": 7,
            "Status": "replicating",
            "Port": 1521,
            ...
        }
    ]
}
```

Ripristino a un'ora specificata da un backup replicato

Puoi ripristinare un'istanza database a un determinato momento temporale da un backup replicato utilizzando la console Amazon RDS . È inoltre possibile utilizzare il `restore-db-instance-to-point-in-time` AWS CLI comando o l'operazione API `RestoreDBInstanceToPointInTime` RDS.

Per informazioni generali sul point-in-time ripristino (PITR), vedere. Ripristino a un'ora specifica per un'istanza database

> **Note**
>
> In RDS per SQL Server, i gruppi di opzioni non vengono copiati Regioni AWS quando vengono replicati i backup automatici. Se è stato associato un gruppo di opzioni personalizzate all'istanza database RDS per SQL Server, è possibile ricreare tale gruppo di opzioni nella regione di destinazione. Quindi ripristina l'istanza database nella regione di destinazione e associarla al gruppo di opzioni personalizzate. Per ulteriori informazioni, consulta Uso di gruppi di opzioni.

Console

Per ripristinare un'istanza database a un'ora specificata da un backup replicato

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Sceglie la regione di destinazione (in cui vengono replicati i backup) dal selettore di regioni.
3. Nel riquadro di navigazione, selezionare Automated backups (Backup automatici).
4. Nella scheda Backup replicati scegli l'istanza database che desideri ripristinare.

5. In Actions (Operazioni), scegliere Restore to point in time (Ripristina a un punto temporale).
6. Scegliere Latest restorable time (Ultimo orario di ripristino) per eseguire il ripristino in base al momento più recente oppure scegliere Custom (Personalizzato) per scegliere una data e un'ora.

   Se si sceglie Custom (Personalizza), immettere la data e l'ora in cui si desidera ripristinare l'istanza.

   > **Note**
   >
   > Gli orari vengono visualizzati nel fuso orario locale, indicato da un offset dell'ora UTC (Coordinated Universal Time). Ad esempio, UTC-5 è l'orario standard degli Stati Uniti orientali/ora legale degli Stati Uniti orientali.

7. Per DB Instance Identifier (Identificatore istanze database), inserire il nome dell'istanza database di destinazione ripristinata.
8. (Facoltativo) Scegliere altre opzioni in base alle esigenze, ad esempio l'attivazione dell'autoscaling.
9. Scegliere Restore to point in time (Ripristina per punto nel tempo).

AWS CLI

Usa il `restore-db-instance-to-point-in-time` AWS CLI comando per creare una nuova istanza DB.

Per ripristinare un'istanza database a un'ora specificata da un backup replicato

- Eseguire uno dei seguenti comandi.

  Per LinuxmacOS, oUnix:

```
aws rds restore-db-instance-to-point-in-time \
    --source-db-instance-automated-backups-arn "arn:aws:rds:us-
east-1:123456789012:auto-backup:ab-L2IJCEXJP7XQ7HOJ4SIEXAMPLE" \
    --target-db-instance-identifier mytargetdbinstance \
    --restore-time 2020-10-14T23:45:00.000Z
```

  Per Windows:

```
aws rds restore-db-instance-to-point-in-time ^
```

```
    --source-db-instance-automated-backups-arn "arn:aws:rds:us-
east-1:123456789012:auto-backup:ab-L2IJCEXJP7XQ7HOJ4SIEXAMPLE" ^
    --target-db-instance-identifier mytargetdbinstance ^
    --restore-time 2020-10-14T23:45:00.000Z
```

API RDS

Per ripristinare un'istanza database a un momento temporale specifico, utilizza la funzionalità dell'API `RestoreDBInstanceToPointInTime` Amazon RDS con i seguenti parametri:

- `SourceDBInstanceAutomatedBackupsArn`
- `TargetDBInstanceIdentifier`
- `RestoreTime`

Arresto della replica di backup automatici

Puoi interrompere la replica di backup per le istanze database utilizzando la console Amazon RDS. È inoltre possibile utilizzare il `stop-db-instance-automated-backups-replication` AWS CLI comando o l'operazione API `StopDBInstanceAutomatedBackupsReplication` RDS.

I backup replicati vengono conservati, in base al periodo di conservazione del backup, impostato al momento della creazione.

Console

Arresta la replica di backup dalla pagina Backup automatici nella regione di origine.

Per interrompere la replica del backup su un Regione AWS

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Seleziona la regione di origine dal selettore di regioni.
3. Nel riquadro di navigazione, selezionare Automated backups (Backup automatici).
4. Nella scheda Area corrente seleziona l'istanza database per la quale desideri interrompere la replica di backup.
5. Per Azioni, seleziona Gestisci replica tra regioni.
6. In Replica di backup deseleziona la casella di controllo Abilita replica in un'altra Regione AWS.

7. Scegliere Save (Salva).

I backup replicati sono riportati nella scheda Mantenuti della pagina Backup automatici nella regione di destinazione.

AWS CLI

Interrompi la replica del backup utilizzando il `stop-db-instance-automated-backups-replication` AWS CLI comando.

Nell'esempio seguente di CLI viene interrotta la replica dei backup automatici di un'istanza database nella regione Stati Uniti occidentali (Oregon).

Per interrompere la replica di backup

- Eseguire uno dei seguenti comandi.

  Per LinuxmacOS, oUnix:

```
aws rds stop-db-instance-automated-backups-replication \
--region us-east-1 \
--source-db-instance-arn "arn:aws:rds:us-west-2:123456789012:db:mydatabase"
```

  Per Windows:

```
aws rds stop-db-instance-automated-backups-replication ^
--region us-east-1 ^
--source-db-instance-arn "arn:aws:rds:us-west-2:123456789012:db:mydatabase"
```

API RDS

Arresta la replica di backup utilizzando la funzionalità dell'API RDS `StopDBInstanceAutomatedBackupsReplication` con i seguenti parametri:

- `Region`
- `SourceDBInstanceArn`

Eliminazione dei backup replicati

Puoi eliminare i backup replicati per le istanze database utilizzando la console Amazon RDS. È inoltre possibile utilizzare il `delete-db-instance-automated-backups` AWS CLI comando o l'operazione API `DeleteDBInstanceAutomatedBackup` RDS.

Console

Elimina i backup replicati nella regione di destinazione dalla pagina Backup automatici.

Per eliminare i backup replicati

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Seleziona la regione di destinazione dal selettore di regioni.
3. Nel riquadro di navigazione, selezionare Automated backups (Backup automatici).
4. Nella scheda Backup replicati scegli l'istanza database per cui desideri eliminare i backup replicati.
5. In Actions (Azioni), selezionare Delete (Elimina).
6. Nella pagina di conferma, immetti **`delete me`** e seleziona Elimina.

AWS CLI

Elimina i backup replicati utilizzando il comando. `delete-db-instance-automated-backup` AWS CLI

Puoi utilizzare il comando della CLI `describe-db-instances` per trovare gli ARN (Amazon Resource Names) dei backup replicati. Per ulteriori informazioni, consulta Ricerca di informazioni sui backup replicati.

Per eliminare i backup replicati

- Eseguire uno dei seguenti comandi.

  PerLinux, omacOS: Unix

```
aws rds delete-db-instance-automated-backup \
--db-instance-automated-backups-arn "arn:aws:rds:us-east-1:123456789012:auto-backup:ab-L2IJCEXJP7XQ7HOJ4SIEXAMPLE"
```

Per Windows:

```
aws rds delete-db-instance-automated-backup ^
--db-instance-automated-backups-arn "arn:aws:rds:us-east-1:123456789012:auto-
backup:ab-L2IJCEXJP7XQ7HOJ4SIEXAMPLE"
```

API RDS

Elimina i backup replicati utilizzando la funzionalità dell'API `DeleteDBInstanceAutomatedBackup` RDS con il parametro `DBInstanceAutomatedBackupsArn`.

Gestione dei backup manuali

Questa sezione mostra come gestire i backup automatici per istanze DB e cluster DB.

Argomenti

Creazione di uno snapshot DB per un'istanza DB Single-AZ

Amazon RDS crea una snapshot dei volumi di storage dell'istanza database, eseguendo il backup dell'intera istanza database anziché dei singoli database. La creazione di questo snapshot DB su un'istanza database Single-AZ provoca una breve interruzione alle operazioni di I/O che può durare da pochi secondi a pochi minuti, a seconda delle dimensioni e della classe dell'istanza database. Per MariaDB, MySQL, Oracle e PostgreSQL l'attività di I/O non viene sospesa nel database principale durante il backup delle implementazioni Multi-AZ, perché il backup viene acquisito durante la fase di standby. Per SQL Server, l'attività di I/O viene sospesa brevemente durante il backup delle implementazioni Multi-AZ.

Quando crei uno snapshot DB è necessario identificare qual è l'istanza database di cui stai effettuando il backup e dare un nome allo snapshot DB in modo da poterlo usare successivamente per il ripristino. La quantità di tempo necessaria per creare uno snapshot varia a seconda della dimensione dei database. Poiché lo snapshot include l'intero volume di storage, anche la dimensione dei file, come i file temporanei, influisce sulla quantità di tempo necessaria per creare lo snapshot.

> **Note**
>
> La tua istanza database deve essere nello stato `available` per poter acquisire uno snapshot di database.
> Per le istanze di PostgreSQL DB, i dati nelle tabelle non registrate potrebbero non essere ripristinati dagli snapshot. Per ulteriori informazioni, consulta Best practice per l'utilizzo di PostgreSQL.

A differenza dei backup automatizzati, gli snapshot manuali non sono soggetti al periodo di retention dei backup. Gli snapshot non scadono.

Per i backup a lungo termine dei dati di MariaDB, MySQL e PostgreSQL, si consiglia di esportare i dati snapshot in Amazon S3. Se la versione principale del motore DB non è più supportata, non è possibile ripristinare tale versione da uno snapshot. Per ulteriori informazioni, consulta Esportazione dei dati dello snapshot DB in Simple Storage Service (Amazon S3).

È possibile creare uno snapshot DB utilizzando AWS Management Console, the o l'API AWS CLI RDS.

Console

Per creare una snapshot DB

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, selezionare Snapshots (Snapshot).

   Viene visualizzato l'elenco Snapshot manuali.
3. Seleziona Acquisisci snapshot.

   Viene visualizzata la finestra Acquisizione di snapshot DB.
4. Scegli l'istanza DB per la quale desideri scattare un'istantanea.
5. Inserisci il nome dell'istantanea.
6. Seleziona Acquisisci snapshot.

Viene visualizzato l'elenco delle istantanee manuali, con lo stato della nuova istantanea DB visualizzato come. `Creating` Dopo che lo stato è diventato `Available`, potrai vedere il tempo di creazione.

AWS CLI

Quando si crea uno snapshot DB utilizzando il AWS CLI, è necessario identificare l'istanza DB di cui eseguire il backup e quindi assegnare un nome allo snapshot DB in modo da poterlo ripristinare in un secondo momento. È possibile farlo utilizzando il AWS CLI `create-db-snapshot`comando con i seguenti parametri:

- `--db-instance-identifier`
- `--db-snapshot-identifier`

In questo esempio crei uno snapshot DB denominato *`mydbsnapshot`* per un'istanza database denominata *`mydbinstance`*.

Example

Per LinuxmacOS, oUnix:

```
aws rds create-db-snapshot \
```

```
    --db-instance-identifier mydbinstance \
    --db-snapshot-identifier mydbsnapshot
```

Per Windows:

```
aws rds create-db-snapshot ^
    --db-instance-identifier mydbinstance ^
    --db-snapshot-identifier mydbsnapshot
```

API RDS

Quando crei uno snapshot DB usando l'API Amazon RDS è necessario identificare qual è l'istanza database di cui stai effettuando il backup e dare un nome allo snapshot DB in modo da poterlo usare successivamente per il ripristino. Puoi eseguire questa operazione utilizzando il comando API Amazon RDS `CreateDBSnapshot` con i seguenti parametri:

- `DBInstanceIdentifier`
- `DBSnapshotIdentifier`

Creazione di uno snapshot di un cluster di database Multi-AZ

Quando crei uno snapshot di cluster database Multi-AZ assicurati di identificare qual è il cluster database Multi-AZ di cui stai effettuando il backup e dare un nome alla snapshot di cluster database in modo da poterlo usare successivamente per il ripristino. Puoi anche condividere uno snapshot di cluster di database multi-AZ. Per istruzioni, consulta the section called “Condivisione di uno snapshot del database”.

È possibile creare un'istantanea del cluster DB Multi-AZ utilizzando l' AWS Management ConsoleAPI AWS CLI, the o RDS.

Console

Per creare uno snapshot del cluster database

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, scegli Databases (Database).
3. Nell'elenco scegliere il cluster di database Multi-AZ per il quale si desidera fare uno snapshot.
4. Per Actions (Operazioni), selezionare Take snapshot (Acquisisci snapshot).

   Viene visualizzata la finestra Acquisizione di snapshot DB.
5. Per Nome snapshot, inserisci il nome dello snapshot.
6. Seleziona Acquisisci snapshot.

Viene visualizzata la pagina Snapshot, con lo stato del nuovo snapshot del cluster di database Multi-AZ come `Creating`. Dopo che lo stato è diventato `Available`, potrai vedere il tempo di creazione.

AWS CLI

Puoi creare uno snapshot del cluster DB Multi-AZ utilizzando il AWS CLI create-db-cluster-snapshotcomando con le seguenti opzioni:

- `--db-cluster-identifier`
- `--db-cluster-snapshot-identifier`

In questo esempio crei uno snapshot del cluster di database Multi-AZ denominato *`mymultiazdbclustersnapshot`* per un cluster di database denominato *`mymultiazdbcluster`*.

Example

PerLinux, omacOS: Unix

```
aws rds create-db-cluster-snapshot \
    --db-cluster-identifier mymultiazdbcluster \
    --db-cluster-snapshot-identifier mymultiazdbclustersnapshot
```

Per Windows:

```
aws rds create-db-cluster-snapshot ^
    --db-cluster-identifier mymultiazdbcluster ^
    --db-cluster snapshot-identifier mymultiazdbclustersnapshot
```

API RDS

Puoi creare uno snapshot del cluster DB Multi-AZ utilizzando l'ClusterSnapshotoperazione Amazon RDS API CreateDB con i seguenti parametri:

- `DBClusterIdentifier`
- `DBClusterSnapshotIdentifier`

Eliminazione di uno snapshot di cluster di database multi-AZ

Puoi eliminare gli snapshot dei cluster di database multi-AZ gestiti da Amazon RDS quando non ti servono più. Per istruzioni, consultare the section called “Eliminazione di una snapshot DB”.

Eliminazione di una snapshot DB

Puoi eliminare snapshot del cluster di database gestiti da Amazon RDS quando non ti servono più.

> **Note**
>
> Per eliminare backup gestiti da AWS Backup, utilizza la console AWS Backup. Per ulteriori informazioni su AWS Backup, consulta la Guida per sviluppatori di AWS Backup.

Eliminazione di una snapshot DB

Puoi eliminare una snapshot DB pubblica, condivisa o manuale utilizzando la AWS Management Console, AWS CLI o l'API RDS.

Per eliminare uno snapshot condiviso o pubblico, devi accedere all'account AWS proprietario dello snapshot.

Se hai snapshot DB automatizzate che desideri eliminare senza eliminare l'istanza database, modifica il periodo di retention dei backup per l'istanza database a 0. Le snapshot automatizzate vengono eliminate all'applicazione della modifica. Se non desideri aspettare fino al periodo di manutenzione successivo, puoi applicare la modifica immediatamente. Dopo aver completato la modifica, puoi riabilitare i backup automatici impostando il periodo di retention dei backup a un numero maggiore di 0. Per ulteriori informazioni sulla modifica di un'istanza di database, consulta Modifica di un'istanza database Amazon RDS.

I backup automatici conservati e le istantanee manuali comportano costi di fatturazione fino alla loro eliminazione. Per ulteriori informazioni, consulta Costi di retention.

Se hai eliminato un'istanza database, puoi eliminare le sue snapshot DB automatizzate rimuovendo i backup automatici per l'istanza database. Per informazioni sui backup automatici, consulta Introduzione ai backup.

Console

Per eliminare una snapshot DB

1. Accedi alla AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, selezionare Snapshots (Snapshot).

Viene visualizzato l'elenco Snapshot manuali.

3. Scegliere la snapshot DB da eliminare.
4. Per Actions (Operazioni), scegliere Delete Snapshot (Elimina snapshot).
5. Nella pagina di conferma, scegliere Delete (Elimina).

AWS CLI

È possibile eliminare un'istantanea del DB utilizzando il AWS CLI comando delete-db-snapshot.

Le seguenti opzioni vengono utilizzate per eliminare una snapshot DB.

- `--db-snapshot-identifier` – L'identificatore per la snapshot DB.

Example

Il seguente codice elimina la snapshot DB `mydbsnapshot`.

Per LinuxmacOS, oUnix:

```
aws rds delete-db-snapshot \
    --db-snapshot-identifier mydbsnapshot
```

Per Windows:

```
aws rds delete-db-snapshot ^
    --db-snapshot-identifier mydbsnapshot
```

API RDS

Puoi eliminare una snapshot DB usando l'operazione API di Amazon RDS copy-db-snapshot.

I seguenti parametri vengono utilizzati per eliminare una snapshot DB.

- `DBSnapshotIdentifier` – L'identificatore per la snapshot DB.

Ripristino da uno snapshot database

Questa sezione mostra come eseguire il ripristino da un'istantanea del DB.

Argomenti

Amazon RDS crea uno snapshot dei volumi di storage dell'istanza database, eseguendo il backup dell'intera istanza database anziché dei singoli database. È possibile creare una nuova istanza database eseguendo il ripristino da uno snapshot di database. Si specifica il nome dello snapshot di database da cui ripristinare, quindi si fornisce un nome per la nuova istanza database che viene creata dal ripristino. Non è possibile eseguire il ripristino da una snapshot di database su un'istanza database esistente. Quando esegui il ripristino, viene creata una nuova istanza database.

È possibile utilizzare l'istanza database ripristinata non appena lo stato diventa `available`. L'istanza del cluster database continuerà a caricare i dati in background. Questo processo è noto come caricamento lento.

Se accedi a dati che non sono ancora stati caricati, l'istanza del cluster database scarica immediatamente i dati richiesti da Amazon S3 e continua a caricare il resto dei dati in background. Per ulteriori informazioni, consulta Snapshot Amazon EBS.

Per ridurre gli effetti del caricamento lento sulle tabelle a cui è necessario accedere rapidamente, è possibile eseguire operazioni che comportano scansioni di tabelle complete, ad esempio `SELECT *`. Ciò consente ad Amazon RDS di scaricare tutti i dati della tabella di backup da S3.

Puoi effettuare il ripristino di un'istanza database e utilizzare un tipo di storage diverso dalla snapshot DB di origine. In questo caso, il processo di ripristino è più lento, a causa del lavoro aggiuntivo richiesto per migrare i dati al nuovo tipo di storage. Se effettui il ripristino su o dallo storage magnetico, il processo di migrazione è particolarmente lento. Questo perché lo storage magnetico non dispone della funzionalità IOPS dello storage Provisioned IOPS o General Purpose (SSD).

È possibile utilizzare AWS CloudFormation per ripristinare un'istanza DB da uno snapshot di un'istanza DB. Per ulteriori informazioni, consulta AWS::RDS::DBInstance nella Guida per l'utente di AWS CloudFormation .

> **Note**
>
> Non è possibile ripristinare un'istanza database da una snapshot DB condivisa e crittografata. Invece puoi copiare la snapshot DB e ripristinare l'istanza database dalla copia. Per ulteriori informazioni, consulta Copia di una snapshot DB..

Per informazioni sul ripristino di un'istanza DB con una versione RDS Extended Support, vedere. Ripristino di un'istanza DB o di un cluster DB Multi-AZ, di un cluster Amazon RDS Extended Support

Considerazioni sui gruppi di parametri

È consigliabile mantenere il gruppo parametri del database per tutti gli snapshot DB creati, in modo che sia possibile associare l'istanza database ripristinata al gruppo di parametri corretto.

Il gruppo parametri del database di default è associato all'istanza ripristinata, a meno che non se ne scelga una diversa. Nel gruppo di parametri di default non sono disponibili impostazioni di parametro personalizzate.

È possibile specificare il gruppo di parametri al momento del ripristino dell'istanza database.

Per ulteriori informazioni sui gruppi di parametri database, consulta Utilizzo di gruppi di parametri.

Considerazioni relative al gruppo di sicurezza

Quando ripristini un'istanza database, il cloud privato virtuale (VPC) di default, il gruppo di sottoreti del database e il gruppo di sicurezza VPC sono associati all'istanza ripristinata, a meno che non si scelgano altri gruppi.

- Se utilizzi la console Amazon RDS, puoi specificare un gruppo di sicurezza VPC personalizzato da associare all'istanza o creare un nuovo gruppo di sicurezza VPC.
- Se utilizzi il AWS CLI, puoi specificare un gruppo di sicurezza VPC personalizzato da associare all'istanza includendo l'`--vpc-security-group-ids`opzione nel comando. `restore-db-instance-from-db-snapshot`
- Se utilizzi l'API di Amazon RDS, puoi includere il parametro `VpcSecurityGroupIds.VpcSecurityGroupId.N` nell'operazione `RestoreDBInstanceFromDBSnapshot`.

Non appena il ripristino è completo e la nuova istanza database è disponibile, puoi anche cambiare le impostazioni del VPC modificando l'istanza database. Per ulteriori informazioni, consulta Modifica di un'istanza database Amazon RDS.

Considerazioni su gruppi di opzioni

Quando si ripristina un'istanza database, il gruppo di opzioni di database predefinito viene associato all'istanza database ripristinata nella maggior parte dei casi.

L'eccezione è quando l'istanza database di origine è associata a un gruppo di opzioni contenente un'opzione persistente o permanente. Ad esempio, se l'istanza database di origine utilizza Oracle TDE (Transparent Data Encryption), l'istanza database ripristinata deve utilizzare un gruppo di opzioni contenente l'opzione TDE.

Se ripristini un'istanza database in un VPC diverso, devi eseguire una delle seguenti operazioni per assegnare un gruppo di opzioni di database:

- Assegnare all'istanza il gruppo di opzioni di default per quel gruppo di VPC.
- Assegnare un altro gruppo di opzioni collegato a tale VPC.
- Creare un nuovo gruppo di opzioni e assegnarlo all'istanza database. Con le opzioni persistenti o permanenti, come Oracle TDE, devi creare un nuovo gruppo di opzioni che includa l'opzione persistente o permanente.

Per ulteriori informazioni sui gruppi di opzioni di database, consulta Uso di gruppi di opzioni.

Considerazioni sull’assegnazione di tag alle risorse

Quando ripristini un'istanza database da uno snapshot DB, RDS controlla se hai specificato nuovi tag. In caso affermativo, i nuovi tag vengono aggiunti all'istanza database ripristinata. Se non ci sono nuovi tag, RDS aggiunge i tag dall'istanza database di origine al momento della creazione dello snapshot nell'istanza database ripristinata.

Per ulteriori informazioni, consulta Copia di tag in snapshot di istanze database.

Considerazioni su Db2

Con il modello BYOL, le tue istanze DB RDS per Db2 devono essere associate a un gruppo di parametri personalizzato che contenga le tue istanze database e le tue. IBM Site ID IBM Customer ID In caso contrario, i tentativi di ripristinare un'istanza DB da un'istantanea falliranno. Per ulteriori informazioni, consulta Porta la tua licenza per Db2 e rdsadmin.restore_database.

Con il Marketplace AWS modello di licenza Db2, è necessario un Marketplace AWS abbonamento attivo per la particolare IBM Db2 edizione che si desidera utilizzare. Se non ne hai già uno, iscriviti a Db2 Marketplace AWS per quell'IBM Db2edizione. Per ulteriori informazioni, consulta Licenza Db2 tramite Marketplace AWS.

Considerazioni su Microsoft SQL Server

Quando ripristini uno snapshot database di RDS per Microsoft SQL Server su una nuova istanza, puoi sempre ripristinare la stessa edizione dello snapshot. In alcuni casi puoi anche cambiare l'edizione dell'istanza database. Le limitazioni di seguito sono riportate sono applicabili quando cambi le edizioni:

- Alla snapshot DB deve essere assegnato uno storage sufficiente per la nuova edizione.
- Sono supportate solo le seguenti modifiche per l'edizione:
  - Da Standard Edition a Enterprise Edition
  - Da Web Edition a Standard Edition o Enterprise Edition
  - Da Express Edition a Web Edition, Standard Edition o Enterprise Edition

Se desideri passare da un'edizione a una nuova edizione non supportata ripristinando una snapshot, puoi tentare di utilizzare la funzione di backup e ripristino nativi. SQL Server verifica la compatibilità

del database con la nuova edizione sulla base delle funzionalità SQL Server abilitate nel database. Per ulteriori informazioni, consulta Importazione ed esportazione di database SQL Server mediante backup e ripristino nativi.

Considerazioni su Oracle Database

Quando ripristini un database Oracle da uno snapshot di database, considera quanto segue:

- Prima di ripristinare uno snapshot di database, è possibile aggiornarlo a una versione successiva. Per ulteriori informazioni, consulta Aggiornamento di uno shapshot DB Oracle.
- Se ripristini uno snapshot di un'istanza CDB che utilizza la configurazione a tenant singolo, è possibile modificare il nome PDB. Non è possibile modificare i nomi PDB quando l'istanza CDB utilizza la configurazione multi-tenant. Per ulteriori informazioni, consulta Backup e ripristino di un CDB.
- Non è possibile modificare il nome CDB, che è sempre `RDSCDB`. Questo nome CDB è lo stesso per tutte le istanze CDB.
- Non è possibile interagire direttamente con i database del tenant in uno snapshot di database. Se ripristini uno snapshot di un'istanza CDB che utilizza la configurazione multi-tenant, ripristini tutti i relativi database del tenant. È possibile utilizzare describe-db-snapshot-tenant-databases per ispezionare i database tenant all'interno di uno snapshot DB prima di ripristinarlo.
- Se usi Oracle GoldenGate, mantieni sempre il gruppo di parametri con il parametro. `compatible` Quando ripristini un'istanza database da una snapshot DB, specifica un gruppo di parametri con un valore `compatible` uguale o superiore.
- È possibile scegliere di rinominare il database quando si ripristina uno snapshot del DB. Se la dimensione totale del redo log online è superiore a 20 GB, RDS potrebbe ripristinare le dimensioni dei redo log online alle impostazioni predefinite di 512 MB (4 x 128 MB). Le dimensioni ridotte consentono di completare l'operazione di ripristino in un tempo ragionevole. È possibile ricreare i redo log online in un secondo momento e modificarne le dimensioni.

Ripristino da uno snapshot

È possibile ripristinare un'istanza DB da un'istantanea del database utilizzando l'API AWS Management Console AWS CLI, the o RDS.

> **Note**
>
> Non puoi ridurre lo spazio di archiviazione quando ripristini un'istanza database. Quando si aumenta lo storage allocato, questo valore deve essere almeno del 10%. Se si prova ad aumentarlo di un valore inferiore al 10%, verrà visualizzato un errore. Non puoi aumentare lo spazio di archiviazione allocato quando ripristini le istanze database RDS per SQL Server.

Console

Per ripristinare un'istanza database da uno snapshot DB

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, selezionare Snapshots (Snapshot).
3. Scegliere la snapshot DB dalla quale effettuare il ripristino.
4. Per Actions (Operazioni), selezionare Restore Snapshot (Ripristina snapshot).
5. Nella pagina Ripristina snapshot, in Identificatore istanza database, immettere il nome dell'istanza database ripristinata.
6. Specifica altre impostazioni, ad esempio la dimensione dello spazio di archiviazione allocato.

   Per informazioni su ciascuna impostazione, consulta Impostazioni per istanze database.
7. Selezionare Ripristina istanza database.

AWS CLI

Per ripristinare un'istanza DB da uno snapshot DB, usa il AWS CLI comando restore-db-instance-from -db-snapshot.

In questo esempio il ripristino avviene da uno snapshot DB creato precedentemente e denominato `mydbsnapshot`. Viene ripristinata una nuova istanza database denominata `mynewdbinstance`. Questo esempio imposta anche la dimensione dello spazio di archiviazione allocato.

È possibile specificare altre impostazioni. Per informazioni su ciascuna impostazione, consulta Impostazioni per istanze database.

Example

PerLinux, o: macOS Unix

```
aws rds restore-db-instance-from-db-snapshot \
    --db-instance-identifier mynewdbinstance \
    --db-snapshot-identifier mydbsnapshot \
    --allocated-storage 100
```

Per Windows:

```
aws rds restore-db-instance-from-db-snapshot ^
    --db-instance-identifier mynewdbinstance ^
    --db-snapshot-identifier mydbsnapshot ^
    --allocated-storage 100
```

Questo comando restituisce un output simile al seguente:

```
DBINSTANCE  mynewdbinstance  db.t3.small  MySQL     50       sa              creating
 3  n  8.0.28  general-public-license
```

API RDS

Per ripristinare un'istanza DB da uno snapshot DB, chiama la funzione API di Amazon RDS RestoreDB InstanceFrom dbSnapshot con i seguenti parametri:

- `DBInstanceIdentifier`
- `DBSnapshotIdentifier`

Ripristino a un'ora specifica per un'istanza database

È possibile ripristinare un'istanza DB in un momento specifico, creando una nuova istanza DB senza modificare l'istanza DB di origine.

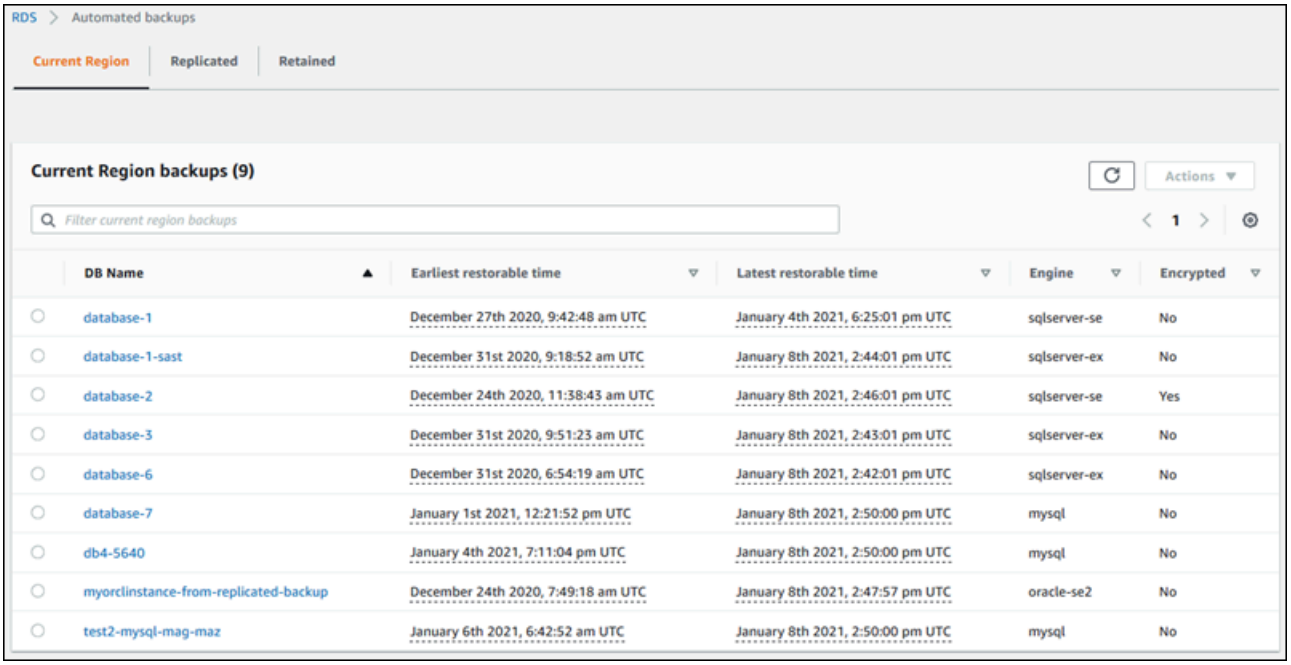
Quando ripristini un'istanza database a un determinato momento, puoi scegliere il gruppo di sicurezza VPC (Virtual Private Cloud) predefinito. In alternativa, puoi applicare un gruppo di sicurezza VPC personalizzato alla tua istanza database.

Le istanze database ripristinate vengono associate automaticamente ai gruppi di parametri e opzioni predefiniti del database. Tuttavia, puoi applicare un gruppo di parametri e un gruppo di opzioni personalizzati specificandoli durante un ripristino.

Se l'istanza database di origine ha tag di risorsa, RDS aggiunge i tag più recenti all'istanza database ripristinata.

RDS carica i log delle transazioni per le istanze database in Amazon S3 ogni cinque minuti. Per visualizzare l'ora di ripristino più recente per un'istanza DB, usa il AWS CLI [describe-db-instances](#)comando e guarda il valore restituito nel `LatestRestorableTime` campo per l'istanza DB. Per visualizzare l'ora di ripristino più recente per ogni istanza del DB nella console Amazon RDS, scegliere Backup automatici.

Puoi eseguire il ripristino point-in-time durante il periodo di retention dei backup. Per visualizzare il tempo di ripristino più breve per ogni istanza del DB, scegliere Backup automatici nella console Amazon RDS.



> **Note**
>
> Si consiglia di ripristinare una dimensione identica o simile dell'istanza database — e IOPS se si utilizza lo storage IOPS con provisioning — come istanza database di origine. È possibile che venga visualizzato un errore se, ad esempio, si sceglie una dimensione di istanza DB con un valore IOPS incompatibile.

Per informazioni sul ripristino di un'istanza DB con una versione RDS Extended Support, vedere. Ripristino di un'istanza DB o di un cluster DB Multi-AZ, di un cluster Amazon RDS Extended Support

Per alcuni motori di database utilizzati da Amazon RDS si applicano considerazioni speciali ai fini del ripristino point-in-time.

- Se utilizzi l'autenticazione tramite password con un'istanza DB RDS for Db2, le azioni di gestione degli utenti, inclu`erdsadmin.add_user`, non verranno registrate nei log. Queste azioni richiedono un backup completo dell'istantanea.

  Con il modello BYOL, le tue istanze DB RDS per Db2 devono essere associate a un gruppo di parametri personalizzato che contenga le tue e le tue. IBM Site ID IBM Customer ID In caso contrario, i tentativi di ripristinare un'istanza DB in un momento specifico falliranno. Per ulteriori informazioni, consulta Porta la tua licenza per Db2 e rdsadmin.restore_database.

  Con il Marketplace AWS modello di licenza Db2, è necessario un Marketplace AWS abbonamento attivo per la particolare IBM Db2 edizione che si desidera utilizzare. Se non ne hai già uno, iscriviti a Db2 Marketplace AWS per quell'IBM Db2edizione. Per ulteriori informazioni, consulta Licenza Db2 tramite Marketplace AWS.
- Durante il ripristino point-in-time per un'istanza database Oracle, puoi specificare un motore di database Oracle, un modello di licenza e un DBName (SID) differenti da utilizzare per la nuova istanza database.
- Quando si esegue il ripristino point-in-time di un'istanza database Microsoft SQL Server, ogni database nell'istanza viene ripristinato a un point-in-time entro 1 secondo da ciascuno degli altri database nell'istanza. Le transazioni che si estendono su più database nell'istanza potrebbero essere ripristinate in modo incoerente.
- Per un'istanza database di SQL Server, le modalità `OFFLINE`, `EMERGENCY` e `SINGLE_USER` non sono supportate. Impostando una di queste modalità per un database, l'ora di ripristino più recente non si sposterà più in avanti per l'intera istanza.

- Alcune azioni, come la modifica del modello di ripristino di un database SQL Server, possono interrompere la sequenza di log utilizzati per point-in-time il ripristino. In alcuni casi, Amazon RDS può rilevare questo problema e all'ultima ora di ripristino sarà impedito di andare avanti. In altri casi, ad esempio quando un database SQL Server utilizza il modello di ripristino `BULK_LOGGED`, l'interruzione nella sequenza di log non viene rilevata. Potrebbe non essere possibile eseguire il ripristino point-in-time di un'istanza database SQL Server in caso di interruzione della sequenza dei log. Per questi motivi, Amazon RDS non supporta la modifica del modello di ripristino dei database SQL Server.

Puoi anche utilizzarlo AWS Backup per gestire i backup delle istanze database di Amazon RDS. Se l'istanza DB è associata a un piano di backup in AWS Backup, tale piano di backup viene utilizzato per il ripristino. point-in-time I backup creati con AWS Backup hanno nomi che terminano con. `awsbackup:`*`AWS-Backup-job-number`* Per informazioni in merito AWS Backup, consulta la Guida per gli AWS Backup sviluppatori.

> **Note**
>
> Le informazioni contenute in questo argomento si applicano ad Amazon RDS. Per informazioni sul ripristino del cluster database di Amazon Aurora, consulta Ripristino di un cluster di database a un determinato momento.

È possibile ripristinare un'istanza DB in un determinato momento utilizzando l' AWS Management Console API RDS o l'API RDS. AWS CLI

> **Note**
>
> Non puoi ridurre lo spazio di archiviazione quando ripristini un'istanza database. Quando si aumenta lo storage allocato, questo valore deve essere almeno del 10%. Se si prova ad aumentarlo di un valore inferiore al 10%, verrà visualizzato un errore. Non puoi aumentare lo spazio di archiviazione allocato quando ripristini le istanze database RDS per SQL Server.

Console

Per ripristinare un'istanza database a un punto temporale specifico

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, selezionare Automated backups (Backup automatici).

   I backup automatici vengono visualizzati nella scheda Current Region (Regione corrente).
3. Scegli l'istanza database da ripristinare.
4. In Actions (Operazioni), scegli Restore to point in time (Ripristina a un istante temporale).

   Viene visualizzata la finestra Restore to point in time (Ripristina a un istante temporale).
5. Scegliere Latest restorable time (Ultimo orario di ripristino) per eseguire il ripristino in base al momento più recente oppure scegliere Custom (Personalizzato) per scegliere una data e un'ora.

   Se scegli Personalizzato, specifica la data e l'ora in cui desideri ripristinare l'istanza.

   > **Note**
   >
   > Gli orari vengono visualizzati nel fuso orario locale, indicato come un offset dell'ora UTC (Coordinated Universal Time). Ad esempio, UTC-5 è l'orario standard degli Stati Uniti orientali/ora legale degli Stati Uniti orientali.
6. Per DB Instance Identifier (Identificatore istanze database), inserire il nome dell'istanza database di destinazione ripristinata. Il nome deve essere univoco.
7. Scegli altre opzioni in base alle esigenze, ad esempio la classe di istanza database, l'archiviazione e se desideri utilizzare la funzione di scalabilità automatica dell'archiviazione.

   Per informazioni su ciascuna impostazione, consulta Impostazioni per istanze database.
8. Scegli Restore to point in time (Ripristina per punto nel tempo).

AWS CLI

Per ripristinare un'istanza DB a un'ora specificata, usa il AWS CLI comando restore-db-instance-to-point-in-time to creare una nuova istanza DB. Questo esempio inoltre imposta la dimensione dello spazio di archiviazione allocato e abilita la scalabilità automatica dell'archiviazione.

Il tagging di risorse è supportato per questa operazione. Quando usi l'opzione `--tags`, i tag dell'istanza database di origine vengono ignorati e vengono utilizzati quelli forniti. In caso contrario, vengono utilizzati i tag più recenti dell'istanza di origine.

È possibile specificare altre impostazioni. Per informazioni su ciascuna impostazione, consulta Impostazioni per istanze database.

Example

Per LinuxmacOS, oUnix:

```
aws rds restore-db-instance-to-point-in-time \
    --source-db-instance-identifier mysourcedbinstance \
    --target-db-instance-identifier mytargetdbinstance \
    --restore-time 2017-10-14T23:45:00.000Z \
    --allocated-storage 100 \
    --max-allocated-storage 1000
```

Per Windows:

```
aws rds restore-db-instance-to-point-in-time ^
    --source-db-instance-identifier mysourcedbinstance ^
    --target-db-instance-identifier mytargetdbinstance ^
    --restore-time 2017-10-14T23:45:00.000Z ^
    --allocated-storage 100 ^
    --max-allocated-storage 1000
```

API RDS

Per ripristinare un'istanza database a un punto temporale specifico, utilizzare l'operazione API Amazon RDS `RestoreDBInstanceToPointInTime` con i parametri seguenti:

- `SourceDBInstanceIdentifier`
- `TargetDBInstanceIdentifier`
- `RestoreTime`

Ripristino di un cluster di database Multi-AZ a un determinato momento

È possibile ripristinare un cluster di database Multi-AZ a un determinato momento, creando un nuovo cluster di database Multi-AZ.

RDS carica i log delle transazioni per i cluster database Multi-AZ in Amazon S3 continuamente. Puoi eseguire il ripristino point-in-time durante il tempo di conservazione del backup. Per visualizzare il primo orario di ripristino per un cluster DB Multi-AZ, utilizzare il AWS CLI describe-db-clusterscomando. Guarda il valore restituito nel campo `EarliestRestorableTime` per il cluster di database. Per visualizzare l'ultima ora di ripristino per un cluster di database Multi-AZ, guarda il valore restituito nel campo `LatestRestorableTime` per il cluster di database.

Quando ripristini un cluster DB Multi-AZ in un determinato momento, puoi scegliere il gruppo di sicurezza VPC predefinito per il tuo cluster DB Multi-AZ oppure puoi applicare un gruppo di sicurezza VPC personalizzato al tuo cluster DB Multi-AZ.

I cluster database Multi-AZ ripristinati vengono associati automaticamente al gruppo di parametri del cluster di datababse predefinito. Tuttavia, è possibile applicare un gruppo di parametri del cluster DB personalizzato specificandolo durante un ripristino.

Se il cluster DB di origine dispone di tag di risorsa, RDS aggiunge i tag più recenti al cluster DB ripristinato.

> **Note**
>
> Si consiglia di ripristinare una dimensione identica o simile del cluster di database Multi-AZ come cluster di database di origine. Si consiglia inoltre di eseguire il ripristino con un valore IOPS uguale o simile se si utilizza l'archiviazione IOPS con provisioning. È possibile che venga visualizzato un errore se, ad esempio, si sceglie una dimensione di cluster database con un valore IOPS incompatibile.
> Se il cluster DB Multi-AZ di origine utilizza lo storage SSD General Purpose (gp3) e dispone di meno di 400 GiB di storage allocato, non è possibile modificare gli IOPS assegnati per il cluster DB ripristinato.

Per informazioni sul ripristino di un cluster DB Multi-AZ con una versione RDS Extended Support, vedere. Ripristino di un'istanza DB o di un cluster DB Multi-AZ, di un cluster Amazon RDS Extended Support

È possibile ripristinare un cluster DB Multi-AZ in un punto temporale utilizzando AWS Management Console, the o l'API AWS CLI RDS.

Console

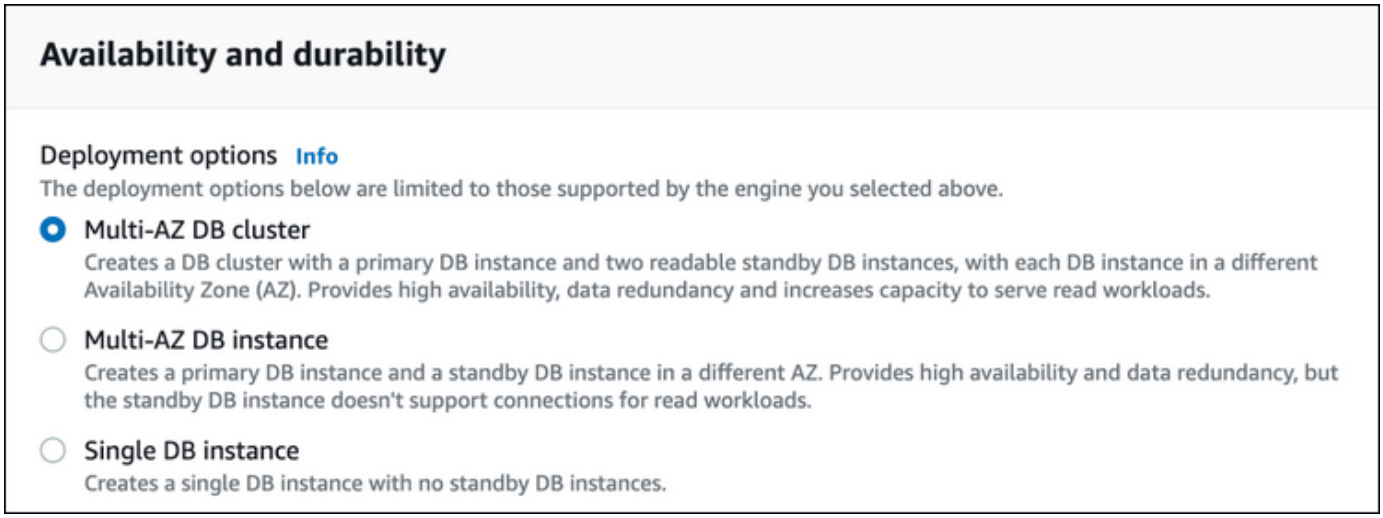
Per ripristinare un cluster di database Multi-AZ a un determinato momento

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, scegli Databases (Database).
3. Seleziona il cluster di database Multi-AZ che desideri ripristinare.
4. In Actions (Operazioni), scegli Restore to point in time (Ripristina a un istante temporale).

   Viene visualizzata la finestra Restore to point in time (Ripristina a un istante temporale).
5. Scegliere Latest restorable time (Ultimo orario di ripristino) per eseguire il ripristino in base al momento più recente oppure scegliere Custom (Personalizzato) per scegliere una data e un'ora.

   Se scegli Custom (Personalizzato), specifica la data e l'ora in cui desideri ripristinare il cluster di database Multi-AZ.

   > **Note**
   >
   > Gli orari vengono visualizzati nel fuso orario locale, indicato come un offset dell'ora UTC (Coordinated Universal Time). Ad esempio, UTC-5 è l'orario standard degli Stati Uniti orientali/ora legale degli Stati Uniti centrali.
6. Per Identificativo cluster di database, specificare il nome del cluster di database Multi-AZ ripristinato.
7. In Availability and durability (Disponibilità e durabilità), scegliere Multi-AZ DB cluster (Cluster di database Multi-AZ).



8. In DB instance class (classe dell'istanza del database), selezionare una classe dell'istanza database.

   Attualmente, i cluster di database Multi-AZ supportano solo le classi di istanza database db.m6gd e db.r6gd. Per altre informazioni sulle classi di istanza database, consulta Classi di istanze database.
9. Per le restanti sezioni, specifica le impostazioni del cluster di database. Per informazioni su ciascuna impostazione, consulta Impostazioni per la creazione di cluster di database Multi-AZ.
10. Scegli Restore to point in time (Ripristina per punto nel tempo).

AWS CLI

Per ripristinare un cluster DB Multi-AZ a un orario specificato, usa il AWS CLI comando restore-db-cluster-to- point-in-time to creare un nuovo cluster DB Multi-AZ.

Attualmente, i cluster di database Multi-AZ supportano solo le classi di istanza database db.m6gd e db.r6gd. Per altre informazioni sulle classi di istanza database, consulta Classi di istanze database.

Example

PerLinux, omacOS: Unix

```
aws rds restore-db-cluster-to-point-in-time \
    --source-db-cluster-identifier mysourcemultiazdbcluster \
    --db-cluster-identifier mytargetmultiazdbcluster \
    --restore-to-time 2021-08-14T23:45:00.000Z \
    --db-cluster-instance-class db.r6gd.xlarge
```

Per Windows:

```
aws rds restore-db-cluster-to-point-in-time ^
    --source-db-cluster-identifier mysourcemultiazdbcluster ^
    --db-cluster-identifier mytargetmultiazdbcluster ^
    --restore-to-time 2021-08-14T23:45:00.000Z ^
    --db-cluster-instance-class db.r6gd.xlarge
```

API RDS

Per ripristinare un cluster DB a un'ora specificata, chiama l'ClusterToPointInTimeoperazione Amazon RDS API RestoreDB con i seguenti parametri:

- `SourceDBClusterIdentifier`
- `DBClusterIdentifier`
- `RestoreToTime`

Ripristino da uno snapshot a un cluster di database Multi-AZ

È possibile ripristinare un'istantanea in un cluster DB Multi-AZ utilizzando l'API AWS Management Console AWS CLI, the o RDS. È possibile ripristinare ciascuno di questi tipi di snapshot in un cluster di database Multi-AZ:

- Uno snapshot di implementazione single-AZ
- Un'istantanea di una distribuzione di cluster DB Multi-AZ con una singola istanza DB
- Uno snapshot di un cluster di database Multi-AZ

Per informazioni sulle implementazioni multi-AZ, consulta Configurazione e gestione di un'implementazione multi-AZ.

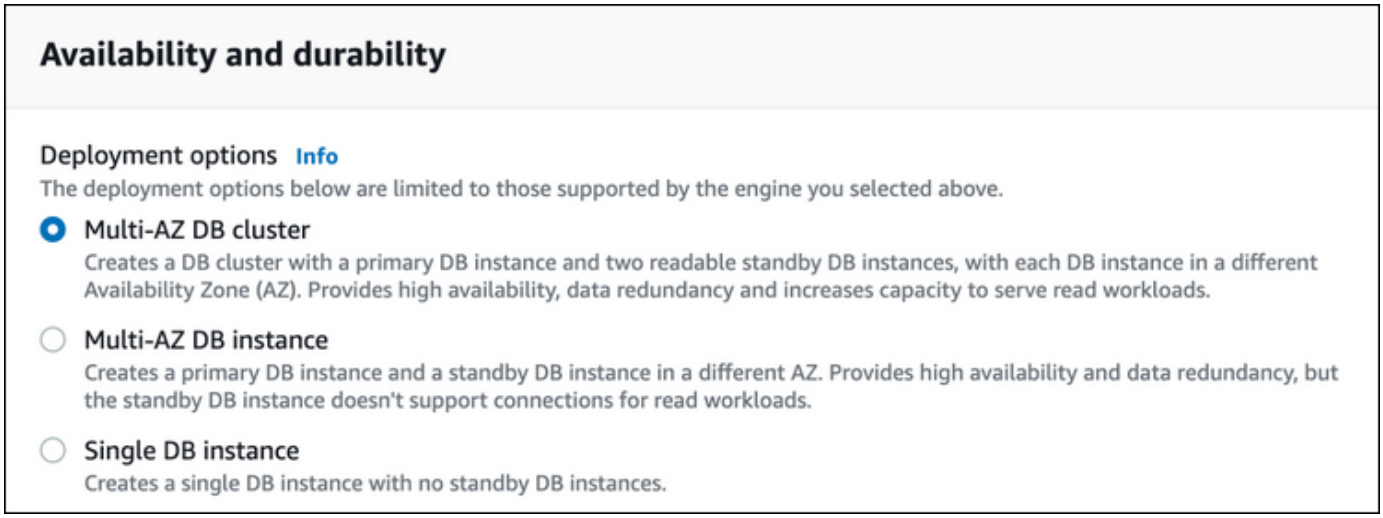
> **Tip**
>
> È possibile migrare una distribuzione Single-AZ o una distribuzione di cluster DB Multi-AZ a una distribuzione di cluster DB Multi-AZ ripristinando un'istantanea.

Per informazioni sul ripristino del cluster DB Multi-AZ con una versione RDS Extended Support, vedere. Ripristino di un'istanza DB o di un cluster DB Multi-AZ, di un cluster Amazon RDS Extended Support

Console

Per ripristinare uno snapshot a un cluster di database Multi-AZ

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel pannello di navigazione, selezionare Snapshots (Snapshot).
3. Scegli la snapshot da usare per il ripristino.
4. Per Actions (Operazioni), selezionare Restore Snapshot (Ripristina snapshot).
5. Nella pagina Restore snapshots (Ripristina snapshot), in Availability and durability (Disponibilità e durabilità), scegliere Multi-AZ DB cluster (Cluster di database Multi-AZ).



6. Per Identificativo cluster database, specifica il nome del cluster database Multi-AZ ripristinato.
7. Per le restanti sezioni, specifica le impostazioni del cluster di database. Per informazioni su ciascuna impostazione, consulta Impostazioni per la creazione di cluster di database Multi-AZ.
8. Selezionare Ripristina istanza database.

AWS CLI

Per ripristinare un'istantanea in un cluster DB Multi-AZ, usa il AWS CLI comando -snapshot. restore-db-cluster-from

Nel seguente esempio il ripristino avviene da uno snapshot creato precedentemente e denominato `mysnapshot`. Viene ripristinato un nuovo cluster di database Multi-AZ denominato `mynewmultiazdbcluster`. È inoltre possibile specificare la classe di istanza database utilizzata dalle istanze database nel cluster di database Multi-AZ. Specificare `mysql` o `postgres` per il motore di database.

Per l'opzione `--snapshot-identifier`, è possibile utilizzare il nome o l'Amazon Resource Name (ARN) per specificare uno snapshot del cluster di database. Tuttavia, è possibile utilizzare solo l'ARN per specificare uno snapshot di database.

Per l'opzione `--db-cluster-instance-class`, specifica la classe di istanza database per il nuovo cluster database multi-AZ. I cluster database multi-AZ supportano solo classi di istanza database specifiche, come le classi di istanza database db.m6gd e db.r6gd. Per altre informazioni sulle classi di istanza database, consulta Classi di istanze database.

Puoi anche specificare altre opzioni.

Example

PerLinux, o: macOS Unix

```
aws rds restore-db-cluster-from-snapshot \
    --db-cluster-identifier mynewmultiazdbcluster \
    --snapshot-identifier mysnapshot \
    --engine mysql|postgres \
    --db-cluster-instance-class db.r6gd.xlarge
```

Per Windows:

```
aws rds restore-db-cluster-from-snapshot ^
    --db-cluster-identifier mynewmultiazdbcluster ^
    --snapshot-identifier mysnapshot ^
    --engine mysql|postgres ^
    --db-cluster-instance-class db.r6gd.xlarge
```

Dopo il ripristino del cluster database, puoi aggiungere il cluster database multi-AZ al gruppo di sicurezza associato all'istanza database o al cluster database usato per creare lo snapshot, se opportuno. In questo modo viene fornita la stessa funzionalità del cluster database o dell'istanza database precedente.

API RDS

Per ripristinare un'istantanea in un cluster DB Multi-AZ, chiamate l'operazione RDS API RestoreDB con i seguenti ClusterFromSnapshot parametri:

- `DBClusterIdentifier`
- `SnapshotIdentifier`
- `Engine`

Puoi inoltre specificare altri parametri facoltativi.

Dopo il ripristino del cluster database, puoi aggiungere il cluster database multi-AZ al gruppo di sicurezza associato all'istanza database o al cluster database usato per creare lo snapshot, se opportuno. In questo modo viene fornita la stessa funzionalità del cluster database o dell'istanza database precedente.

Ripristino da uno snapshot del cluster DB Multi-AZ a un'istanza DB Single-AZ

Uno snapshot di cluster database multi-AZ è uno snapshot dei volumi di archiviazione del cluster database con il backup dell'intero cluster database anziché dei singoli database. Puoi ripristinare uno snapshot di cluster database multi-AZ in un'implementazione single-AZ o a un'implementazione di istanza database multi-AZ. Per informazioni sulle implementazioni multi-AZ, consulta Configurazione e gestione di un'implementazione multi-AZ.

> **Note**
>
> È inoltre possibile ripristinare uno snapshot di cluster database multi-AZ in un nuovo cluster database multi-AZ. Per istruzioni, consulta Ripristino da uno snapshot a un cluster di database Multi-AZ.

Per informazioni sul ripristino di un cluster DB Multi-AZ con una versione RDS Extended Support, vedere. Ripristino di un'istanza DB o di un cluster DB Multi-AZ, di un cluster Amazon RDS Extended Support

Utilizza l'API AWS Management Console AWS CLI, the o RDS per ripristinare uno snapshot del cluster Multi-AZ DB in una distribuzione Single-AZ o in un'istanza DB Multi-AZ.

Console

Per ripristinare uno snapshot di cluster database multi-AZ in un'implementazione single-AZ o in un'implementazione di istanza database multi-AZ

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel pannello di navigazione, selezionare Snapshots (Snapshot).
3. Scegli lo snapshot di cluster database multi-AZ di cui vuoi eseguire il ripristino.
4. Per Actions (Operazioni), selezionare Restore Snapshot (Ripristina snapshot).
5. Nella pagina Restore snapshot (Ripristina snapshot), in Availability and durability (Disponibilità e durabilità), scegli una delle seguenti opzioni:

    - Single DB instance (Istanza database singola): ripristina lo snapshot in una sola istanza database senza istanza database in standby.

- Multi-AZ DB instance (Istanza database multi-AZ): ripristina lo snapshot in un'implementazione di istanza database multi-AZ con un'istanza database primaria e un'istanza database standby.

6. Per DB Instance Identifier (Identificatore di istanza database), immetti il nome dell'istanza database ripristinata.
7. Per le restanti sezioni, specifica le impostazioni dell'istanza database. Per informazioni su ciascuna impostazione, consulta Impostazioni per istanze database.
8. Selezionare Ripristina istanza database.

AWS CLI

Per ripristinare uno snapshot del cluster DB Multi-AZ in una distribuzione di istanza DB, usa il AWS CLI comando -db-snapshot. restore-db-instance-from

Nell'esempio seguente si esegue il ripristino di uno snapshot di cluster database multi-AZ creato in precedenza denominato `myclustersnapshot`. Viene ripristinato in una nuova implementazione di istanza database multi-AZ con un'istanza database primaria denominata `mynewdbinstance`. Per l'opzione `--db-cluster-snapshot-identifier`, specifica il nome dello snapshot di cluster database multi-AZ.

Per l'opzione `--db-instance-class`, specifica la classe di istanza database per la nuova implementazione di istanza database. Per altre informazioni sulle classi di istanza database, consulta Classi di istanze database.

Puoi anche specificare altre opzioni.

Example

Per, o: Linux macOS Unix

```
aws rds restore-db-instance-from-db-snapshot \
    --db-instance-identifier mynewdbinstance \
    --db-cluster-snapshot-identifier myclustersnapshot \
    --engine mysql \
    --multi-az \
    --db-instance-class db.r6g.xlarge
```

Per Windows:

```
aws rds restore-db-instance-from-db-snapshot ^
```

```
    --db-instance-identifier mynewdbinstance ^
    --db-cluster-snapshot-identifier myclustersnapshot ^
    --engine mysql ^
    --multi-az ^
    --db-instance-class db.r6g.xlarge
```

Dopo il ripristino dell'istanza database, puoi aggiungerla al gruppo di sicurezza associato al cluster database multi-AZ utilizzato per creare lo snapshot, se opportuno. In questo modo viene fornita la stessa funzionalità del cluster database multi-AZ precedente.

API RDS

Per ripristinare uno snapshot del cluster DB Multi-AZ su un'implementazione di istanza DB, chiama l'operazione API RDS RestoreDB InstanceFrom dbSnapshot con i seguenti parametri:

- `DBInstanceIdentifier`
- `DBClusterSnapshotIdentifier`
- `Engine`

Puoi inoltre specificare altri parametri facoltativi.

Dopo il ripristino dell'istanza database, puoi aggiungerla al gruppo di sicurezza associato al cluster database multi-AZ utilizzato per creare lo snapshot, se opportuno. In questo modo viene fornita la stessa funzionalità del cluster database multi-AZ precedente.

Tutorial: ripristino di un'istanza database Amazon RDS da uno snapshot DB

Spesso quando lavori con Amazon RDS hai un'istanza database che utilizzi occasionalmente e non a tempo pieno. Ad esempio, supponi di avere un sondaggio trimestrale sui clienti che utilizza un'istanza Amazon EC2 per ospitare un sito Web di sondaggi sui clienti. Hai anche un'istanza database che viene utilizzata per archiviare i risultati del sondaggio. Un modo per risparmiare denaro in uno scenario del genere è acquisire uno snapshot DB dell'istanza database dopo il completamento del sondaggio. Quindi elimini l'istanza database e la ripristini quando devi ripetere il sondaggio.

Quando ripristini l'istanza database, fornisci il nome dello snapshot DB da cui eseguire il ripristino. Quindi specifichi un nome per la nuova istanza database creata dall'operazione di ripristino.

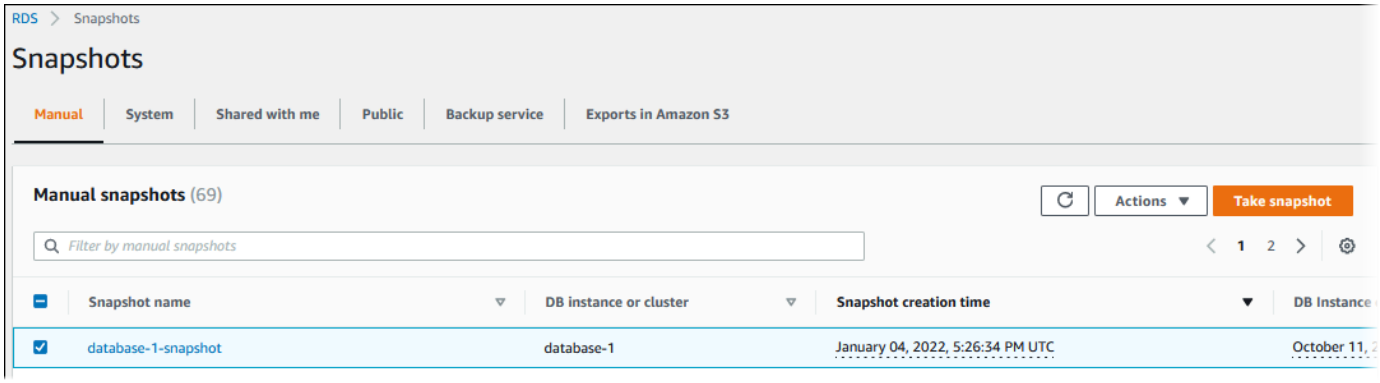
Per ulteriori informazioni sul ripristino di un'istanza database da uno snapshot, consulta Ripristino da uno snapshot database.

Ripristino di un'istanza database da uno snapshot DB

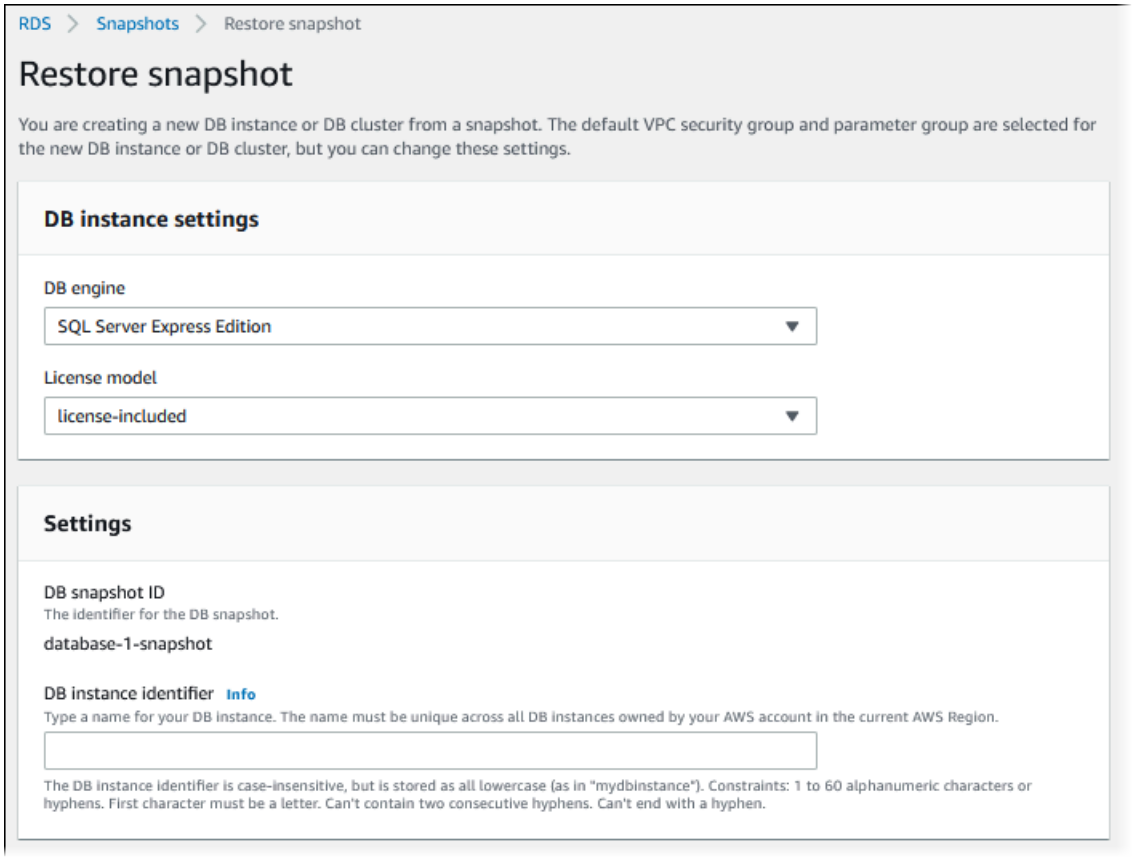
Procedi come segue per eseguire il ripristino da uno snapshot nella AWS Management Console.

Per ripristinare un'istanza database da uno snapshot DB

1. Accedi alla AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, selezionare Snapshots (Snapshot).
3. Scegliere la snapshot DB dalla quale effettuare il ripristino.
4. Per Actions (Operazioni), selezionare Restore Snapshot (Ripristina snapshot).



Viene visualizzata la pagina Restore snapshot (Ripristina snapshot).



5. Sotto DB instance settings (Impostazioni dell'istanza database), utilizza le impostazioni di default per DB engine (Motore del database) e License model (Modello di licenza) (per Oracle o Microsoft SQL Server).
6. Sotto Settings (Impostazioni), per DB instance identifier (Identificatore istanze DB) inserisci il nome univoco da usare per l'istanza database ripristinata, ad esempio **`mynewdbinstance`**.

   Se si sta eseguendo il ripristino da un'istanza database eliminata dopo aver creato lo snapshot DB, è possibile usare il nome dell'istanza database.
7. In Durabilità e disponibilità, scegli se creare un'istanza in standby in un'altra zona di disponibilità.

   Per questo tutorial, non creare un'istanza in standby.
8. Sotto Connectivity (Connettività), utilizza le impostazioni di default per quanto segue:

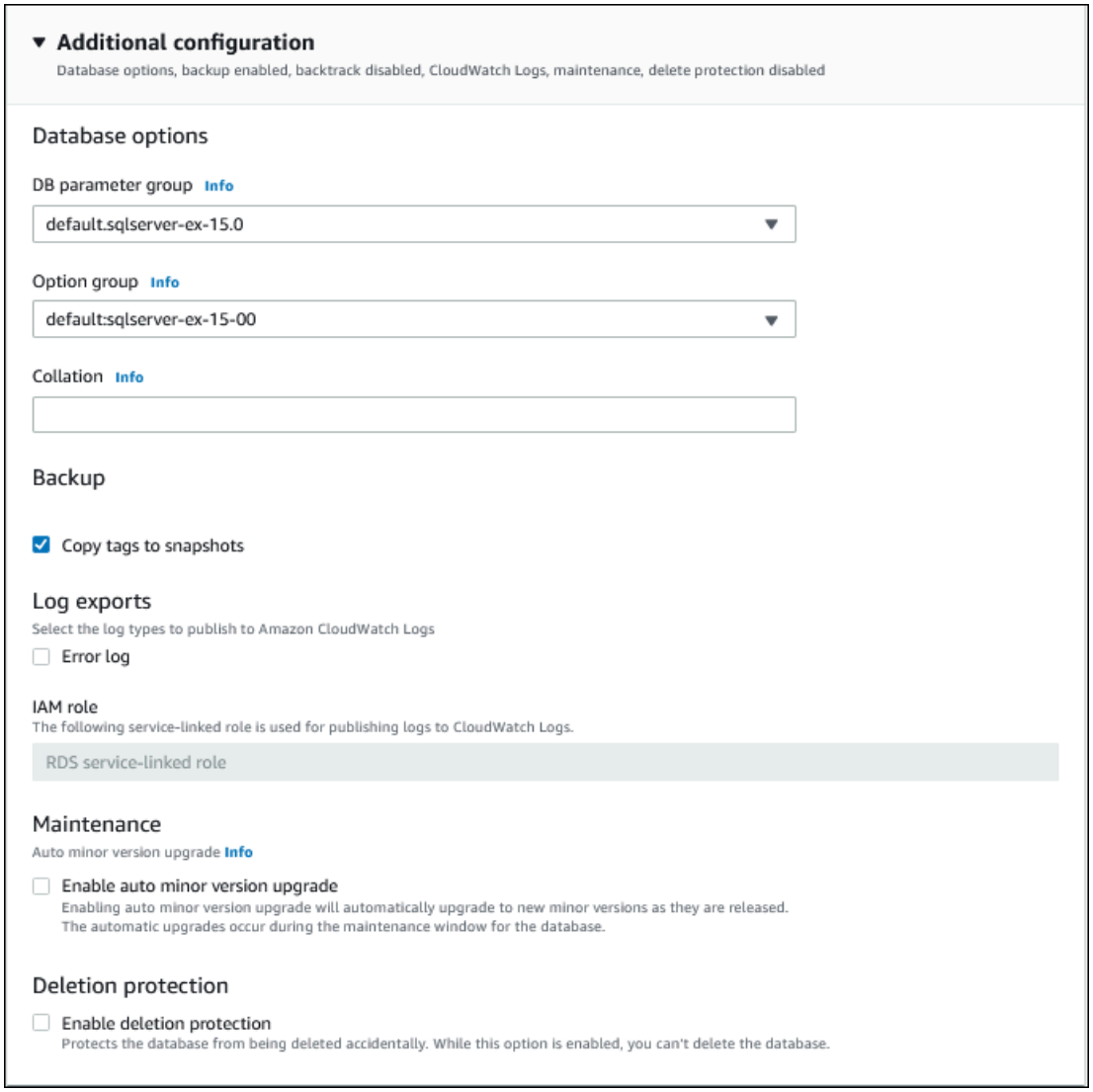
   - Virtual Private Cloud (VPC) (Cloud privato virtuale (VPC)
   - DB subnet group (Gruppo di sottoreti DB)
   - Accesso pubblico
   - VPC security group (firewall) (Gruppo di sicurezza VPC (firewall)
9. Scegli la DB instance class (Classe di istanza database).

Per questo tutorial, scegli Burstable classes (includes t classes) (Classi espandibili (include le classi t) e quindi scegli db.t3.small.

10. Per Encryption (Crittografia), utilizza le impostazioni di default.

    Se l'istanza database di origine per lo snapshot è stata crittografata, anche l'istanza database ripristinata viene crittografata. Non è possibile renderla non crittografato.

11. Espandi Additional configuration (Configurazione aggiuntiva) nella parte inferiore della pagina.



12. Esegui l'operazione seguente sotto Database options (Opzioni database):

    a. Seleziona il DB parameter group (Gruppo parametri del database).

       Per questo tutorial, utilizza il gruppo di parametri di default.

    b. Scegli Option group (Gruppo di opzioni).

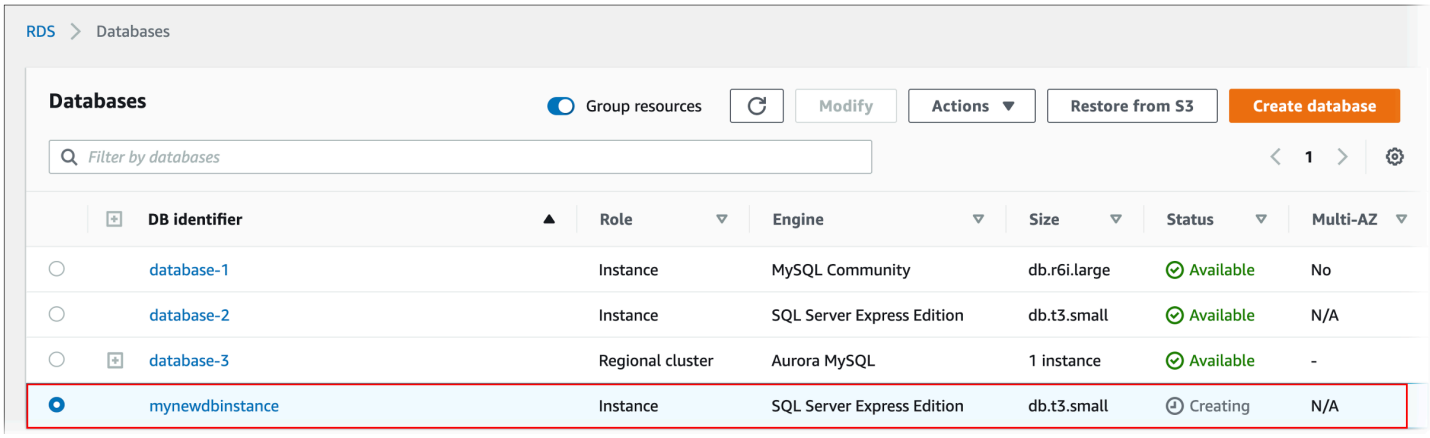
Per questo tutorial, utilizza il gruppo di opzioni di default.

> **Important**
>
> In alcuni casi, puoi eseguire il ripristino da uno snapshot DB di un'istanza database che utilizza un'opzione persistente o permanente. In tal caso, assicurati di scegliere un gruppo di opzioni che utilizzi la stessa opzione.

c. Per Deletion protection (Protezione da eliminazione), scegli la casella di controllo Enable deletion protection (Abilita protezione da eliminazione).

13. Selezionare Ripristina istanza database.

La pagina Databases (Database) visualizza l'istanza database ripristinata, con uno stato `Creating`.



Copia di una snapshot DB.

Con Amazon RDS è possibile copiare backup automatici o snapshot di database manuali. Dopo aver copiato uno snapshot, la copia è uno snapshot manuale. È possibile creare più copie di un backup automatico o di uno snapshot manuale, ma ogni copia deve avere un identificatore univoco.

È possibile copiare un'istantanea all'interno della stessa Regione AWS, copiare un'istantanea dall'altra Regioni AWS e copiare istantanee condivise.

Limitazioni

Di seguito sono riportate alcune limitazioni che si applicano quando si copiano le snapshot:

- Non puoi copiare uno snapshot nelle o dalle regioni Cina (Pechino) o Cina (Ningxia).
- È possibile copiare un'istantanea tra AWS GovCloud (Stati Uniti orientali) e (Stati Uniti occidentali). AWS GovCloud Tuttavia, non è possibile copiare un'istantanea tra queste regioni GovCloud (Stati Uniti) e regioni che non sono regioni GovCloud (Stati Uniti).
- Se elimini una snapshot origine prima che la snapshot target diventi disponibile, la copia della snapshot potrebbe non riuscire. Verifica che la snapshot target abbia lo stato di `AVAILABLE` prima di eliminare una snapshot origine.
- Puoi avere un massimo di 20 richieste di copia di snapshot in corso in una singola regione di destinazione per account.
- Quando vengono richieste più copie di snapshot per la stessa istanza database di origine, vengono accodate internamente. Le copie richieste in seguito non verranno avviate fino al completamento delle copie di snapshot precedenti. Per ulteriori informazioni, consulta Perché la creazione di snapshot EBS o AMI EC2 è lenta? nel Knowledge Center. AWS
- A seconda del soggetto Regioni AWS coinvolto e della quantità di dati da copiare, il completamento di una copia istantanea tra diverse regioni può richiedere ore. In alcuni casi, potrebbe esserci un numero elevato di richieste di copia di snapshot tra regioni da una determinata regione di origine. In questi casi, Amazon RDS potrebbe mettere in coda le richieste di copia tra regioni provenienti dalla regione di origine fino al completamento di alcune copie in corso. Nessuna informazione di progresso viene visualizzata sulle richieste di copia mentre sono in coda. Le informazioni sul progresso vengono visualizzate quando inizia la copia.
- Se una copia è ancora in sospeso quando si avvia un'altra copia, la seconda copia sarà avviata solo al termine della prima copia.
- Non è possibile copiare un'istantanea di un cluster DB Multi-AZ.

Conservare gli snapshot

Amazon RDS elimina i backup automatici in diverse situazioni:

- Al termine del periodo di conservazione.
- Quando si disabilitano i backup automatici per una istanza database.
- Quando si elimina una istanza database.

Se si desidera mantenere uno snapshot automatico per un periodo più lungo, è possbile copiarlo e creare uno snapshot manuale che sarà conservato finché non lo si elimina personalmente. I costi di archiviazione di Amazon RDS potrebbero applicarsi agli snapshot manuali se superano lo spazio di archiviazione predefinito.

Per ulteriori informazioni sui costi di storage dei backup, consulta Prezzi di Amazon RDS.

Copia di snapshot condivise

Puoi copiare istantanee condivise con te da altri. Account AWS In alcuni casi, è possibile copiare un'istantanea crittografata che è stata condivisa da un'altra persona. Account AWS In questi casi, è necessario avere accesso allo AWS KMS key strumento utilizzato per crittografare l'istantanea.

> **Note**
>
> I costi di storage di Amazon RDS si applicano agli snapshot condivisi che copi. Amazon RDS potrebbe allegare l'ARN dell'istanza DB di origine allo snapshot che hai copiato.

Puoi copiare uno snapshot DB condiviso Regioni AWS se lo snapshot non è crittografato. Tuttavia, se lo snapshot DB condiviso è crittografato, potrai copiarlo solo nella stessa regione.

> **Note**
>
> La copia di istantanee incrementali condivise nello stesso Regione AWS è supportata quando non sono crittografate o crittografate utilizzando la stessa chiave KMS dello snapshot completo iniziale. Se si utilizza una chiave del servizio di gestione delle chiavi diversa per crittografare le istantanee successive durante la copia, tali snapshot condivisi sono istantanee complete. Per ulteriori informazioni, consulta Copia snapshot incrementale.

Gestione della crittografia

Puoi copiare una snapshot che è stata crittografata utilizzando una chiave KMS. Se la copia di una snapshot crittografata, la copia della snapshot deve anche essere crittografata. Se copi un'istantanea crittografata all'interno della stessa Regione AWS, puoi crittografare la copia con la stessa chiave KMS dell'istantanea originale. Oppure puoi specificare una chiave KMS diversa.

Se copi uno snapshot crittografato tra regioni, devi specificare una chiave KMS valida nella Regione AWS di destinazione. Può essere una chiave KMS specifica per la regione o una chiave multi-regione. Per ulteriori informazioni sulle chiavi multi-regione, consulta Utilizzo delle chiavi multi-regione in AWS KMS.

La snapshot di origine resta crittografata nel processo di copia. Per ulteriori informazioni, consulta Limiti relativi a cluster di database crittografate Amazon RDS.

Puoi anche crittografare una copia di una snapshot crittografata. In questo modo, puoi aggiungere rapidamente la crittografia a un'istanza database non crittografata. Ciò significa che puoi creare uno snapshot dell'istanza database quando sei pronto per la crittografia. È quindi possibile creare una copia di tale snapshot e specificare una chiave KMS per crittografare lo snapshot. Puoi quindi ripristinare un'istanza database crittografata dalla snapshot crittografata.

Copia snapshot incrementale

Una snapshot incrementale contiene solo i dati modificati dopo la snapshot più recente della stessa istanza database. La copia di snapshot incrementali è più rapida e comporta costi di archiviazione inferiori rispetto alla copia di snapshot complete.

Il fatto che una copia dell'istantanea sia incrementale è determinato dall'ultima copia dell'istantanea completata e dallo snapshot di origine. Se la copia snapshot più recente è stata eliminata, la copia successiva è una copia completa, non una copia incrementale. Una copia istantanea sarà dello stesso tipo dell'istantanea di origine. Se l'istantanea di origine è un'istantanea incrementale, la copia dell'istantanea sarà un'istantanea incrementale.

Quando si copia un'istantanea Account AWS, la copia è una copia incrementale solo se sono soddisfatte tutte le seguenti condizioni:

- La copia istantanea più recente è della stessa istanza DB di origine ed esiste ancora nell'account di destinazione.
- Tutte le copie dello snapshot nell'account di destinazione o non sono crittografate o sono state crittografate utilizzando la stessa chiave KMS.

- Se l'istanza database di origine è un'istanza Multi-AZ, non ha eseguito il failover su un'altra istanza AZ da quando è stato acquisito l'ultimo snapshot.

Negli esempi seguenti viene illustrata la differenza tra snapshot completi e incrementali. Si applicano sia alle istantanee condivise che a quelle non condivise.

Snapshot	Chiave di crittografia	Completo o incrementale
S1	K1	Full
S2	K1	Incrementale di S1
S3	K1	Incrementale di S2
S4	K1	Incrementale di S3
Copia di S1 (S1C)	K2	Full
Copia di S2 (S2C)	K3	Full
Copia di S3 (S3C)	K3	Incrementale di S2C
Copia di S4 (S4C)	K3	Incrementale di S3C
Copia 2 di S4 (S4C2)	K4	Full

> **Note**
>
> In questi esempi, gli snapshot S2, S3 e S4 sono incrementali solo se lo snapshot precedente esiste ancora.
> Lo stesso vale per le copie. Le copie degli snapshot S3C e S4C sono incrementali solo se la copia precedente esiste ancora.

Per informazioni sulla copia di istantanee incrementali su più file, consulta. Regioni AWSCopie complete e incrementali

Copia di snapshot tra regioni

È possibile copiare snapshot di database tra Regioni AWS. Tuttavia, esistono alcuni vincoli e considerazioni per la copia di snapshot tra le regioni.

Richiesta di una copia di snapshot DB tra regioni

Per comunicare con la regione di origine per richiedere una copia di snapshot DB tra regioni, il richiedente (ruolo IAM o utente IAM) deve avere accesso allo snapshot DB di origine e alla regione di origine.

Alcune condizioni nella policy IAM del richiedente possono causare l'esito negativo della richiesta. Negli esempi seguenti si assume che si stia copiando lo snapshot DB da Stati Uniti orientali (Ohio) a US East (N. Virginia) . Questi esempi mostrano le condizioni nella policy IAM del richiedente che causano l'esito negativo della richiesta:

- La policy del richiedente ha una condizione per `aws:RequestedRegion`.

```
...
"Effect": "Allow",
"Action": "rds:CopyDBSnapshot",
"Resource": "*",
"Condition": {
    "StringEquals": {
        "aws:RequestedRegion": "us-east-1"
    }
}
```

La richiesta ha esito negativo perché la policy non consente l'accesso alla regione di origine. Perché una richiesta sia completata correttamente, specifica sia le regioni di origine che quelle di destinazione.

```
...
"Effect": "Allow",
"Action": "rds:CopyDBSnapshot",
"Resource": "*",
"Condition": {
    "StringEquals": {
        "aws:RequestedRegion": [
            "us-east-1",
            "us-east-2"
```

```
        ]
    }
}
```

- La policy del richiedente non consente l'accesso allo snapshot DB di origine.

```
...
"Effect": "Allow",
"Action": "rds:CopyDBSnapshot",
"Resource": "arn:aws:rds:us-east-1:123456789012:snapshot:target-snapshot"
...
```

Perché una richiesta sia completata correttamente, specifica sia gli snapshot di origine che quelli di destinazione.

```
...
"Effect": "Allow",
"Action": "rds:CopyDBSnapshot",
"Resource": [
    "arn:aws:rds:us-east-1:123456789012:snapshot:target-snapshot",
    "arn:aws:rds:us-east-2:123456789012:snapshot:source-snapshot"
]
...
```

- La policy del richiedente rifiuta `aws:ViaAWSService`.

```
...
"Effect": "Allow",
"Action": "rds:CopyDBSnapshot",
"Resource": "*",
"Condition": {
    "Bool": {"aws:ViaAWSService": "false"}
}
```

La comunicazione con la regione di origine viene effettuata da RDS per conto del richiedente. Per una richiesta andata a buon fine, non negare le chiamate effettuate dai servizi. AWS

- La policy del richiedente ha una condizione per `aws:SourceVpc` o `aws:SourceVpce`.

Queste richieste potrebbero non riuscire perché quando RDS effettua la chiamata alla regione remota, non proviene dall'endpoint VPC o dal VPC specificato.

Se è necessario utilizzare una delle condizioni precedenti che causerebbero un errore di una richiesta, è possibile includere una seconda istruzione con `aws:CalledVia` nella policy in modo che la richiesta abbia esito positivo. Ad esempio, è possibile utilizzare `aws:CalledVia` con `aws:SourceVpce` come riportato di seguito:

```
...
"Effect": "Allow",
"Action": "rds:CopyDBSnapshot",
"Resource": "*",
"Condition": {
    "Condition" : {
        "ForAnyValue:StringEquals" : {
          "aws:SourceVpce": "vpce-1a2b3c4d"
        }
     }
},
{
    "Effect": "Allow",
    "Action": [
        "rds:CopyDBSnapshot"
    ],
    "Resource": "*",
    "Condition": {
        "ForAnyValue:StringEquals": {
            "aws:CalledVia": [
                "rds.amazonaws.com"
            ]
        }
    }
}
```

Per ulteriori informazioni, consulta Policy e autorizzazioni in IAM nella Guida per l'utente di IAM.

Autorizzazione della copia di snapshot

Dopo una richiesta di copia di snapshot DB tra regioni restituisce `success`, RDS avvia la copia in background. Viene creata un'autorizzazione per RDS per accedere allo snapshot di origine. Questa autorizzazione collega lo snapshot DB di origine allo snapshot DB di destinazione e consente a RDS di copiare solo lo snapshot di destinazione specificato.

L'autorizzazione è verificata da RDS utilizzando l'autorizzazione `rds:CrossRegionCommunication` nel ruolo IAM collegato al servizio. Se la copia è autorizzata, RDS comunica con la regione di origine e completa la copia.

RDS non ha accesso agli snapshot DB che non erano state autorizzati in precedenza da una richiesta `CopyDBSnapshot`. L'autorizzazione viene revocata al completamento della copia.

RDS utilizza il ruolo collegato al servizio per verificare l'autorizzazione nella regione di origine. Se si elimina il ruolo collegato al servizio durante il processo di copia, la copia avrà esito negativo.

Per ulteriori informazioni, consulta Utilizzo di ruoli collegati ai servizi nella Guida per l'utente di IAM.

Utilizzo delle credenziali AWS Security Token Service

I token di sessione dell'endpoint global AWS Security Token Service (AWS STS) sono validi solo se abilitati per impostazione predefinita (regioni commerciali). Regioni AWS Se utilizzi le credenziali dell'operazione `assumeRole` API in AWS STS, utilizza l'endpoint regionale se la regione di origine è una regione che richiede l'attivazione. In caso contrario, la richiesta ha esito negativo. Ciò accade perché le credenziali devono essere valide in entrambe le regioni, il che vale per le regioni che hanno aderito solo quando viene utilizzato l'endpoint regionale. AWS STS

Per utilizzare l'endpoint globale, assicurarsi che sia abilitato per entrambe le regioni nelle operazioni. Imposta l'endpoint globale su `Valid in all Regioni AWS` nelle impostazioni dell'account. AWS STS

La stessa regola si applica alle credenziali nel parametro URL prefirmato.

Per ulteriori informazioni, consulta Managing AWS STS nella Guida per l'utente di IAM. Regione AWS

Latenza e richieste di copia multiple

A seconda del soggetto Regioni AWS coinvolto e della quantità di dati da copiare, il completamento di una copia istantanea tra diverse regioni può richiedere ore.

In alcuni casi, potrebbe esserci un numero elevato di richieste di copia di snapshot tra regioni da una determinata Regione AWS di origine. In questi casi, Amazon RDS potrebbe mettere in coda nuove richieste di copia interregionali provenienti da tale fonte fino al completamento di alcune copie Regione AWS in corso. Nessuna informazione di progresso viene visualizzata sulle richieste di copia mentre sono in coda. Le informazioni sul progresso vengono visualizzate quando inizia la copia.

Copie complete e incrementali

Quando si copia uno snapshot in uno snapshot diverso Regione AWS da quello di origine, la prima copia è una copia istantanea completa, anche se si copia uno snapshot incrementale. Una copia snapshot completa contiene tutti i dati e i metadati necessari per archiviare l'istanza database. Dopo la prima copia dello snapshot, è possibile copiare istantanee incrementali della stessa istanza DB nella stessa regione di destinazione all'interno della stessa. Account AWS Per ulteriori informazioni sugli snapshot incrementali, vedere Copia snapshot incrementale.

La copia incrementale delle istantanee Regioni AWS è supportata sia per le istantanee non crittografate che per quelle crittografate.

Quando si copia un'istantanea Regioni AWS, la copia è una copia incrementale se sono soddisfatte le seguenti condizioni:

- Lo snapshot è stato precedentemente copiato nella regione di destinazione.
- La copia snapshot più recente esiste ancora nella regione di destinazione.
- Tutte le copie dello snapshot nella Regione di destinazione sono non crittografate o sono state crittografate utilizzando la stessa chiave KMS.

Considerazioni su gruppi di opzioni

I gruppi di opzioni del database sono specifici del Regione AWS tipo in cui vengono creati e non è possibile utilizzare un gruppo di opzioni proveniente da uno Regione AWS all'altro. Regione AWS

Per i database Oracle, puoi utilizzare l'API AWS CLI o RDS per copiare il gruppo di opzioni DB personalizzato da uno snapshot che è stato condiviso con il tuo. Account AWSÈ possibile copiare i gruppi di opzioni solo all'interno della stessa Regione AWS. Il gruppo di opzioni non viene copiato se è già stato copiato nell'account di destinazione e non è stata apportata alcuna modifica dopo la copia. Se il gruppo di opzioni di origine è stato copiato in precedenza, ma è stato modificato dopo la copia, RDS copia la nuova versione nell'account di destinazione. I gruppi di opzioni predefiniti non vengono copiati.

Quando copi uno snapshot tra regioni, puoi specificare un nuovo gruppo di opzione per lo snapshot. Consigliamo di preparare un nuovo gruppo di opzioni prima di copiare la snapshot. Nella destinazione Regione AWS, crea un gruppo di opzioni con le stesse impostazioni dell'istanza DB originale. Se ne esiste già uno nella nuova Regione AWS, puoi usare quello.

In alcuni casi, è possibile copiare uno snapshot e non specificare un nuovo gruppo di opzioni per lo snapshot. In questi casi, quando si ripristina lo snapshot l'istanza database ottiene il gruppo di opzioni predefinito. Per assegnare alla nuova istanza al nuovo cluster di database le stesse opzioni dell'originale, completa la seguente procedura:

1. Nella destinazione Regione AWS, create un gruppo di opzioni con le stesse impostazioni dell'istanza DB originale. Se ne esiste già uno nella nuova Regione AWS, puoi usare quello.
2. Dopo aver ripristinato l'istantanea nella destinazione Regione AWS, modifica la nuova istanza DB e aggiungi il gruppo di opzioni nuovo o esistente del passaggio precedente.

Considerazioni sui gruppi di parametri

Quando copi uno snapshot tra regioni, la copia non include il gruppo di parametri usato dall'istanza database originale. Quando ripristini uno snapshot per creare una nuova istanza DB, a quell'istanza DB viene assegnato il gruppo di parametri predefinito in cui Regione AWS è stata creata. Per assegnare alla nuova istanza database gli stessi parametri dell'originale, completa la seguente procedura:

1. Nella destinazione Regione AWS, create un gruppo di parametri DB con le stesse impostazioni dell'istanza DB originale. Se ne esiste già uno nella nuova Regione AWS, puoi usare quello.
2. Dopo aver ripristinato l'istantanea nella destinazione Regione AWS, modifica la nuova istanza DB e aggiungi il gruppo di parametri nuovo o esistente del passaggio precedente.

Copia di una snapshot DB.

Utilizza le procedure in questo argomento, per copiare una snapshot DB. Per una panoramica su come copiare una snapshot, consulta Copia di una snapshot DB.

Per ognuna di esse Account AWS, è possibile copiare fino a 20 istantanee DB alla volta da una Regione AWS all'altra. Se si copia un'istantanea del DB su un'altra Regione AWS, si crea un'istantanea del DB manuale che viene conservata in quella copia. Regione AWS La copia di uno snapshot DB dall'origine comporta costi di Regione AWS trasferimento dati Amazon RDS.

Per ulteriori informazioni sui prezzi del trasferimento dati, consulta Prezzi di Amazon RDS.

Dopo che la copia dello snapshot DB è stata creata nel nuovo database Regione AWS, la copia dello snapshot DB si comporta come tutte le altre snapshot DB in essa contenute. Regione AWS

È possibile copiare uno snapshot DB utilizzando AWS Management Console, the o l' AWS CLI API RDS.

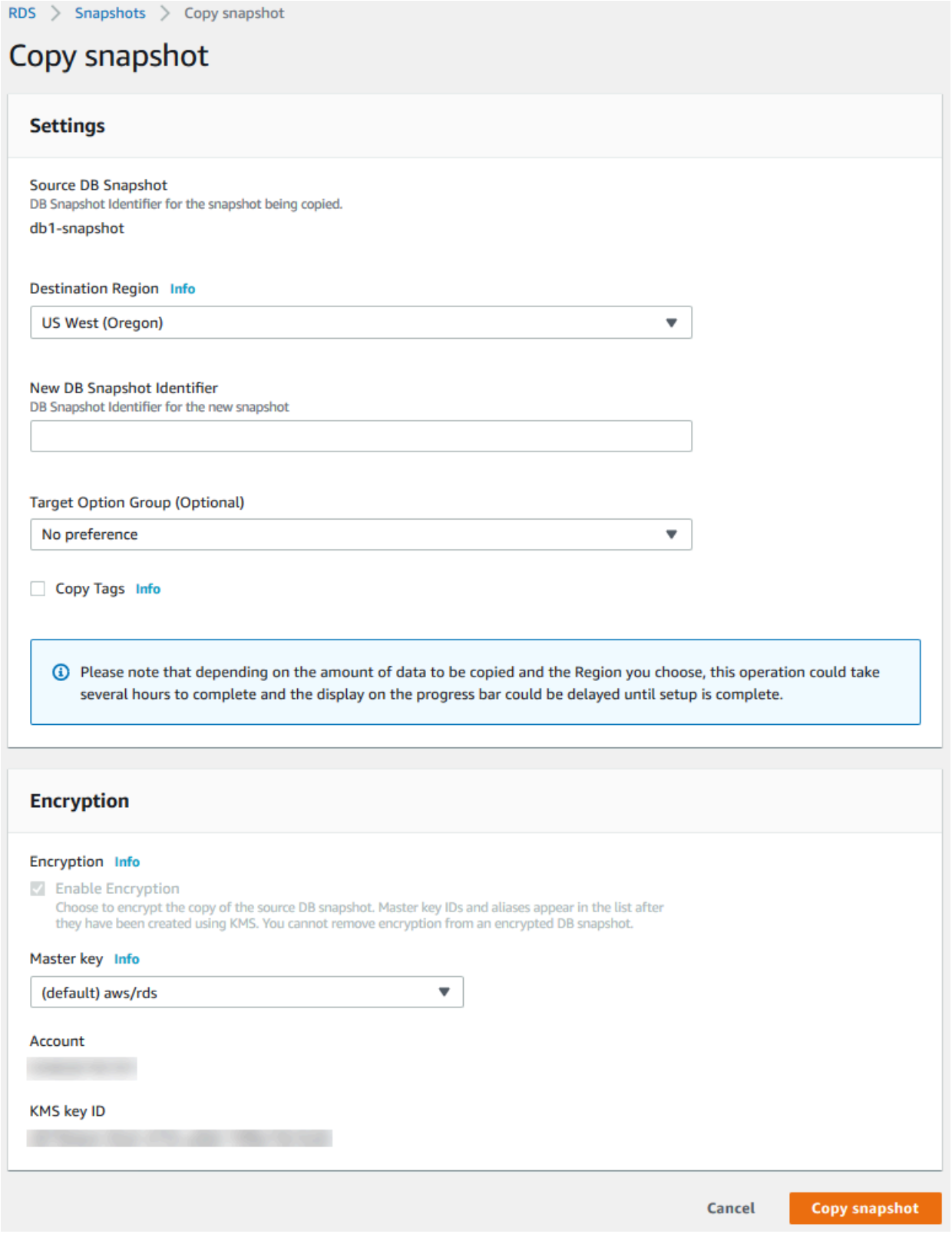
Console

La procedura seguente copia uno snapshot DB crittografato o non crittografato, nella stessa regione Regione AWS o in più regioni, utilizzando. AWS Management Console

Per copiare una snapshot DB

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, selezionare Snapshots (Snapshot).
3. Seleziona la snapshot DB che desideri copiare.
4. In Operazioni, seleziona Copia snapshot.

   Viene visualizzata la pagina Copia snapshot.



5. Per Target Option Group (optional) (Gruppo di opzioni di destinazione (facoltativo)), seleziona un nuovo gruppo di opzioni.

Specificate questa opzione se state copiando uno snapshot da uno Regione AWS all'altro e l'istanza DB utilizza un gruppo di opzioni non predefinito.

Se l'istanza database di origine utilizza Transparent Data Encryption per il server Oracle o Microsoft SQL, devi specificare questa opzione quando esegui la copia tra regioni. Per ulteriori informazioni, consulta Considerazioni su gruppi di opzioni.

6. (Facoltativo) Per copiare lo snapshot del DB in un'altra Regione AWS, per Regione di destinazione, scegli la nuova. Regione AWS

> **Note**
>
> La destinazione Regione AWS deve avere la stessa versione del motore di database disponibile come origine Regione AWS.

7. Per New DB Snapshot Identifier (Nuovo identificatore snapshot database), digita il nome della copia dello snapshot di database.

   È possibile creare più copie di un backup automatico o di uno snapshot manuale, ma ogni copia deve avere un identificatore univoco.

8. (Facoltativo) Seleziona Copy Tags (Copia tag) per copiare i tag e i valori dalla snapshot alla copia della snapshot.

9. (Facoltativo) Per Crittografia, effettuare le seguenti operazioni:

   a. Scegli Abilita crittografia se lo snapshot DB non è crittografato, ma desideri crittografare la copia.

   > **Note**
   >
   > Se lo snapshot DB è crittografato, è necessario crittografare la copia, quindi la casella di controllo è già selezionata.

   b. Per AWS KMS key, specifica l'identificatore di chiave KMS da utilizzare per crittografare la copia di snapshot DB.

10. Selezionare Copy Snapshot (Copia snapshot).

AWS CLI

È possibile copiare un'istantanea del DB utilizzando il AWS CLI comando copy-db-snapshot. Se stai copiando l'istantanea in una nuova Regione AWS, esegui il comando nella nuova. Regione AWS

Le seguenti opzioni vengono utilizzate per copiare una snapshot DB. Non tutte le opzioni sono necessarie per tutti gli scenari. Utilizza le descrizioni e gli esempi che seguono per determinare quali opzioni utilizzare.

- `--source-db-snapshot-identifier` – Identificatore per lo snapshot DB origine.
  - Se lo snapshot di origine è lo stesso Regione AWS della copia, specificate un identificatore DB snapshot valido. Ad esempio, `rds:mysql-instance1-snapshot-20130805`.
  - Se lo snapshot di origine è nella Regione AWS stessa copia ed è stato condiviso con il tuo Account AWS, specifica un ARN di snapshot DB valido. Ad esempio, `arn:aws:rds:us-west-2:123456789012:snapshot:mysql-instance1-snapshot-20130805`.
  - Se lo snapshot di origine si trova in un formato Regione AWS diverso da quello della copia, specificare un ARN di snapshot DB valido. Ad esempio, `arn:aws:rds:us-west-2:123456789012:snapshot:mysql-instance1-snapshot-20130805`.
  - Se stai copiando da uno snapshot DB manuale condiviso, questo parametro deve essere l'Amazon Resource Name (ARN) della snapshot DB condivisa.
  - Se si copia un'istantanea crittografata, questo parametro deve essere nel formato ARN per l'origine Regione AWS e deve corrispondere a quello nel `SourceDBSnapshotIdentifier` parametro. `PreSignedUrl`
- `--target-db-snapshot-identifier` – Identificatore per la nuova copia dello snapshot DB crittografato.
- `--copy-option-group` – Copia il gruppo di opzioni da uno snapshot che è stato condiviso con il tuo Account AWS.
- `--copy-tags` – Includi l'opzione di copia dei tag e i valori dello snapshot nella copia dello snapshot.
- `--option-group-name` – Gruppo di opzioni da associare alla copia dello snapshot.

  Specificate questa opzione se state copiando uno snapshot da uno Regione AWS all'altro e l'istanza DB utilizza un gruppo di opzioni non predefinito.

  Se l'istanza database di origine utilizza Transparent Data Encryption per il server Oracle o Microsoft SQL, devi specificare questa opzione quando esegui la copia tra regioni. Per ulteriori informazioni, consulta Considerazioni su gruppi di opzioni.

- `--kms-key-id`: l'identificatore di chiave KMS per uno snapshot DB crittografato. L'identificatore della chiave KMS è il nome della risorsa Amazon Resource Name (ARN), l'identificatore della chiave o l'alias della chiave per la chiave KMS.
  - Se copi uno snapshot DB crittografato dal tuo Account AWS, puoi specificare un valore per questo parametro per crittografare la copia con una nuova chiave KMS. Se non specifichi un valore per questo parametro, la copia della snapshot DB viene crittografata con la stessa chiave KMS della snapshot DB origine.
  - Se copi un'istantanea DB crittografata condivisa da un'altra Account AWS, devi specificare un valore per questo parametro.
  - Se specifichi questo parametro quando copi una snapshot crittografata, la copia viene crittografata.
  - Se copi un'istantanea crittografata in un'altra Regione AWS, devi specificare una chiave KMS per la destinazione. Regione AWS Le chiavi KMS sono specifiche del sistema in Regione AWS cui vengono create e non è possibile utilizzare le chiavi di crittografia l'una nell'altra Regione AWS . Regione AWS

Example Da non crittografata, alla stessa regione

Il codice seguente crea una copia di un'istantanea, con il nuovo nome`mydbsnapshotcopy`, nello stesso dello snapshot Regione AWS di origine. Quando viene creata la copia, il gruppo di opzioni database e i tag dello snapshot originale vengono copiati nella copia dello snapshot.

PerLinux, omacOS: Unix

```
aws rds copy-db-snapshot \
    --source-db-snapshot-identifier arn:aws:rds:us-west-2:123456789012:snapshot:mysql-instance1-snapshot-20130805 \
    --target-db-snapshot-identifier mydbsnapshotcopy \
    --copy-option-group \
    --copy-tags
```

Per Windows:

```
aws rds copy-db-snapshot ^
    --source-db-snapshot-identifier arn:aws:rds:us-west-2:123456789012:snapshot:mysql-instance1-snapshot-20130805 ^
    --target-db-snapshot-identifier mydbsnapshotcopy ^
    --copy-option-group ^
```

```
    --copy-tags
```

Example Da non crittografata, tra regioni

Il codice seguente crea una copia di un'istantanea, con il nuovo nome`mydbsnapshotcopy`, Regione AWS nella quale viene eseguito il comando.

Per LinuxmacOS, oUnix:

```
aws rds copy-db-snapshot \
    --source-db-snapshot-identifier arn:aws:rds:us-east-1:123456789012:snapshot:mysql-instance1-snapshot-20130805 \
    --target-db-snapshot-identifier mydbsnapshotcopy
```

Per Windows:

```
aws rds copy-db-snapshot ^
    --source-db-snapshot-identifier arn:aws:rds:us-east-1:123456789012:snapshot:mysql-instance1-snapshot-20130805 ^
    --target-db-snapshot-identifier mydbsnapshotcopy
```

Example Da crittografata, tra regioni

Nell'esempio di codice riportato di seguito viene copiata uno snapshot DB crittografato dalla regione Stati Uniti occidentali (Oregon) alla regione US East (N. Virginia). Emetti il comando nella regione di destinazione (us-est-1).

Per LinuxmacOS, oUnix:

```
aws rds copy-db-snapshot \
    --source-db-snapshot-identifier arn:aws:rds:us-west-2:123456789012:snapshot:mysql-instance1-snapshot-20161115 \
    --target-db-snapshot-identifier mydbsnapshotcopy \
    --kms-key-id my-us-east-1-key \
    --option-group-name custom-option-group-name
```

Per Windows:

```
aws rds copy-db-snapshot ^
```

```
    --source-db-snapshot-identifier arn:aws:rds:us-west-2:123456789012:snapshot:mysql-
instance1-snapshot-20161115 ^
    --target-db-snapshot-identifier mydbsnapshotcopy ^
    --kms-key-id my-us-east-1-key ^
    --option-group-name custom-option-group-name
```

Il `--source-region` parametro è obbligatorio quando si copia un'istantanea crittografata tra le regioni AWS GovCloud (Stati Uniti orientali) e AWS GovCloud (Stati Uniti occidentali). Per `--source-region`, specificare la Regione AWS dell'istanza database di origine.

Se non si specifica `--source-region`, è necessario specificare un valore per `--pre-signed-url`. Un URL prefirmato è un URL che contiene una richiesta firmata Signature Version 4 per il comando `copy-db-snapshot` chiamato nella Regione AWS di origine. Per ulteriori informazioni sull'*`pre-signed-url`*opzione, consulta copy-db-snapshotla sezione Command Reference.AWS CLI

API RDS

Puoi copiare uno snapshot DB usando l'operazione API di Amazon RDS CopyDBSnapshot. Se stai copiando l'istantanea su una nuova Regione AWS, esegui l'azione nella nuova. Regione AWS

I seguenti parametri vengono utilizzati per copiare una snapshot DB. Non tutti parametri sono necessari per tutti gli scenari. Utilizza le descrizioni e gli esempi che seguono per determinare quali parametri utilizzare.

- `SourceDBSnapshotIdentifier` – Identificatore per lo snapshot DB origine.
  - Se lo snapshot di origine è Regione AWS uguale alla copia, specificate un identificatore DB snapshot valido. Ad esempio, `rds:mysql-instance1-snapshot-20130805`.
  - Se lo snapshot di origine è nella Regione AWS stessa copia ed è stato condiviso con il tuo Account AWS, specifica un ARN di snapshot DB valido. Ad esempio, `arn:aws:rds:us-west-2:123456789012:snapshot:mysql-instance1-snapshot-20130805`.
  - Se lo snapshot di origine si trova in un formato Regione AWS diverso da quello della copia, specificare un ARN di snapshot DB valido. Ad esempio, `arn:aws:rds:us-west-2:123456789012:snapshot:mysql-instance1-snapshot-20130805`.
  - Se stai copiando da uno snapshot DB manuale condiviso, questo parametro deve essere l'Amazon Resource Name (ARN) della snapshot DB condivisa.
  - Se si copia un'istantanea crittografata, questo parametro deve essere nel formato ARN per l'origine Regione AWS e deve corrispondere a quello nel `SourceDBSnapshotIdentifier` parametro. `PreSignedUrl`

- `TargetDBSnapshotIdentifier` – Identificatore per la nuova copia dello snapshot DB crittografato.
- `CopyOptionGroup` – Imposta questo parametro su `true` per copiare il gruppo di opzioni dallo snapshot condiviso alla copia dello snapshot. Il valore predefinito è `false`.
- `CopyTags` – Imposta questo parametro su `true` per copiare i tag e i valori dallo snapshot alla copia dello snapshot. Il valore di default è `false`.
- `OptionGroupName` – Gruppo di opzioni da associare alla copia dello snapshot.

  Specificate questo parametro se state copiando uno snapshot da uno Regione AWS all'altro e l'istanza DB utilizza un gruppo di opzioni non predefinito.

  Se l'istanza database di origine utilizza Transparent Data Encryption per il server Oracle o Microsoft SQL, devi specificare questo parametro quando esegui la copia tra regioni. Per ulteriori informazioni, consulta Considerazioni su gruppi di opzioni.
- `KmsKeyId`: l'identificatore di chiave KMS per uno snapshot DB crittografato. L'identificatore della chiave KMS è il nome della risorsa Amazon Resource Name (ARN), l'identificatore della chiave o l'alias della chiave per la chiave KMS.
  - Se copi uno snapshot DB crittografato dal tuo Account AWS, puoi specificare un valore per questo parametro per crittografare la copia con una nuova chiave KMS. Se non specifichi un valore per questo parametro, la copia della snapshot DB viene crittografata con la stessa chiave KMS della snapshot DB origine.
  - Se copi un'istantanea DB crittografata condivisa da un'altra Account AWS, devi specificare un valore per questo parametro.
  - Se specifichi questo parametro quando copi una snapshot crittografata, la copia viene crittografata.
  - Se copi un'istantanea crittografata in un'altra Regione AWS, devi specificare una chiave KMS per la destinazione. Regione AWS Le chiavi KMS sono specifiche del sistema in Regione AWS cui vengono create e non è possibile utilizzare le chiavi di crittografia l'una nell'altra Regione AWS . Regione AWS
- `PreSignedUrl`— L'URL che contiene una richiesta firmata Signature Version 4 per l'operazione `CopyDBSnapshot` API nell'origine Regione AWS che contiene lo snapshot del DB di origine da copiare.

  Specificare questo parametro quando si copia uno snapshot DB crittografato Regione AWS da un altro utilizzando l'API Amazon RDS. Puoi specificare l'opzione della regione di origine anziché

questo parametro quando copi uno snapshot di database crittografato da un'altra Regione AWS utilizzando la AWS CLI.

L'URL prefirmato deve essere una richiesta valida per l'operazione API `CopyDBSnapshot` che può essere eseguita nella Regione AWS di origine che contiene lo snapshot di database crittografato da copiare. La richiesta URL prefirmata deve contenere i seguenti valori di parametro:

- `DestinationRegion`— Il file in Regione AWS cui verrà copiato lo snapshot DB crittografato. Regione AWS È la stessa in cui viene chiamata l'`CopyDBSnapshot`operazione che contiene questo URL predefinito.

  Si assuma, ad esempio, di copiare uno snapshot DB crittografato dalla regione us-west-2 nella regione us-est-1. È quindi necessario richiamare l'operazione `CopyDBSnapshot` nella regione us-east-1 e fornire un URL prefirmato che contenga una chiamata all'operazione `CopyDBSnapshot` nella regione us-west-2. In questo esempio, `DestinationRegion` nell'URL prefirmato deve essere impostato sulla regione Stati Uniti orientali 1.
- `KmsKeyId`: l'identificatore della chiave KMS per la chiave da utilizzare per crittografare la copia dello snapshot di database nella Regione AWS di destinazione. Si tratta dello stesso identificatore sia per l'`CopyDBSnapshot`operazione chiamata nella destinazione Regione AWS sia per l'operazione contenuta nell'URL predefinito.
- `SourceDBSnapshotIdentifier` – L'identificatore di snapshot DB per lo snapshot crittografato da copiare. L'identificatore deve essere nel formato Amazon Resource Name (ARN) per la Regione AWS di origine. Ad esempio, se stai copiando uno snapshot DB crittografato dalla regione us-west-2, il tuo sarà `SourceDBSnapshotIdentifier` simile al seguente esempio:. `arn:aws:rds:us-west-2:123456789012:snapshot:mysql-instance1-snapshot-20161115`

Per ulteriori informazioni sulle richieste firmate Signature Version 4, consulta quanto segue:

- Autenticazione delle richieste: utilizzo dei parametri di query (versione 4 AWS della firma) nell'Amazon Simple Storage Service API Reference
- Processo di firma della versione 4 di Signature in Riferimenti generali di AWS

Example Da non crittografata, alla stessa regione

Il codice seguente crea una copia di un'istantanea, con il nuovo nome`mydbsnapshotcopy`, con lo stesso Regione AWS nome dell'istantanea di origine. Quando viene creata la copia, tutti i tag della snapshot originale vengono copiati nella copia della snapshot.

```
https://rds.us-west-1.amazonaws.com/
 ?Action=CopyDBSnapshot
 &CopyTags=true
 &SignatureMethod=HmacSHA256
 &SignatureVersion=4
 &SourceDBSnapshotIdentifier=mysql-instance1-snapshot-20130805
 &TargetDBSnapshotIdentifier=mydbsnapshotcopy
 &Version=2013-09-09
 &X-Amz-Algorithm=AWS4-HMAC-SHA256
 &X-Amz-Credential=AKIADQKE4SARGYLE/20140429/us-west-1/rds/aws4_request
 &X-Amz-Date=20140429T175351Z
 &X-Amz-SignedHeaders=content-type;host;user-agent;x-amz-content-sha256;x-amz-date
 &X-Amz-Signature=9164337efa99caf850e874a1cb7ef62f3cea29d0b448b9e0e7c53b288ddffed2
```

Example Da non crittografata, tra regioni

Il seguente codice crea una copia di uno snapshot, con il nuovo nome `mydbsnapshotcopy`, nella regione Stati Uniti occidentali (California settentrionale).

```
https://rds.us-west-1.amazonaws.com/
 ?Action=CopyDBSnapshot
 &SignatureMethod=HmacSHA256
 &SignatureVersion=4
 &SourceDBSnapshotIdentifier=arn%3Aaws%3Ards%3Aus-east-1%3A123456789012%3Asnapshot
%3Amysql-instance1-snapshot-20130805
 &TargetDBSnapshotIdentifier=mydbsnapshotcopy
 &Version=2013-09-09
 &X-Amz-Algorithm=AWS4-HMAC-SHA256
 &X-Amz-Credential=AKIADQKE4SARGYLE/20140429/us-west-1/rds/aws4_request
 &X-Amz-Date=20140429T175351Z
 &X-Amz-SignedHeaders=content-type;host;user-agent;x-amz-content-sha256;x-amz-date
 &X-Amz-Signature=9164337efa99caf850e874a1cb7ef62f3cea29d0b448b9e0e7c53b288ddffed2
```

Example Da crittografata, tra regioni

Il seguente codice crea una copia di uno snapshot, con il nuovo nome `mydbsnapshotcopy`, nella regione US East (N. Virginia).

```
https://rds.us-east-1.amazonaws.com/
    ?Action=CopyDBSnapshot
    &KmsKeyId=my-us-east-1-key
    &OptionGroupName=custom-option-group-name
```

```
    &PreSignedUrl=https%253A%252F%252Frds.us-west-2.amazonaws.com%252F
         %253FAction%253DCopyDBSnapshot
         %2526DestinationRegion%253Dus-east-1
         %2526KmsKeyId%253Dmy-us-east-1-key
         %2526SourceDBSnapshotIdentifier%253Darn%25253Aaws%25253Ards%25253Aus-
west-2%25253A123456789012%25253Asnapshot%25253Amysql-instance1-snapshot-20161115
         %2526SignatureMethod%253DHmacSHA256
         %2526SignatureVersion%253D4
         %2526Version%253D2014-10-31
         %2526X-Amz-Algorithm%253DAWS4-HMAC-SHA256
         %2526X-Amz-Credential%253DAKIADQKE4SARGYLE%252F20161117%252Fus-west-2%252Frds
%252Faws4_request
         %2526X-Amz-Date%253D20161117T215409Z
         %2526X-Amz-Expires%253D3600
         %2526X-Amz-SignedHeaders%253Dcontent-type%253Bhost%253Buser-agent%253Bx-amz-
content-sha256%253Bx-amz-date
         %2526X-Amz-Signature
%253D255a0f17b4e717d3b67fad163c3ec26573b882c03a65523522cf890a67fca613
    &SignatureMethod=HmacSHA256
    &SignatureVersion=4
    &SourceDBSnapshotIdentifier=arn%3Aaws%3Ards%3Aus-west-2%3A123456789012%3Asnapshot
%3Amysql-instance1-snapshot-20161115
    &TargetDBSnapshotIdentifier=mydbsnapshotcopy
    &Version=2014-10-31
    &X-Amz-Algorithm=AWS4-HMAC-SHA256
    &X-Amz-Credential=AKIADQKE4SARGYLE/20161117/us-east-1/rds/aws4_request
    &X-Amz-Date=20161117T221704Z
    &X-Amz-SignedHeaders=content-type;host;user-agent;x-amz-content-sha256;x-amz-date
    &X-Amz-Signature=da4f2da66739d2e722c85fcfd225dc27bba7e2b8dbea8d8612434378e52adccf
```

Condivisione di uno snapshot del database

Con Amazon RDS puoi condividere uno snapshot di database manuale nei modi seguenti:

- La condivisione di uno snapshot DB manuale, crittografato o non crittografato, consente agli utenti autorizzati Account AWS di copiare lo snapshot.
- La condivisione di un'istantanea database manuale non crittografata consente Account AWS agli utenti autorizzati di ripristinare direttamente un'istanza DB dalla snapshot anziché prenderne una copia e ripristinarla da quella. Tuttavia, non puoi ripristinare un'istanza database da una snapshot DB condivisa e crittografata. Invece puoi copiare la snapshot DB e ripristinare l'istanza database dalla copia.

> **Note**
>
> Per condividere uno snapshot di database automatico, occorre creare uno snapshot di database manuale copiando lo snapshot automatico e poi condividere la copia. Questo processo si applica anche alle risorse AWS generate dal backup.

Per ulteriori informazioni sulla creazione di una copia di una snapshot, consulta Copia di una snapshot DB.. Per ulteriori informazioni sul ripristino di un'istanza database da uno snapshot di database, consulta Ripristino da uno snapshot database.

È possibile condividere un'istantanea manuale con un massimo di 20 altre persone. Account AWS

Le seguenti limitazioni si applicano alla condivisione di istantanee manuali con altri utenti: Account AWS

- Quando ripristini un'istanza DB da uno snapshot condiviso utilizzando AWS Command Line Interface (AWS CLI) o l'API Amazon RDS, devi specificare l'Amazon Resource Name (ARN) dello snapshot condiviso come identificatore dello snapshot.
- Non è possibile condividere uno snapshot di database che utilizza un gruppo di opzioni con opzioni permanenti o persistenti, ad eccezione delle istanze Oracle DB, che hanno l'opzione `Timezone` o `OLS` (o entrambe).

  Non è possibile rimuovere un'opzione permanente da un gruppo di opzioni. Non è possibile rimuovere i gruppi di opzioni con opzioni persistenti da un'istanza database una volta che il gruppo di opzioni è stato assegnato all'istanza database.

La tabella seguente elenca le opzioni permanenti e persistenti e i loro motori di database correlati.

Nome opzione	Persistente	Permanente	Motore database
TDE	Sì	No	Microsoft SQL Server Enterprise Edition
TDE	Sì	Sì	Oracle Enterprise Edition
Fuso orario	Sì	Sì	Oracle Enterprise Edition Oracle Standard Edition Oracle Standard Edition One Oracle Standard Edition 2

Per le istanze database di Oracle, puoi copiare gli snapshot DB condivisi che hanno l'opzione `Timezone` o `OLS` (o entrambe). Per eseguire l'operazione, specifica un gruppo di opzioni target che include queste opzioni quando copi lo snapshot DB. L'opzione OLS è permanente e persistente solo per le istanze Oracle DB in esecuzione su Oracle 12.2 o versioni successive. Per ulteriori informazioni su queste opzioni, consulta Fuso orario Oracle e Oracle Label Security.

- Non è possibile condividere un'istantanea di un cluster DB Multi-AZ.

Indice

- Interruzione della condivisione delle istantanee

Condivisione di uno snapshot

È possibile condividere uno snapshot DB utilizzando AWS Management Console, the o l'API AWS CLI RDS.

Console

Utilizzando la console Amazon RDS, puoi condividere uno snapshot DB manuale con un massimo di 20 persone. Account AWS Puoi anche utilizzare la console per interrompere la condivisione di una snapshot manuale con uno o più account.

Come condividere uno snapshot manuale tramite la console Amazon RDS

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, selezionare Snapshots (Snapshot).
3. Selezionare lo snapshot manuale da condividere.
4. Per Actions (Operazioni), seleziona Share snapshot (Condividi snapshot).
5. Scegliere una delle opzioni seguenti per DB snapshot visibility (Visibilità snapshot DB).

   - Se l'origine non è crittografata, scegli Pubblica per consentire a tutti gli AWS account di ripristinare un'istanza DB dallo snapshot DB manuale, oppure scegli Privato per consentire solo a Account AWS ciò che hai specificato di ripristinare un'istanza DB dallo snapshot DB manuale.

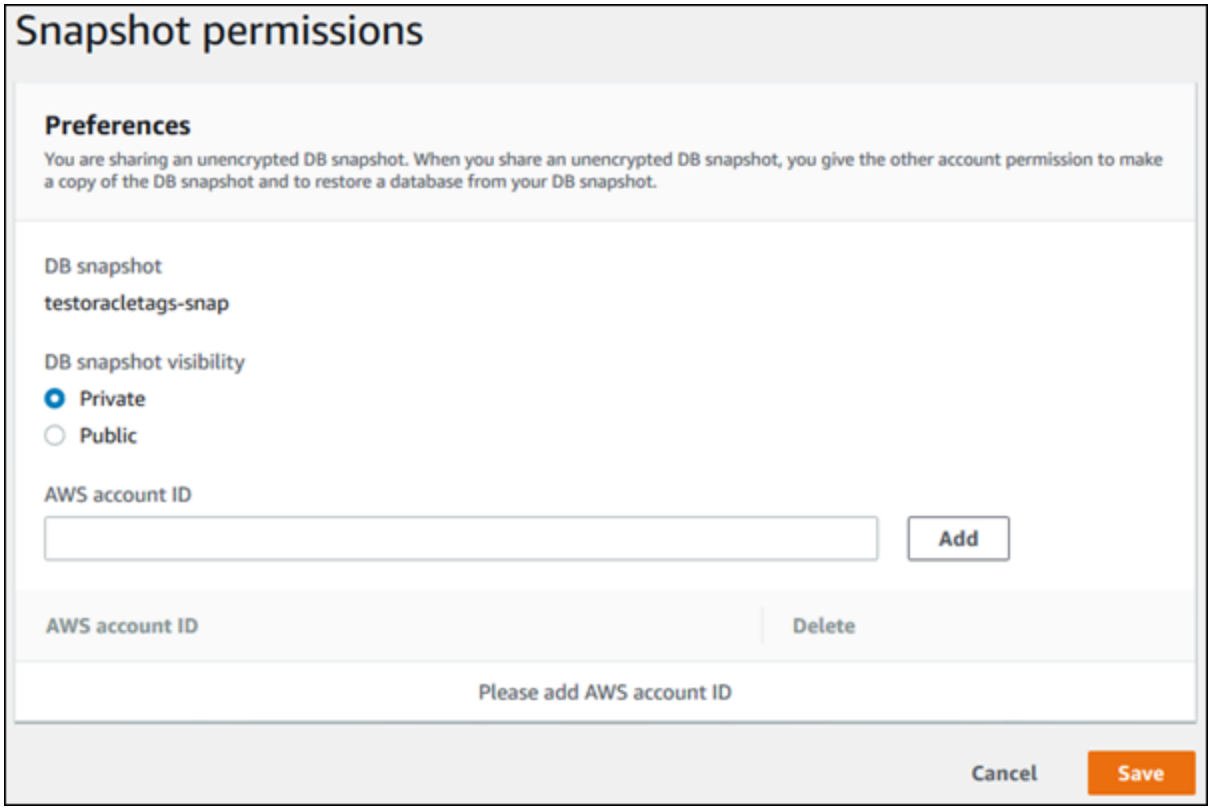
     > **Warning**
     >
     > Se imposti la visibilità dello snapshot DB su Pubblico, tutti Account AWS possono ripristinare un'istanza DB dallo snapshot DB manuale e avere accesso ai tuoi dati. Non condividere le snapshot DB manuali che contengono informazioni private come Public (Pubblica).
     > Per ulteriori informazioni, consulta Condivisione di snapshot pubblici.

   - Se l'origine è crittografata, l'opzione DB snapshot visibility (Visibilità snapshot DB) è impostata su Private (Privato), perché gli snapshot crittografati non possono essere condivisi come pubblici.

> **Note**
>
> Le istantanee che sono state crittografate con l'impostazione predefinita non AWS KMS key possono essere condivise. Per informazioni su come risolvere questo problema, consultaCondivisione di snapshot crittografati.

6. Per AWS Account ID, inserisci l' Account AWS identificatore di un account a cui desideri consentire il ripristino di un'istanza DB dallo snapshot manuale, quindi scegli Aggiungi. Ripeti l'operazione per includere Account AWS identificatori aggiuntivi, fino a 20. Account AWS

   Se commetti un errore durante l'aggiunta di un Account AWS identificatore all'elenco degli account consentiti, puoi eliminarlo dall'elenco scegliendo Elimina a destra dell'identificatore errato Account AWS .



7. Dopo aver aggiunto gli identificatori per tutti quelli a Account AWS cui desideri consentire il ripristino dell'istantanea manuale, scegli Salva per salvare le modifiche.

AWS CLI

Per condividere una snapshot DB, utilizza il comando `aws rds modify-db-snapshot-attribute`. Utilizzate il `--values-to-add` parametro per aggiungere un elenco degli ID autorizzati a ripristinare Account AWS l'istantanea manuale.

Example di condividere uno snapshot con un singolo account

L'esempio seguente abilita l' Account AWS identificatore `123456789012` per ripristinare lo snapshot del DB denominato. `db7-snapshot`

PerLinux, omacOS: Unix

```
aws rds modify-db-snapshot-attribute \
--db-snapshot-identifier db7-snapshot \
--attribute-name restore \
--values-to-add 123456789012
```

Per Windows:

```
aws rds modify-db-snapshot-attribute ^
--db-snapshot-identifier db7-snapshot ^
--attribute-name restore ^
--values-to-add 123456789012
```

Example di condividere uno snapshot con più account

L'esempio seguente abilita due Account AWS identificatori `111122223333` e`444455556666`, per ripristinare lo snapshot del DB denominato. `manual-snapshot1`

PerLinux, omacOS: Unix

```
aws rds modify-db-snapshot-attribute \
--db-snapshot-identifier manual-snapshot1 \
--attribute-name restore \
--values-to-add {"111122223333","444455556666"}
```

Per Windows:

```
aws rds modify-db-snapshot-attribute ^
--db-snapshot-identifier manual-snapshot1 ^
```

```
--attribute-name restore ^
--values-to-add "[\"111122223333\",\"444455556666\"]"
```

> **Note**
>
> Quando usi il prompt comandi di Windows, non devi inserire le doppie virgolette (") nel codice JSON precedendole con il backslash (\).

Per elencare gli Account AWS utenti abilitati al ripristino di un'istantanea, utilizzare il `describe-db-snapshot-attributes` AWS CLI comando.

API RDS

Puoi anche condividere uno snapshot DB manuale con altri utenti Account AWS utilizzando l'API Amazon RDS. Per eseguire questa operazione, chiama l'operazione `ModifyDBSnapshotAttribute`. `AttributeName`Specificate `restore` e utilizzate il `ValuesToAdd` parametro per aggiungere un elenco degli ID autorizzati a ripristinare lo snapshot manuale. Account AWS

Per rendere pubblica e ripristinabile da tutti un'istantanea manuale Account AWS, usa il valore. `all` Tuttavia, fai attenzione a non aggiungere `all` valore alle istantanee manuali che contengono informazioni private che non desideri siano disponibili per tutti. Account AWS Inoltre, non è necessario specificare `all` per le snapshot crittografate, perché l'operazione di rendere pubbliche queste snapshot non è supportata.

Per elencare tutte le istantanee Account AWS consentite per ripristinare un'istantanea, utilizza l'operazione `DescribeDBSnapshotAttributes`API.

Condivisione di snapshot pubblici

È inoltre possibile condividere un'istantanea manuale non crittografata come pubblica, in modo da renderla disponibile a tutti. Account AWS Asicurati, quando condividi uno snapshot come pubblico, che non siano incluse nessuna delle tue informazioni personali.

Quando un'istantanea viene condivisa pubblicamente, concede a tutti i Account AWS permessi sia per copiarla che per creare istanze DB da essa.

Non ti viene addebitata l'archiviazione di backup degli snapshot pubblici di proprietà di altri account. Ti verranno addebitate solo gli snapshot di tua proprietà.

Se si copia uno snapshot pubblico, si è proprietari della copia. Viene addebitato l'archiviazione di backup dello snapshot istantaneo. Se si crea un'istanza database da uno snapshot pubblico, ti viene addebitata tale istanza. Per informazioni sui prezzi di Amazon RDS, consulta la pagina del prodotto Amazon RDS.

È possibile eliminare solo gli snapshot pubblici di tua proprietà. Per eliminare un'istantanea condivisa o pubblica, assicurati di accedere alla persona proprietaria della Account AWS snapshot.

Visualizzazione di istantanee pubbliche di proprietà di altri Account AWS

Puoi visualizzare gli snapshot pubblici di proprietà di altri account in una particolare AWS regione nella scheda Pubblico della pagina Snapshot nella console Amazon RDS. Gli snapshot (quelli di proprietà del tuo account) non vengono visualizzati in questa scheda.

Per visualizzare gli snapshot pubblici

1. Apri la console di Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel pannello di navigazione, selezionare Snapshots (Snapshot).
3. Scegliere la scheda Public (Pubblico).

   Vengono visualizzati gli snapshot pubblici. È possibile vedere quale account possiede uno snapshot pubblico nella colonna Owner (Proprietario).

   > **Note**
   >
   > Potrebbe essere necessario modificare le preferenze della pagina, selezionando l'icona a forma di ingranaggio in alto a destra dell'elenco Public snapshots (Snapshot pubblici), per visualizzare questa colonna.

Visualizzazione degli snapshot pubblici

Puoi usare il seguente AWS CLI comando (solo Unix) per visualizzare gli snapshot pubblici di tua Account AWS proprietà in una particolare regione. AWS

```
aws rds describe-db-snapshots --snapshot-type public --include-public |
 grep account_number
```

Se si dispone di snapshot pubblici, l'output restituito è simile all'esempio seguente.

```
"DBSnapshotArn": "arn:aws:rds:us-east-1:123456789012:snapshot:mysnapshot1",
"DBSnapshotArn": "arn:aws:rds:us-east-1:123456789012:snapshot:mysnapshot2",
```

> **Note**
>
> Potresti vedere voci duplicate per `DBSnapshotIdentifier` o `SourceDBSnapshotIdentifier`.

Condivisione di istantanee pubbliche da versioni obsolete del motore DB

Il ripristino o la copia di istantanee pubbliche da versioni obsolete del motore DB non è supportato.

I motori DB RDS per Oracle e RDS per PostgreSQL supportano l'aggiornamento diretto delle versioni del motore snapshot DB. Puoi aggiornare le tue istantanee e condividerle nuovamente pubblicamente. Per ulteriori informazioni, consulta gli argomenti seguenti:

- Aggiornamento di uno shapshot DB Oracle
- Aggiornamento di una versione del motore di snapshot database PostgreSQL

Per altri motori di database, esegui i seguenti passaggi per rendere lo snapshot pubblico esistente non supportato disponibile per il ripristino o la copia:

1. Contrassegna l'istantanea come privata.
2. Ripristinare lo snapshot:
3. Aggiorna l'istanza DB ripristinata a una versione del motore supportata.
4. Crea una nuova istantanea.
5. Condividete nuovamente l'istantanea pubblicamente.

Condivisione di snapshot crittografati

Puoi condividere gli snapshot di database crittografati con stato inattivo usando l'algoritmo di crittografia AES-256, come descritto in Crittografia delle risorse Amazon RDS.

Per la condivisione di snapshot crittografati vigono le seguenti restrizioni:

- Non puoi condividere le snapshot crittografate come pubbliche.

- Non puoi condividere le snapshot Oracle o Microsoft SQL Server che sono crittografate utilizzando Transparent Data Encryption (TDE).
- Non è possibile condividere un'istantanea che è stata crittografata utilizzando la chiave KMS predefinita di chi ha condiviso Account AWS l'istantanea.

Per risolvere il problema della chiave KMS predefinita, esegui le seguenti attività:

1. Crea una chiave gestita dal cliente e consenti l'accesso ad essa.
2. Copia e condividi l'istantanea dall'account di origine.
3. Copia l'istantanea condivisa nell'account di destinazione.

Crea una chiave gestita dal cliente e consenti l'accesso ad essa

Per prima cosa crei una chiave KMS personalizzata nella Regione AWS stessa immagine crittografata del DB. Durante la creazione della chiave gestita dal cliente, concedi l'accesso ad essa per un'altra chiave. Account AWS

Per creare una chiave gestita dal cliente e darvi accesso

1. Accedi a AWS Management Console dalla fonte Account AWS.
2. Apri la AWS KMS console all'indirizzo https://console.aws.amazon.com/kms.
3. Per modificare la Regione AWS, usa il selettore della regione nell'angolo superiore destro della pagina.
4. Nel riquadro di navigazione, scegli Chiavi gestite dal cliente.
5. Scegliere Create key (Crea chiave).
6. Nella pagina Configura chiave:
    a. Per Tipo di chiave, seleziona Symmetric.
    b. Per Utilizzo della chiave, seleziona Crittografa e decrittografa.
    c. Espandere Advanced options (Opzioni avanzate).
    d. Per l'origine del materiale chiave, seleziona KMS.
    e. Per Regionalità, seleziona la chiave per regione singola.
    f. Seleziona Successivo.
7. Nella pagina Aggiungi etichette:

a. Per Alias, inserisci un nome visualizzato per la tua chiave KMS, ad esempio. **`share-snapshot`**

b. (Facoltativo) Inserisci una descrizione per la tua chiave KMS.

c. (Facoltativo) Aggiungi tag alla tua chiave KMS.

d. Seleziona Successivo.

8. Nella pagina Definisci le autorizzazioni per gestire la chiave scegli Avanti.
9. Nella pagina Definisci le autorizzazioni di utilizzo delle chiavi:

a. Per Altro Account AWS, scegli Aggiungi un altro Account AWS.

b. Inserisci l'ID del file Account AWS a cui desideri concedere l'accesso.

Puoi dare accesso a più di uno Account AWS.

c. Seleziona Successivo.

10. Controlla la tua chiave KMS, quindi scegli Fine.

Copia e condividi l'istantanea dall'account di origine

Successivamente si copia lo snapshot del DB di origine in una nuova istantanea utilizzando la chiave gestita dal cliente. Quindi lo condividi con il destinatario. Account AWS

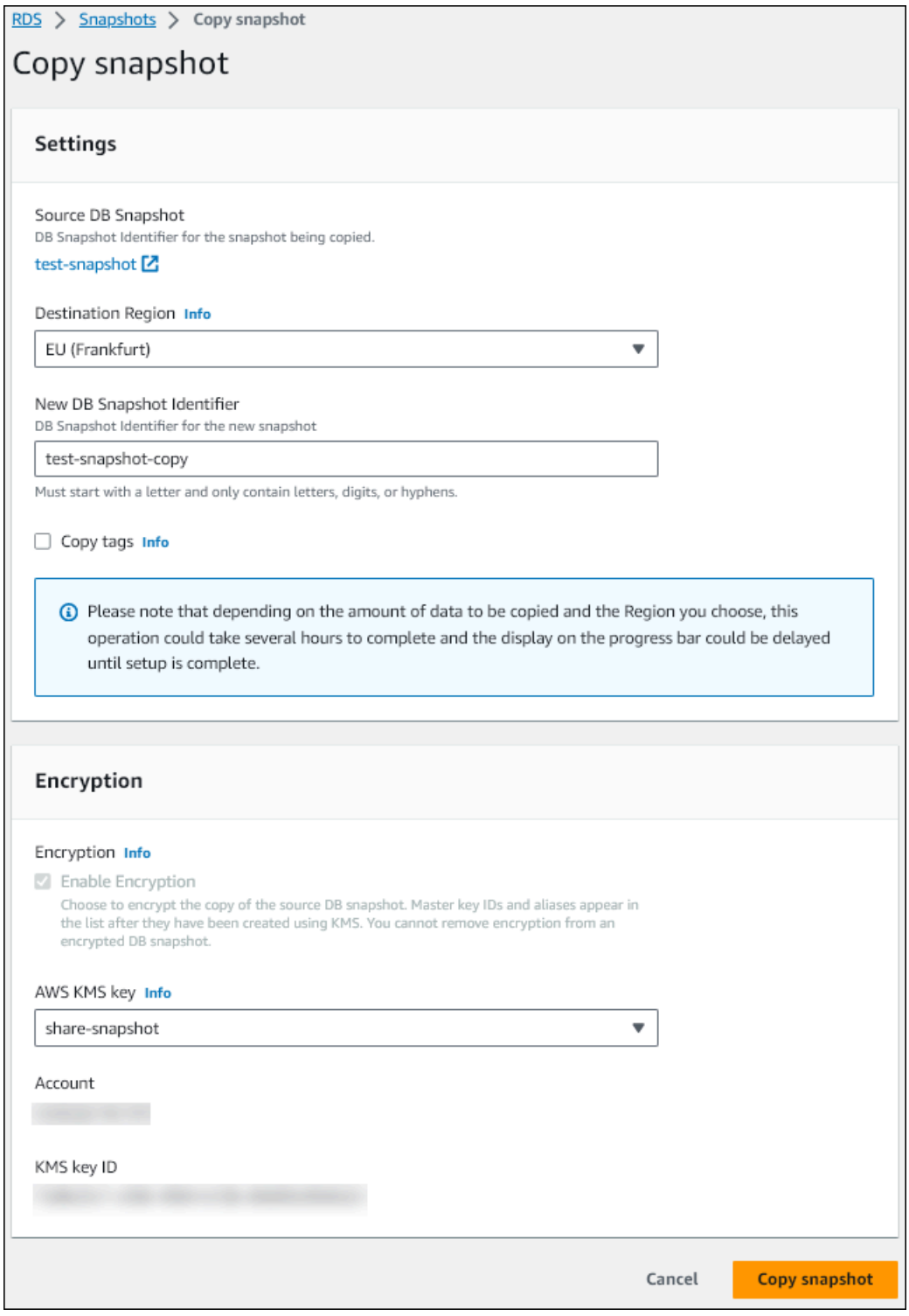
Per copiare e condividere l'istantanea

1. Accedi a AWS Management Console dalla fonte Account AWS.
2. Apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/
3. Nel pannello di navigazione, selezionare Snapshots (Snapshot).
4. Seleziona lo snapshot DB che desideri copiare.
5. In Operazioni, seleziona Copia snapshot.
6. Nella pagina Copia istantanea:

a. Per Regione di destinazione, scegli la posizione Regione AWS in cui hai creato la chiave gestita dal cliente nella procedura precedente.

b. Immettete il nome della copia dello snapshot DB in New DB Snapshot Identifier.

c. Per AWS KMS key, scegli la chiave gestita dal cliente che hai creato.



d. Selezionare Copy Snapshot (Copia snapshot).

7. Quando la copia dell'istantanea è disponibile, selezionala.
8. Per Actions (Operazioni), seleziona Share snapshot (Condividi snapshot).
9. Nella pagina delle autorizzazioni dello snapshot:

a. Inserisci l'Account AWS ID con cui vuoi condividere la copia dell'istantanea, quindi scegli Aggiungi.

b. Selezionare Salva.

L'istantanea è condivisa.

Copia l'istantanea condivisa nell'account di destinazione

Ora puoi copiare l'istantanea condivisa nella destinazione. Account AWS

Per copiare l'istantanea condivisa

1. Accedi a AWS Management Console dalla destinazione Account AWS.
2. Apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/
3. Nel pannello di navigazione, selezionare Snapshots (Snapshot).
4. Scegli la scheda Condivisi con me.
5. Seleziona l'istantanea condivisa.
6. In Operazioni, seleziona Copia snapshot.
7. Scegliete le impostazioni per copiare l'istantanea come nella procedura precedente, ma utilizzate una AWS KMS key che appartenga all'account di destinazione.

   Selezionare Copy Snapshot (Copia snapshot).

Interruzione della condivisione delle istantanee

Per interrompere la condivisione di uno snapshot DB, rimuovi l'autorizzazione dalla destinazione. Account AWS

Console

Per interrompere la condivisione di uno snapshot DB manuale con un Account AWS

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, selezionare Snapshots (Snapshot).

3. Selezionare lo snapshot manuale di cui interrompere la condivisione.
4. Scegli Actions (Operazioni) e quindi Share Snapshot (Condividi snapshot).
5. Per rimuovere l'autorizzazione per un AWS account Account AWS, scegli Elimina come identificatore dell'account dall'elenco degli account autorizzati.
6. Scegliere Salva per salvare le modifiche.

CLI

Per rimuovere un Account AWS identificatore dall'elenco, utilizza il `--values-to-remove` parametro.

Example di interrompere la condivisione degli snapshot

L'esempio seguente impedisce all' Account AWS ID 444455556666 di ripristinare l'istantanea.

PerLinux, o: macOS Unix

```
aws rds modify-db-snapshot-attribute \
--db-snapshot-identifier manual-snapshot1 \
--attribute-name restore \
--values-to-remove 444455556666
```

Per Windows:

```
aws rds modify-db-snapshot-attribute ^
--db-snapshot-identifier manual-snapshot1 ^
--attribute-name restore ^
--values-to-remove 444455556666
```

API RDS

Per rimuovere l'autorizzazione di condivisione per un Account AWS, usa l'`ModifyDBSnapshotAttribute`operazione con `AttributeName` set to `restore` e il `ValuesToRemove` parametro. Per contrassegnare una snapshot manuale come privata, rimuovi il valore `all` dall'elenco dei valori per l'attributo `restore`.

Esportazione dei dati dello snapshot DB in Simple Storage Service (Amazon S3)

È possibile esportare i dati dello snapshot DB in un bucket Simple Storage Service (Amazon S3). Il processo di esportazione viene eseguito in background e non influisce sulle prestazioni dell'istanza del cluster di database attivo.

Quando si esporta uno snapshot di database, Amazon RDS estrae i dati dallo snapshot e li archivia in un bucket Amazon S3. I dati vengono archiviati in un formato Apache Parquet compresso e coerente.

È possibile esportare tutti i tipi di istantanee DB, incluse istantanee manuali, istantanee di sistema automatizzate e istantanee create dal servizio. AWS Backup Per impostazione predefinita, vengono esportati tutti i dati nello snapshot. Tuttavia, è possibile scegliere di esportare set specifici di database, schemi o tabelle.

Dopo l'esportazione dei dati, è possibile analizzare i dati esportati direttamente mediante strumenti quali Amazon Athena o Amazon Redshift Spectrum. Per ulteriori informazioni sull'utilizzo di Athena per leggere i dati di Parquet, consulta Parquet SerDe nella Guida per l'utente di Amazon Athena. Per ulteriori informazioni sull'utilizzo Redshift Spectrum per leggere i dati Parquet, consulta COPY da formati di dati a colonna nella Guida per gli sviluppatori di database di Amazon Redshift.

Argomenti

Disponibilità di regioni e versioni

Il supporto varia a seconda delle versioni specifiche di ciascun motore di database e a seconda delle Regioni AWS. Per ulteriori informazioni sulla disponibilità di versioni e regioni in caso di esportazione di snapshot in S3, consulta Regioni e motori DB supportati per l'esportazione di snapshot in S3 in Amazon RDS.

Limitazioni

L'esportazione dei dati snapshot DB in Simple Storage Service (Amazon S3) presenta le seguenti limitazioni:

- Non è possibile eseguire contemporaneamente più attività di esportazione per lo stesso snapshot database. Ciò è valido sia per le esportazioni totali sia per le esportazioni parziali.
- L'esportazione di snapshot da istanze database che utilizzano lo storage magnetico non è supportata.
- Le esportazioni verso S3 non supportano i prefissi S3 contenenti i due punti (:).
- I seguenti caratteri nel percorso del file S3 vengono convertiti in caratteri di sottolineatura (_) durante l'esportazione:

  ```
  \ ` " (space)
  ```

- Se un database, uno schema o una tabella contiene caratteri diversi da quelli riportati di seguito, l'esportazione parziale non è supportata. Tuttavia, è possibile esportare l'intero snapshot DB.
  - Lettere latine (A–Z)
  - Numeri (0–9)
  - Simbolo del dollaro ($)
  - Carattere di sottolineatura (_)
- Gli spazi ( ) e alcuni caratteri non sono supportati nei nomi delle colonne delle tabelle del database. Le tabelle con i seguenti caratteri nei nomi delle colonne vengono ignorate durante l'esportazione:

  ```
  , ; { } ( ) \n \t = (space)
  ```

- Le tabelle con barre (/) nei rispettivi nomi vengono ignorate durante l'esportazione.
- Le tabelle temporanee e non registrate di RDS per PostgreSQL vengono ignorate durante l'esportazione.

- Se i dati contengono un oggetto di grandi dimensioni, ad esempio un BLOB o un CLOB, vicino o superiore a 500 MB, l'esportazione non riesce.
- Se una tabella contiene una riga di grandi dimensioni, vicine o superiori a 2 GB, la tabella viene ignorata durante l'esportazione.
- Per le esportazioni parziali, l'`ExportOnly`elenco ha una dimensione massima di 200 KB.
- Si consiglia vivamente di utilizzare un nome univoco per ogni attività di esportazione. Se non utilizzi un nome di attività univoco, potresti ricevere il seguente messaggio di errore:

  ExportTaskAlreadyExistsFault: Si è verificato un errore (ExportTaskAlreadyExists) durante la chiamata dell' StartExportTaskoperazione: l'operazione di esportazione con l'ID *xxxxx* esiste già.
- È possibile eliminare uno snapshot durante l'esportazione dei suoi dati in S3, ma vengono comunque addebitati i costi di storage per tale snapshot fino al completamento dell'attività di esportazione.
- Non è possibile ripristinare i dati di snapshot esportati da S3 in una nuova istanza database.

Panoramica sull'esportazione dei dati degli snapshot

Per esportare i dati dello snapshot DB in un bucket Simple Storage Service (Amazon S3) puoi utilizzare il processo riportato di seguito. Per ulteriori dettagli, consulta le seguenti sezioni:

1. Identificare lo snapshot da esportare.

   Utilizzare uno snapshot automatico o manuale esistente oppure creare uno snapshot manuale di un'istanza database.
2. Configurare l'accesso al bucket Simple Storage Service (Amazon S3).

   Un bucket è un container per oggetti o file Simple Storage Service (Amazon S3). Per fornire le informazioni per accedere a un bucket, attenersi alla seguente procedura:

   a. Identificare il bucket S3 in cui deve essere esportato lo snapshot. Il bucket S3 deve trovarsi nella stessa AWS regione dell'istantanea. Per ulteriori informazioni, consulta Identificazione del bucket Simple Storage Service (Amazon S3) in cui esportare.

   b. Crea un ruolo AWS Identity and Access Management (IAM) che conceda all'attività di esportazione degli snapshot l'accesso al bucket S3. Per ulteriori informazioni, consulta Fornire l'accesso a un bucket Simple Storage Service (Amazon S3) utilizzando un ruolo IAM.

3. Crea una crittografia simmetrica per la crittografia lato server. AWS KMS key La chiave KMS viene utilizzata dall'attività di esportazione delle istantanee per configurare la crittografia AWS KMS lato server durante la scrittura dei dati di esportazione su S3.

   La policy della chiave KMS deve includere entrambe le autorizzazioni `kms:CreateGrant` e `kms:DescribeKey`. Per ulteriori informazioni sull'uso delle chiavi KMS in Amazon RDS, consulta Gestione di AWS KMS key.

   Se hai una dichiarazione di rifiuto nella tua politica sulle chiavi KMS, assicurati di escludere esplicitamente il responsabile del servizio. AWS `export.rds.amazonaws.com`

   Puoi utilizzare una chiave KMS all'interno del tuo AWS account oppure puoi utilizzare una chiave KMS per più account. Per ulteriori informazioni, consulta Utilizzo di un account incrociato AWS KMS key per crittografare le esportazioni Amazon S3.
4. Esportare lo snapshot in Simple Storage Service (Amazon S3) utilizzando la console o il comando CLI `start-export-task`. Per ulteriori informazioni, consulta Esportazione di uno snapshot DB in un bucket Amazon S3.
5. Per accedere ai dati esportati nel bucket Simple Storage Service (Amazon S3), consulta Caricamento, download e gestione di oggetti nella Guida per l'utente di Amazon Simple Storage Service.

Configurazione dell'accesso a un bucket Simple Storage Service (Amazon S3)

Per esportare i dati dello snapshot DB in un file Simple Storage Service (Amazon S3), è innanzitutto necessario concedere allo snapshot l'autorizzazione per accedere al bucket Simple Storage Service (Amazon S3). È quindi possibile creare un ruolo IAM per consentire al servizio Amazon RDS di scrivere nel bucket Amazon S3.

Argomenti

- Identificazione del bucket Simple Storage Service (Amazon S3) in cui esportare
- Fornire l'accesso a un bucket Simple Storage Service (Amazon S3) utilizzando un ruolo IAM
- Utilizzo di un bucket Simple Storage Service (Amazon S3) multiaccount
- Utilizzo di un account incrociato AWS KMS key per crittografare le esportazioni Amazon S3

Identificazione del bucket Simple Storage Service (Amazon S3) in cui esportare

Identificare il bucket Simple Storage Service (Amazon S3) in cui esportare lo snapshot DB. Utilizzare un bucket S3 esistente o crearne uno nuovo.

> **Note**
>
> Il bucket S3 in cui esportare deve trovarsi nella stessa AWS regione dell'istantanea.

Per ulteriori informazioni sull'utilizzo dei bucket Simple Storage Service (Amazon S3), vedere quanto segue in Guida per l'utente di Amazon Simple Storage Service:

- Come visualizzare le proprietà di un bucket S3?
- In che modo si abilita la crittografia di default per un bucket Amazon S3?
- Come creare un bucket S3?

Fornire l'accesso a un bucket Simple Storage Service (Amazon S3) utilizzando un ruolo IAM

Prima di esportare i dati dello snapshot DB in Simple Storage Service (Amazon S3), fornire l'autorizzazione di accesso in scrittura alle attività di esportazione dello snapshot al bucket Simple Storage Service (Amazon S3).

Per concedere l'autorizzazione, crea una policy IAM che fornisca accesso al bucket, crea un ruolo IAM e collega la policy al ruolo. Successivamente assegnare il ruolo IAM all'attività di esportazione dello snapshot.

> **Important**
>
> Se prevedi di utilizzare il AWS Management Console per esportare la tua istantanea, puoi scegliere di creare automaticamente la policy IAM e il ruolo quando esporti la snapshot. Per istruzioni, consulta Esportazione di uno snapshot DB in un bucket Amazon S3.

Per fornire alle attività dello snapshot DB l'accesso a Amazon S3

1. Creare una policy IAM Questa policy fornisce le autorizzazioni al bucket e all'oggetto che consentono all'attività di esportazione snapshot l'accesso a Amazon S3.

Includi nella policy le seguenti operazioni necessarie per consentire il trasferimento dei file da Amazon RDS a un bucket S3:

- `s3:PutObject*`
- `s3:GetObject*`
- `s3:ListBucket`
- `s3:DeleteObject*`
- `s3:GetBucketLocation`

Includi nella policy le seguenti risorse per identificare il bucket S3 e gli oggetti nel bucket. Il seguente elenco di risorse mostra il formato Amazon Resource Name (ARN) per l'accesso a Amazon S3.

- `arn:aws:s3:::your-s3-bucket`
- `arn:aws:s3:::your-s3-bucket/*`

Per ulteriori informazioni sulla creazione di una policy IAM per Amazon RDS, consultare Creazione e utilizzo di una policy IAM per l'accesso al database IAM. Consulta anche il Tutorial: Creare e collegare la prima policy gestita dal cliente nella Guida per l'utente di IAM.

Il AWS CLI comando seguente crea una policy IAM denominata `ExportPolicy` con queste opzioni. Concede l'accesso a un bucket denominato `your-s3-bucket`.

> **Note**
>
> Dopo aver creato la policy, prendere nota del relativo ARN. Per la fase successiva, in cui si associa la policy a un ruolo IAM, è necessario l'ARN.

```
aws iam create-policy  --policy-name ExportPolicy --policy-document '{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ExportPolicy",
            "Effect": "Allow",
            "Action": [
```

```
                "s3:PutObject*",
                "s3:ListBucket",
                "s3:GetObject*",
                "s3:DeleteObject*",
                "s3:GetBucketLocation"
            ],
            "Resource": [
                "arn:aws:s3:::your-s3-bucket",
                "arn:aws:s3:::your-s3-bucket/*"
            ]
        }
    ]
}'
```

2. Crea un ruolo IAM in modo che Amazon RDS possa assumere questo ruolo IAM per tuo conto per accedere ai bucket Amazon S3. Per ulteriori informazioni, consulta la pagina relativa alla creazione di un ruolo per delegare le autorizzazioni a un utente IAM nella Guida per l'utente IAM.

   L'esempio seguente mostra l'utilizzo del AWS CLI comando per creare un ruolo denominato`rds-s3-export-role`.

```
aws iam create-role  --role-name rds-s3-export-role  --assume-role-policy-document
 '{
    "Version": "2012-10-17",
    "Statement": [
      {
        "Effect": "Allow",
        "Principal": {
           "Service": "export.rds.amazonaws.com"
         },
        "Action": "sts:AssumeRole"
      }
    ]
  }'
```

3. Collegare la policy IAM al ruolo IAM creato.

   Il AWS CLI comando seguente collega la politica creata in precedenza al ruolo denominato`rds-s3-export-role`. Sostituire *`your-policy-arn`* con l'ARN della policy annotato nella fase precedente.

```
aws iam attach-role-policy  --policy-arn your-policy-arn  --role-name rds-s3-
export-role
```

Utilizzo di un bucket Simple Storage Service (Amazon S3) multiaccount

Puoi utilizzare i bucket Amazon S3 su più account. AWS Per utilizzare un bucket tra account, aggiungi un criterio bucket per consentire l'accesso al ruolo IAM utilizzato per le esportazioni S3. Per informazioni, consulta Esempio 2: il proprietario del bucket concede autorizzazioni per il bucket multiaccount.

- Allega una policy di bucket al bucket, come mostrato nell'esempio riportato di seguito.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::123456789012:role/Admin"
            },
            "Action": [
                "s3:PutObject*",
                "s3:ListBucket",
                "s3:GetObject*",
                "s3:DeleteObject*",
                "s3:GetBucketLocation"
            ],
            "Resource": [
                "arn:aws:s3:::mycrossaccountbucket",
                "arn:aws:s3:::mycrossaccountbucket/*"
            ]
        }
    ]
}
```

Utilizzo di un account incrociato AWS KMS key per crittografare le esportazioni Amazon S3

Puoi utilizzare un account multiplo AWS KMS key per crittografare le esportazioni Amazon S3. Innanzitutto, aggiungi una policy chiave all'account locale, quindi aggiungi le policy IAM nell'account esterno. Per ulteriori informazioni, consulta Autorizzazione per gli utenti in altri account a utilizzare una chiave KMS.

Per utilizzare una chiave KMS multiaccount

1. Aggiungi una policy di chiave all'account locale.

   Il seguente esempio fornisce `ExampleRole` e `ExampleUser` nell'account esterno 444455556666 autorizzazioni nell'account locale 123456789012.

   ```
   {
       "Sid": "Allow an external account to use this KMS key",
       "Effect": "Allow",
       "Principal": {
           "AWS": [
               "arn:aws:iam::444455556666:role/ExampleRole",
               "arn:aws:iam::444455556666:user/ExampleUser"
           ]
       },
       "Action": [
           "kms:Encrypt",
           "kms:Decrypt",
           "kms:ReEncrypt*",
           "kms:GenerateDataKey*",
           "kms:CreateGrant",
           "kms:DescribeKey",
           "kms:RetireGrant"
       ],
       "Resource": "*"
   }
   ```

2. Aggiungere le policy IAM nell'account esterno

   La policy IAM dell'esempio seguente consente al principale di utilizzare la chiave KMS nell'account 123456789012 per le operazioni di crittografia. Per concedere questa autorizzazione a `ExampleRole` e `ExampleUser` nell'account 444455556666, collega la policy ad essi nell'account.

```
{
    "Sid": "Allow use of KMS key in account 123456789012",
    "Effect": "Allow",
    "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:CreateGrant",
        "kms:DescribeKey",
        "kms:RetireGrant"
    ],
    "Resource": "arn:aws:kms:us-
west-2:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab"
}
```

Esportazione di uno snapshot DB in un bucket Amazon S3

Puoi avere in corso fino a cinque attività simultanee di esportazione di snapshot DB per volta. Account AWS

> **Note**
>
> L'esportazione di snapshot RDS può richiedere qualche minuto a seconda del tipo e delle dimensioni del database. L'attività di esportazione ripristina e ridimensiona innanzitutto l'intero database prima di estrarre i dati su Simple Storage Service (Amazon S3). Lo stato di avanzamento dell'attività durante questa fase viene visualizzato come Avvio. Quando l'attività passa all'esportazione dei dati in S3, lo stato di avanzamento diventa In progress (In corso). Il tempo necessario per completare l'esportazione dipende dai dati memorizzati nel database. Ad esempio, le tabelle con chiave primaria numerica o colonne indice ben distribuite esporteranno più velocemente. Le tabelle che non contengono una colonna adatta al partizionamento e le tabelle con un solo indice su una colonna basata su stringhe richiedono più tempo. Questo tempo di esportazione più lungo si verifica perché l'esportazione utilizza un processo a thread singolo più lento.

Puoi esportare uno snapshot DB su Amazon S3 utilizzando AWS Management Console l'API, AWS CLI the o RDS.

Se si utilizza una funzione Lambda per esportare uno snapshot, aggiungere l'operazione `kms:DescribeKey` alla policy della funzione Lambda. Per ulteriori informazioni, consulta Autorizzazioni di AWS Lambda.

Console

L'opzione Export to Amazon S3 (Esporta in Simple Storage Service (Amazon S3)) viene visualizzata solo per gli snapshot che possono essere esportati in Simple Storage Service (Amazon S3). Uno snapshot potrebbe non essere disponibile per l'esportazione a causa dei seguenti motivi:

- Il motore del database non è supportato per l'esportazione S3.
- La versione dell'istanza database non è supportata per l'esportazione S3.
- L'esportazione da S3 non è supportata nella AWS regione in cui è stata creata la snapshot.

Per esportare uno snapshot DB

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel pannello di navigazione, selezionare Snapshots (Snapshot).
3. Dalle schede, scegliere il tipo di snapshot che si desidera esportare.
4. Nell'elenco degli snapshot, scegliere lo snapshot che si desidera esportare.
5. Per Actions (Operazioni), scegli Export to Amazon S3 (Esporta in Simple Storage Service (Amazon S3).

   Viene visualizzata la finestra Export to Amazon S3 (Esporta in Simple Storage Service (Amazon S3)).
6. Per Export identifier (Identificatore di esportazione), immettere un nome per identificare l'attività di esportazione. Questo valore viene utilizzato anche per il nome del file creato nel bucket S3.
7. Scegli i dati da esportare:

   - Scegliere All (Tutti) per esportare tutti i dati nello snapshot.
   - Scegliere Partial (Parziali) per esportare parti specifiche dello snapshot. Per identificare le parti dello snapshot da esportare, immettere uno o più database, schemi o tabelle per Identifiers (Identificatori), separati da spazi.

     Utilizza il seguente formato:

```
database[.schema][.table] database2[.schema2][.table2] ... databasen[.schemaN]
[.tablen]
```

Ad esempio:

```
mydatabase mydatabase2.myschema1 mydatabase2.myschema2.mytable1
 mydatabase2.myschema2.mytable2
```

8. Per S3 bucket (Bucket S3), scegliere il bucket in cui esportare.

   Per assegnare i dati esportati a un percorso di cartella nel bucket S3, immettere il percorso opzionale per S3 prefix (Prefisso S3).

9. Per il ruolo IAM, scegliere un ruolo che conceda l'accesso in scrittura al bucket S3 scelto o creare un nuovo ruolo.

   - Se è stato creato un ruolo seguendo le fasi in Fornire l'accesso a un bucket Simple Storage Service (Amazon S3) utilizzando un ruolo IAM, scegliere tale ruolo.
   - Se non è stato creato un ruolo che fornisce l'accesso in scrittura al bucket S3 scelto, scegli Create a new role (Crea un nuovo ruolo) per creare automaticamente il ruolo. Immettere quindi un nome per il ruolo nel nome del ruolo IAM.

10. Per AWS KMS key, immettere l'ARN per la chiave da utilizzare per crittografare i dati esportati.
11. Scegliere Export to Amazon S3 (Esporta in Simple Storage Service (Amazon S3)).

AWS CLI

Per esportare uno snapshot DB in Amazon S3 utilizzando, usa AWS CLI il comando con start-export-taskle seguenti opzioni richieste:

- `--export-task-identifier`
- `--source-arn`
- `--s3-bucket-name`
- `--iam-role-arn`
- `--kms-key-id`

Negli esempi seguenti, viene denominata l'attività di esportazione delle istantanee *my-snapshot-export*, che esporta un'istantanea in un bucket S3 denominato. *my-export-bucket*

Example

PerLinux, o: macOS Unix

```
aws rds start-export-task \
    --export-task-identifier my-snapshot-export \
    --source-arn arn:aws:rds:AWS_Region:123456789012:snapshot:snapshot-name \
    --s3-bucket-name my-export-bucket \
    --iam-role-arn iam-role \
    --kms-key-id my-key
```

Per Windows:

```
aws rds start-export-task ^
    --export-task-identifier my-snapshot-export ^
    --source-arn arn:aws:rds:AWS_Region:123456789012:snapshot:snapshot-name ^
    --s3-bucket-name my-export-bucket ^
    --iam-role-arn iam-role ^
    --kms-key-id my-key
```

Di seguito è riportato un output di esempio.

```
{
    "Status": "STARTING",
    "IamRoleArn": "iam-role",
    "ExportTime": "2019-08-12T01:23:53.109Z",
    "S3Bucket": "my-export-bucket",
    "PercentProgress": 0,
    "KmsKeyId": "my-key",
    "ExportTaskIdentifier": "my-snapshot-export",
    "TotalExtractedDataInGB": 0,
    "TaskStartTime": "2019-11-13T19:46:00.173Z",
    "SourceArn": "arn:aws:rds:AWS_Region:123456789012:snapshot:snapshot-name"
}
```

Per fornire un percorso di cartella nel bucket S3 per l'esportazione delle istantanee, includi l'`--s3-prefix`opzione nel comando. start-export-task

API RDS

Per esportare uno snapshot DB su Amazon S3 utilizzando l'API Amazon RDS, utilizza l'operazione con i StartExportTaskseguenti parametri obbligatori:

- `ExportTaskIdentifier`
- `SourceArn`
- `S3BucketName`
- `IamRoleArn`
- `KmsKeyId`

Monitoraggio delle esportazioni di snapshot

Puoi monitorare le esportazioni di snapshot DB utilizzando l'API AWS Management Console AWS CLI, the o RDS.

Console

Per monitorare le esportazioni di snapshot DB

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel pannello di navigazione, selezionare Snapshots (Snapshot).
3. Per monitorare l'elenco delle esportazioni di snapshot, scegliere la scheda Esportazioni in Simple Storage Service (Amazon S3).
4. Per visualizzare informazioni su un'esportazione di snapshot specifica, scegliere l'attività di esportazione.

AWS CLI

Per monitorare le esportazioni di snapshot DB utilizzando il AWS CLI, usa il describe-export-taskscomando.

Nell'esempio seguente viene illustrato come visualizzare le informazioni correnti su tutte le esportazioni di snapshot.

Example

```
aws rds describe-export-tasks

{
    "ExportTasks": [
        {
            "Status": "CANCELED",
            "TaskEndTime": "2019-11-01T17:36:46.961Z",
            "S3Prefix": "something",
            "ExportTime": "2019-10-24T20:23:48.364Z",
            "S3Bucket": "examplebucket",
            "PercentProgress": 0,
            "KmsKeyId": "arn:aws:kms:AWS_Region:123456789012:key/K7MDENG/
bPxRfiCYEXAMPLEKEY",
            "ExportTaskIdentifier": "anewtest",
            "IamRoleArn": "arn:aws:iam::123456789012:role/export-to-s3",
            "TotalExtractedDataInGB": 0,
            "TaskStartTime": "2019-10-25T19:10:58.885Z",
            "SourceArn": "arn:aws:rds:AWS_Region:123456789012:snapshot:parameter-
groups-test"
        },
{
            "Status": "COMPLETE",
            "TaskEndTime": "2019-10-31T21:37:28.312Z",
            "WarningMessage": "{\"skippedTables\":[],\"skippedObjectives\":[],\"general
\":[{\"reason\":\"FAILED_TO_EXTRACT_TABLES_LIST_FOR_DATABASE\"}]}",
            "S3Prefix": "",
            "ExportTime": "2019-10-31T06:44:53.452Z",
            "S3Bucket": "examplebucket1",
            "PercentProgress": 100,
            "KmsKeyId": "arn:aws:kms:AWS_Region:123456789012:key/2Zp9Utk/
h3yCo8nvbEXAMPLEKEY",
            "ExportTaskIdentifier": "thursday-events-test",
            "IamRoleArn": "arn:aws:iam::123456789012:role/export-to-s3",
            "TotalExtractedDataInGB": 263,
            "TaskStartTime": "2019-10-31T20:58:06.998Z",
            "SourceArn":
 "arn:aws:rds:AWS_Region:123456789012:snapshot:rds:example-1-2019-10-31-06-44"
        },
        {
            "Status": "FAILED",
            "TaskEndTime": "2019-10-31T02:12:36.409Z",
```

```
            "FailureCause": "The S3 bucket edgcuc-export isn't located in the current
 AWS Region. Please, review your S3 bucket name and retry the export.",
            "S3Prefix": "",
            "ExportTime": "2019-10-30T06:45:04.526Z",
            "S3Bucket": "examplebucket2",
            "PercentProgress": 0,
            "KmsKeyId": "arn:aws:kms:AWS_Region:123456789012:key/2Zp9Utk/
h3yCo8nvbEXAMPLEKEY",
            "ExportTaskIdentifier": "wednesday-afternoon-test",
            "IamRoleArn": "arn:aws:iam::123456789012:role/export-to-s3",
            "TotalExtractedDataInGB": 0,
            "TaskStartTime": "2019-10-30T22:43:40.034Z",
            "SourceArn":
 "arn:aws:rds:AWS_Region:123456789012:snapshot:rds:example-1-2019-10-30-06-45"
        }
    ]
}
```

Per visualizzare informazioni su un'esportazione di snapshot specifica, includere l'opzione `--export-task-identifier` nel comando `describe-export-tasks`. Per filtrare l'output, includere l'opzione `--Filters`. Per ulteriori opzioni, vedete il describe-export-taskscomando.

API RDS

Per visualizzare informazioni sulle esportazioni di snapshot DB utilizzando l'API Amazon RDS, utilizza l'DescribeExportTasksoperazione.

Per tenere traccia del completamento del flusso di lavoro di esportazione o per attivare un altro flusso di lavoro, è possibile sottoscrivere gli argomenti del Servizio di notifica semplice Amazon. Per ulteriori informazioni su Amazon SNS, consulta Utilizzo della notifica degli eventi di Amazon RDS.

Annullamento di un'attività di esportazione di snapshot

Puoi annullare un'attività di esportazione di snapshot DB utilizzando l'API AWS Management Console AWS CLI, the o RDS.

> **Note**
>
> L'annullamento di un'attività di esportazione di snapshot non rimuove i dati esportati in Simple Storage Service (Amazon S3). Per informazioni su come eliminare i dati utilizzando la

console, consultare Come eliminare oggetti da un bucket S3? Per eliminare i dati utilizzando l'interfaccia della riga di comando (CLI), utilizzare il comando delete-object.

Console

Per annullare un'attività di esportazione di uno snapshot

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel pannello di navigazione, selezionare Snapshots (Snapshot).
3. Scegli la scheda Exports in Simple Storage Service (Amazon S3) (Esportazioni in Simple Storage Service (Amazon S3)).
4. Scegliere l'attività di esportazione di snapshot che si desidera annullare.
5. Seleziona Cancel (Annulla).
6. Scegli Cancel export task (Annulla attività di esportazione) nella pagina di conferma.

AWS CLI

Per annullare un'operazione di esportazione di istantanee utilizzando il AWS CLI, usa il cancel-export-taskcomando. Il comando richiede l'opzione `--export-task-identifier`.

Example

```
aws rds cancel-export-task --export-task-identifier my_export
{
    "Status": "CANCELING",
    "S3Prefix": "",
    "ExportTime": "2019-08-12T01:23:53.109Z",
    "S3Bucket": "examplebucket",
    "PercentProgress": 0,
    "KmsKeyId": "arn:aws:kms:AWS_Region:123456789012:key/K7MDENG/bPxRfiCYEXAMPLEKEY",
    "ExportTaskIdentifier": "my_export",
    "IamRoleArn": "arn:aws:iam::123456789012:role/export-to-s3",
    "TotalExtractedDataInGB": 0,
    "TaskStartTime": "2019-11-13T19:46:00.173Z",
    "SourceArn": "arn:aws:rds:AWS_Region:123456789012:snapshot:export-example-1"
}
```

API RDS

Per annullare un'operazione di esportazione di snapshot utilizzando l'API Amazon RDS, utilizza l'CancelExportTaskoperazione con il `ExportTaskIdentifier` parametro.

Messaggi di errore per le attività di esportazione di Amazon S3

Nella tabella seguente vengono descritti i messaggi restituiti quando le attività di esportazione di Amazon S3 non riescono.

Messaggio di errore	Descrizione
Si è verificato un errore interno sconosciuto.	L'elaborazione della richiesta non è riuscita a causa di un errore, un'eccezione o un guasto interno sconosciuto.
Si è verificato un errore interno sconosciuto durante la scrittura dei metadati dell'attività di esportazione nel bucket S3 [nome bucket].	L'elaborazione della richiesta non è riuscita a causa di un errore, un'eccezione o un guasto interno sconosciuto.
L'esportazione RDS non è riuscita a scrivere i metadati dell'attività di esportazione perché non può assumere il ruolo IAM [ruolo ARN].	L'attività di esportazione assume il ruolo IAM per verificare se è consentito scrivere metadati nel bucket S3. Se l'attività non può assumere il ruolo IAM, non riesce.
L'esportazione RDS non è riuscita a scrivere i metadati dell'attività di esportazione nel bucket S3 [nome bucket] utilizzando il ruolo IAM [ruolo ARN] con la chiave KMS [ID chiave]. Codice di errore: [codice di errore]	Mancano una o più autorizzazioni, quindi l'attività di esportazione non può accedere al bucket S3. Questo messaggio di errore viene generato quando si riceve uno dei seguenti codici di errore: • `AWSSecurityTokenServiceException` con il codice di errore `AccessDenied` • `AmazonS3Exception` con il codice di erroreNoSuchBucket ,AccessDenied ,KMS.KMSInvalidStateException ,403 Forbidden , oppure KMS.DisabledException

Messaggio di errore	Descrizione
	Questi codici di errore indicano che le impostazioni non sono configurate correttamente per il ruolo IAM, il bucket S3 o la chiave KMS.
Il ruolo IAM [ruolo ARN] non è autorizzato a chiamare [azione S3] sul bucket S3 [nome bucket]. Controlla le tue autorizzazioni e riprova l'esportazione.	La policy IAM è configurata in modo errato. L'autorizzazione per l'azione S3 specifica sul bucket S3 è mancante e questa condizione causa l'esito negativo dell'attività di esportazione.
Controllo chiave KMS non riuscito. Controlla le credenziali sulla tua chiave KMS e riprova.	Controllo delle credenziali della chiave KMS non riuscito.
Controllo delle credenziali S3 non riuscito. Controlla le autorizzazioni per il bucket S3 e la policy IAM.	Il controllo delle credenziali S3 non è riuscito.
Il bucket S3 [nome bucket] non è valido. O non si trova nella corrente Regione AWS o non esiste. Esamina il nome del bucket S3 e riprova l'esportazione.	Bucket S3 non è valido.
Il bucket S3 [nome del bucket] non si trova nella regione corrente. AWS Esamina il nome del bucket S3 e riprova l'esportazione.	Il bucket S3 si trova nella regione sbagliata. AWS

Risoluzione degli errori di autorizzazione PostgreSQL

Quando si esportano i database PostgreSQL in Simple Storage Service (Amazon S3), è possibile che venga visualizzato un errore `PERMISSIONS_DO_NOT_EXIST` che indica che alcune tabelle sono state ignorate. Questo errore si verifica in genere quando l'utente con privilegi avanzati che hai specificato durante la creazione dell'istanza database, non dispone delle autorizzazioni per accedere alle tabelle.

Per risolvere questo errore, eseguire il comando seguente:

```
GRANT ALL PRIVILEGES ON ALL TABLES IN SCHEMA schema_name TO superuser_name
```

Per ulteriori informazioni sui privilegi utente con privilegi avanzati, vedere Privilegi dell'account utente master.

Convenzione di denominazione file

I dati esportati per tabelle specifiche vengono memorizzati nel formato *base_prefix*/*files*, dove il prefisso di base è il seguente:

```
export_identifier/database_name/schema_name.table_name/
```

Ad esempio:

```
export-1234567890123-459/rdststdb/rdststdb.DataInsert_7ADB5D19965123A2/
```

Esistono due convenzioni di denominazione per i file.

- Convenzione attuale:

  ```
  batch_index/part-partition_index-random_uuid.format-based_extension
  ```

  L'indice batch è un numero di sequenza che rappresenta un batch di dati letti dalla tabella. Se non riusciamo a partizionare la tabella in piccoli blocchi da esportare in parallelo, ci saranno più indici batch. La stessa cosa accade se la tabella è partizionata in più tabelle. Ci saranno più indici batch, uno per ciascuna delle partizioni di tabella della tabella principale.

  Se riusciamo a partizionare la tabella in piccoli blocchi da leggere in parallelo, ci sarà solo la cartella batch index. 1

  All'interno della cartella dell'indice batch, ci sono uno o più file Parquet che contengono i dati della tabella. Il prefisso del nome del file Parquet è. `part-`*`partition_index`* Se la tabella è partizionata, ci saranno più file che iniziano con l'indice delle partizioni. `00000`

  Possono esserci delle lacune nella sequenza dell'indice delle partizioni. Ciò accade perché ogni partizione è ottenuta da una query a intervalli nella tabella. Se non ci sono dati nell'intervallo di quella partizione, quel numero di sequenza viene ignorato.

Ad esempio, supponiamo che la `id` colonna sia la chiave primaria della tabella e che i suoi valori minimo e massimo siano e. `100` `1000` Quando proviamo a esportare questa tabella con nove partizioni, la leggiamo con query parallele come le seguenti:

```
SELECT * FROM table WHERE id <= 100 AND id < 200
SELECT * FROM table WHERE id <= 200 AND id < 300
```

Questo dovrebbe generare nove file, da a`part-00000-`*`random_uuid`*`.gz.parquet`. `part-00008-`*`random_uuid`*`.gz.parquet` Tuttavia, se non ci sono righe con ID compresi tra `200` e`350`, una delle partizioni completate è vuota e non viene creato alcun file per essa. Nell'esempio precedente, `part-00001-`*`random_uuid`*`.gz.parquet` non viene creato.

- Convenzione precedente:

```
part-partition_index-random_uuid.format-based_extension
```

È la stessa della convenzione attuale, ma senza il *`batch_index`* prefisso, ad esempio:

```
part-00000-c5a881bb-58ff-4ee6-1111-b41ecff340a3-c000.gz.parquet
part-00001-d7a881cc-88cc-5ab7-2222-c41ecab340a4-c000.gz.parquet
part-00002-f5a991ab-59aa-7fa6-3333-d41eccd340a7-c000.gz.parquet
```

La convenzione di denominazione file è soggetta a modifiche. Pertanto, quando usi le tabelle di destinazione ti consigliamo di leggere tutto quanto riportato all'interno del prefisso di base della tabella.

Conversione dei dati durante l'esportazione in un bucket Simple Storage Service (Amazon S3)

Quando si esporta uno snapshot di database in un bucket Amazon S3, Amazon RDS converte, esporta e memorizza i dati nel formato Parquet. Per ulteriori informazioni su Parquet, consultare il sito Web Apache Parquet.

Parquet archivia tutti i dati in uno dei seguenti tipi primitivi:

- BOOLEAN
- INT32

- INT64
- INT96
- FLOAT
- DOUBLE
- BYTE_ARRAY: un array di byte a lunghezza variabile, noto anche come binario
- FIXED_LEN_BYTE_ARRAY: un array di byte a lunghezza fissa utilizzato quando i valori hanno una dimensione costante

I tipi di dati Parquet sono pochi per ridurre la complessità di lettura e scrittura del formato. Parquet fornisce tipi logici per estendere i tipi primitivi. Un tipo logico viene implementato come annotazione con i dati in un campo di metadati `LogicalType`. L'annotazione di tipo logico spiega come interpretare il tipo primitivo.

Quando il tipo logico `STRING` annota un tipo `BYTE_ARRAY`, indica che l'array di byte deve essere interpretato come una stringa di caratteri con codifica UTF-8. Al termine di un'attività di esportazione, Amazon RDS notifica all'utente se si è verificata una conversione di stringa. I dati sottostanti esportati sono sempre uguali ai dati provenienti dall'origine. Tuttavia, a causa della differenza di codifica in UTF-8, alcuni caratteri potrebbero apparire diversi dall'origine quando vengono letti in strumenti come Athena.

Per ulteriori informazioni, consultare la sezione relativa alle definizioni dei tipi logici di Parquet nella documentazione di Parquet.

Argomenti

- Mappatura dei tipi di dati MySQL e MariaDB su Parquet
- Mappatura dei tipi di dati PostgreSQL su Parquet

Mappatura dei tipi di dati MySQL e MariaDB su Parquet

La tabella seguente mostra la mappature dai tipi di dati MySQL e MariaDB nei tipi di dati Parquet quando i dati vengono convertiti ed esportati in Simple Storage Service (Amazon S3).

Tipo di dati origine	Tipo Parquet primitivo	Annotazione del tipo logico	Note di conversione
Tipi di dati numerici			

Tipo di dati origine	Tipo Parquet primitivo	Annotazione del tipo logico	Note di conversione
BIGINT	INT64		
BIGINT UNSIGNED	FIXED_LEN _BYTE_ARRAY(9)	DECIMAL(20,0)	Parquet supporta solo tipi firmati, quindi la mappatura richiede un byte aggiuntivo (8 più 1) per memorizza re il tipo BIGINT_UN SIGNED.
BIT	BYTE_ARRAY		
DECIMAL	INT32	DECIMAL(p,s)	Se il valore di origine è inferiore a 2^{31}, viene archiviato come INT32.
	INT64	DECIMAL(p,s)	Se il valore di origine è 2^{31} o superiore, ma inferiore a 2^{63}, viene archiviato come INT64.
	FIXED_LEN _BYTE_ARRAY(N)	DECIMAL(p,s)	Se il valore di origine è 2^{63} o superiore , viene archiviato come FIXED_LEN _BYTE_ARRAY(N).

Tipo di dati origine	Tipo Parquet primitivo	Annotazione del tipo logico	Note di conversione
	BYTE_ARRAY	STRING	Parquet non supporta la precisione decimale superiore a 38. Il valore decimale viene convertito in una stringa di tipo BYTE_ARRAY e codificato come UTF8.
DOUBLE	DOUBLE		
FLOAT	DOUBLE		
INT	INT32		
INT UNSIGNED	INT64		
MEDIUMINT	INT32		
MEDIUMINT UNSIGNED	INT64		
NUMERIC	INT32	DECIMAL(p,s)	Se il valore di origine è inferiore a 2^{31}, viene archiviato come INT32.
	INT64	DECIMAL(p,s)	Se il valore di origine è 2^{31} o superiore, ma inferiore a 2^{63}, viene archiviato come INT64.

Tipo di dati origine	Tipo Parquet primitivo	Annotazione del tipo logico	Note di conversione
	FIXED_LEN _ARRAY(N)	DECIMAL(p,s)	Se il valore di origine è 2^{63} o superiore , viene archiviato come FIXED_LEN _BYTE_ARRAY(N).
	BYTE_ARRAY	STRING	Parquet non supporta la precisione numerica superiore a 38. Questo valore numerico viene convertito in una stringa di tipo BYTE_ARRAY e codificato come UTF8.
SMALLINT	INT32		
SMALLINT UNSIGNED	INT32		
TINYINT	INT32		
TINYINT UNSIGNED	INT32		
Tipi di dati stringa			
BINARY	BYTE_ARRAY		
BLOB	BYTE_ARRAY		
CHAR	BYTE_ARRAY		
ENUM	BYTE_ARRAY	STRING	

Tipo di dati origine	Tipo Parquet primitivo	Annotazione del tipo logico	Note di conversione
LINESTRING	BYTE_ARRAY		
LONGBLOB	BYTE_ARRAY		
LONGTEXT	BYTE_ARRAY	STRING	
MEDIUMBLOB	BYTE_ARRAY		
MEDIUMTEXT	BYTE_ARRAY	STRING	
MULTILINESTRING	BYTE_ARRAY		
SET	BYTE_ARRAY	STRING	
TEXT	BYTE_ARRAY	STRING	
TINYBLOB	BYTE_ARRAY		
TINYTEXT	BYTE_ARRAY	STRING	
VARBINARY	BYTE_ARRAY		
VARCHAR	BYTE_ARRAY	STRING	
Tipi di dati data e ora			
DATE	BYTE_ARRAY	STRING	Una data viene convertita in una stringa di tipo BYTE_ARRAY e codificata come UTF8.
DATETIME	INT64	TIMESTAMP _MICROS	

Tipo di dati origine	Tipo Parquet primitivo	Annotazione del tipo logico	Note di conversione
TIME	BYTE_ARRAY	STRING	Un tipo TIME viene convertito in una stringa di tipo BYTE_ARRAY e codificato come UTF8.
TIMESTAMP	INT64	TIMESTAMP _MICROS	
YEAR	INT32		
Tipi di dati geometrici			
GEOMETRY	BYTE_ARRAY		
GEOMETRYC OLLECTION	BYTE_ARRAY		
MULTIPOINT	BYTE_ARRAY		
MULTIPOLYGON	BYTE_ARRAY		
POINT	BYTE_ARRAY		
POLYGON	BYTE_ARRAY		
Tipo di dati JSON			
JSON	BYTE_ARRAY	STRING	

Mappatura dei tipi di dati PostgreSQL su Parquet

Nella tabella seguente viene illustrata la mappatura dai tipi di dati PostgreSQL ai tipi di dati Parquet quando i dati vengono convertiti ed esportati in Simple Storage Service (Amazon S3).

Tipo di dati PostgreSQL	Tipo Parquet primitivo	Annotazione del tipo logico	Note relative alla mappatura
Tipi di dati numerici			
BIGINT	INT64		
BIGSERIAL	INT64		
DECIMAL	BYTE_ARRAY	STRING	Un tipo DECIMAL viene convertito in una stringa di tipo BYTE_ARRAY e codificato come UTF8. Questa conversione serve a evitare complicazioni dovute alla precisione dei dati e ai valori di dati che non sono un numero (NaN).
DOUBLE PRECISION	DOUBLE		
INTEGER	INT32		
MONEY	BYTE_ARRAY	STRING	
REAL	FLOAT		
SERIAL	INT32		
SMALLINT	INT32	INT_16	
SMALLSERIAL	INT32	INT_16	
Tipi di dati stringa e correlati			

Tipo di dati PostgreSQL	Tipo Parquet primitivo	Annotazione del tipo logico	Note relative alla mappatura
ARRAY	BYTE_ARRAY	STRING	Un array viene convertito in una stringa e codificato come BINARY (UTF8). Questa conversione serve a evitare complicazioni dovute alla precisione dei dati, ai valori di dati che non sono un numero (NaN) e ai valori di dati temporali.
BIT	BYTE_ARRAY	STRING	
BIT VARYING	BYTE_ARRAY	STRING	
BYTEA	BINARY		
CHAR	BYTE_ARRAY	STRING	
CHAR(N)	BYTE_ARRAY	STRING	
ENUM	BYTE_ARRAY	STRING	
NAME	BYTE_ARRAY	STRING	
TEXT	BYTE_ARRAY	STRING	
TEXT SEARCH	BYTE_ARRAY	STRING	
VARCHAR(N)	BYTE_ARRAY	STRING	

Tipo di dati PostgreSQL	Tipo Parquet primitivo	Annotazione del tipo logico	Note relative alla mappatura
XML	BYTE_ARRAY	STRING	
Tipi di dati data e ora			
DATE	BYTE_ARRAY	STRING	
INTERVAL	BYTE_ARRAY	STRING	
TIME	BYTE_ARRAY	STRING	
TIME WITH TIME ZONE	BYTE_ARRAY	STRING	
TIMESTAMP	BYTE_ARRAY	STRING	
TIMESTAMP WITH TIME ZONE	BYTE_ARRAY	STRING	
Tipi di dati geometrici			
BOX	BYTE_ARRAY	STRING	
CIRCLE	BYTE_ARRAY	STRING	
LINE	BYTE_ARRAY	STRING	
LINESEGMENT	BYTE_ARRAY	STRING	
PATH	BYTE_ARRAY	STRING	
POINT	BYTE_ARRAY	STRING	
POLYGON	BYTE_ARRAY	STRING	
Tipi di dati JSON			
JSON	BYTE_ARRAY	STRING	
JSONB	BYTE_ARRAY	STRING	

Tipo di dati PostgreSQL	Tipo Parquet primitivo	Annotazione del tipo logico	Note relative alla mappatura
Altri tipi di dati			
BOOLEAN	BOOLEAN		
CIDR	BYTE_ARRAY	STRING	Tipo di dati di rete
COMPOSITE	BYTE_ARRAY	STRING	
DOMAIN	BYTE_ARRAY	STRING	
INET	BYTE_ARRAY	STRING	Tipo di dati di rete
MACADDR	BYTE_ARRAY	STRING	
OBJECT IDENTIFIER	N/A		
PG_LSN	BYTE_ARRAY	STRING	
RANGE	BYTE_ARRAY	STRING	
UUID	BYTE_ARRAY	STRING	

Utilizzo AWS Backup per gestire i backup automatici

AWS Backup è un servizio di backup completamente gestito che semplifica la centralizzazione e l'automazione del backup dei dati tra i AWS servizi nel cloud e in locale. I backup dei database Amazon RDS possono essere facilmente gestiti in AWS Backup.

> **Note**
>
> I backup gestiti da AWS Backup sono considerati snapshot DB manuali, ma non vengono conteggiati ai fini della quota di snapshot DB per RDS. I nomi dei backup creati con terminano con AWS Backup . `awsbackup:`*`backup-job-number`*

Per ulteriori informazioni in merito AWS Backup, consulta la Guida per gli AWS Backup sviluppatori.

Per visualizzare i backup gestiti da AWS Backup

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, selezionare Snapshots (Snapshot).
3. Seleziona la scheda Servizio di backup.

   I tuoi AWS Backup backup sono elencati nella sezione Istantanee del servizio di backup.

Monitoraggio di parametri in un'istanza Amazon RDS

Nelle sezioni seguenti, è possibile trovare una panoramica del monitoraggio Amazon RDS e una spiegazione su come accedere ai parametri. Per informazioni su come monitorare eventi, registri e flussi di attività del database, consulta Monitoraggio di eventi, registri e flussi in un'istanza di database Amazon RDS .

Argomenti

Panoramica del monitoraggio dei parametri di Amazon RDS

Il monitoraggio è importante per garantire l'affidabilità, la disponibilità e le prestazioni di Amazon RDS e delle soluzioni AWS. Per eseguire più facilmente il debug di errori in più punti, ti consigliamo di raccogliere i dati di monitoraggio di tutte le parti della soluzione AWS.

Argomenti

Piano di monitoraggio

Prima di iniziare il monitoraggio di Amazon RDS, crea un piano di monitoraggio. Questo piano deve rispondere alle domande seguenti:

- Quali sono gli obiettivi del monitoraggio?
- Quali risorse verranno monitorate?
- Con quale frequenza eseguirai il monitoraggio di queste risorse?
- Quali strumenti di monitoraggio verranno usati?
- Chi eseguirà le attività di monitoraggio?
- Chi deve ricevere la notifica quando si verifica un problema?

Baseline delle prestazioni

Per raggiungere gli obiettivi di monitoraggio è necessario stabilire una baseline. Pertanto devi misurare le prestazioni in condizioni di carico diverse e in diversi momenti nell'ambiente Amazon RDS. Puoi monitorare parametri come i seguenti:

- Throughput di rete
- Connessioni client
- I/O per operazioni di lettura, scrittura o metadati
- Saldi credito burst per le istanze database

Ti consigliamo di archiviare i dati cronologici delle prestazioni per Amazon RDS. Utilizzando i dati archiviati puoi confrontare le prestazioni correnti con le tendenze passate. Puoi distinguere i normali modelli di prestazioni dalle anomalie e definire i metodi per risolvere i problemi.

Linee guida per le prestazioni

In generale, i valori accettabili per i parametri delle prestazioni dipendono dalle attività dell'applicazione in relazione alla tua baseline. Indagare le variazioni della baseline coerenti o che rappresentano dei trend. I seguenti parametri sono spesso fonte di problemi di prestazioni:

- Consumo elevato di CPU o RAM – Valori elevati per il consumo di CPU o RAM potrebbero essere appropriati, purché tengano conto degli obiettivi dell'applicazione (come throughput o concorrenza) e siano previsti.
- Consumo dello spazio su disco: esamina il consumo dello spazio su disco se lo spazio usato supera costantemente l'85% dello spazio su disco totale. Verifica se è possibile eliminare dati dall'istanza o archiviare dati su un sistema diverso per liberare spazio.
- Traffico di rete – Per il traffico di rete, rivolgiti al tuo amministratore di sistema per identificare il throughput previsto per la rete del dominio e la connessione Internet. Indaga il traffico di rete se il throughput è costantemente al di sotto del valore previsto.
- Connessioni al database – Se noti un numero elevato di connessioni utente insieme a un peggioramento delle prestazioni e del tempo di risposta dell'istanza, valuta se limitare le connessioni al database . Il numero ideale di connessioni utente per l'istanza database dipende dalla classe di istanza e dalla complessità delle operazioni eseguite. Per determinare il numero di connessioni di database, associa l'istanza database a un gruppo di parametri dove il parametro `User Connections` è impostato su un valore diverso da 0 (illimitato). Puoi utilizzare un gruppo di parametri esistente o crearne uno nuovo. Per ulteriori informazioni, consulta Utilizzo di gruppi di parametri.
- Parametri di IOPS: poiché i valori previsti per i parametri di IOPS dipendono dalle specifiche del disco e dalla configurazione del server, usa i valori di riferimento per identificare i comportamenti tipici. Verifica se i valori sono costantemente diversi dalla baseline. Per prestazioni IOPS ottimali, verifica che il working set tipico possa essere caricato nella memoria per ridurre al minimo le operazioni di lettura e scrittura.

Quando le prestazioni non rientrano nella baseline stabilita, potrebbe essere necessario apportare modifiche per ottimizzare la disponibilità del database per il carico di lavoro. Ad esempio, potrebbe essere necessario modificare la classe di istanza dell'istanza database. In alternativa, potrebbe

essere necessario modificare il numero di istanze database e leggere le repliche disponibili per i client.

Strumenti di monitoraggio

Il monitoraggio è importante per mantenere l'affidabilità, la disponibilità e le prestazioni di Amazon RDS e delle altre soluzioni AWS. AWS fornisce strumenti di monitoraggio per controllare Amazon RDS, segnalare eventuali problemi ed eseguire operazioni automatiche quando appropriato.

Argomenti

Strumenti di monitoraggio automatici

Si consiglia di automatizzare il più possibile i processi di monitoraggio.

Argomenti

Stato di istanza di Amazon RDS e suggerimenti

Per controllare Amazon RDS e segnalare l'eventuale presenza di problemi, puoi usare gli strumenti automatici seguenti:

- Stato dell’istanza di Amazon RDS: visualizzare i dettagli sullo stato corrente dell’istanza utilizzando la console Amazon RDS, la AWS CLI o l’API RDS.
- Raccomandazioni di Amazon RDS — Rispondi alle raccomandazioni automatiche per le risorse di database, come istanze DB, le repliche di lettura e gruppi di parametri del database. Per ulteriori informazioni, consulta Visualizzazione e risposta ai consigli di RDS.

CloudWatch

Amazon RDS Aurora si integra con CloudWatch Amazon per funzionalità di monitoraggio aggiuntive.

- Amazon CloudWatch: questo servizio monitora AWS le tue risorse e le applicazioni su cui esegui AWS in tempo reale. Puoi utilizzare le seguenti CloudWatch funzionalità di Amazon con Amazon RDS :
  - CloudWatch Parametri Amazon: Amazon RDS invia automaticamente i parametri ogni minuto CloudWatch per ogni database attivo. Non sono previsti costi aggiuntivi per i parametri di Amazon RDS in. CloudWatch Per ulteriori informazioni, consulta Monitoraggio dei parametri di Amazon RDS con Amazon CloudWatch.
  - CloudWatch Allarmi Amazon: puoi controllare una singola metrica Amazon RDS in un periodo di tempo specifico. È quindi possibile eseguire una o più operazioni in base al valore del parametro rispetto a una soglia impostata. Per ulteriori informazioni, consulta Monitoraggio dei parametri di Amazon RDS con Amazon CloudWatch.

Amazon RDS Performance Insights e monitoraggio del sistema operativo

Per monitorare le prestazioni di Amazon RDS, puoi usare i seguenti strumenti automatici:

- Performance Insights Amazon RDS – Aiuta a valutare in modo rapido il carico del database e a determinare quando e dove intervenire. Per ulteriori informazioni, consulta Monitoraggio del carico DB con Performance Insights su Amazon RDS.
- Monitoraggio avanzato di Amazon RDS – Osserva i parametri in tempo reale per il sistema operativo. Per ulteriori informazioni, consulta Monitoraggio dei parametri del sistema operativo con il monitoraggio avanzato.

Servizi integrati

I seguenti servizi AWS sono integrati con Amazon RDS:

- Amazon EventBridge è un servizio di bus eventi senza server che semplifica la connessione delle applicazioni con dati provenienti da una varietà di fonti. Per ulteriori informazioni, consulta Monitoraggio di eventi Amazon RDS.
- Amazon CloudWatch Logs consente di monitorare, archiviare e accedere ai file di log da istanze Amazon RDS CloudTrail, e altre fonti. Per ulteriori informazioni, consulta Monitoraggio dei file di log di Amazon RDS.
- AWS CloudTrail acquisisce le chiamate API e gli eventi correlati effettuati da o per conto del tuo Account AWSe fornisce i file di log a un bucket Simple Storage Service (Amazon S3) specificato.

Per ulteriori informazioni, consulta Monitoraggio delle chiamate API di Amazon RDS in AWS CloudTrail.

- Database Activity Streams Per ulteriori informazioni, consulta Monitoraggio di Amazon RDS tramite i flussi di attività del database.

Strumenti di monitoraggio manuali

È necessario monitorare manualmente gli elementi non coperti dagli allarmi. CloudWatch Amazon RDS AWS Trusted Advisor e CloudWatch le altre dashboard AWS della console forniscono una at-a-glance panoramica dello stato del tuo AWS ambiente. Consigliamo anche di controllare i file di log nell'istanza database.

- Dalla console Amazon RDS, puoi monitorare i seguenti elementi per le risorse:
  - Il numero di connessioni a un'istanza database
  - Il numero di operazioni di lettura e scrittura a un'istanza database
  - Quantità di storage utilizzato al momento dall'istanza database
  - La quantità di memoria e CPU utilizzati per un'istanza database
  - La quantità di traffico di rete verso e da un'istanza database
- Dal pannello di controllo Trusted Advisor, puoi rivedere i seguenti controlli di ottimizzazione dei costi, sicurezza, tolleranza ai guasti e miglioramento delle prestazioni:
  - Istanze database Amazon RDS inattive
  - Rischio accesso gruppo di sicurezza Amazon RDS
  - Backup Amazon RDS
  - Multi-AZ Amazon RDS

  Per ulteriori informazioni su questi controlli, consulta best practice Trusted Advisor (Controlli).
- CloudWatch la home page mostra:
  - Stato e allarmi attuali
  - Grafici degli allarmi e delle risorse
  - Stato di integrità dei servizi

  Inoltre, è possibile utilizzare CloudWatch per effettuare le seguenti operazioni:
  - Crea pannelli di controllo personalizzati per monitorare i servizi rilevanti.
  - Creare grafici dei dati dei parametri per la risoluzione di problemi e il rilevamento di tendenze.

- Ricercare e analizzare tutti i parametri delle risorse AWS.
- Creare e modificare gli allarmi per ricevere le notifiche dei problemi.

Visualizzazione dello stato dell'istanza del

Utilizzando la console Amazon RDS, puoi accedere rapidamente allo stato della tua istanza del DB.

Argomenti

Visualizzazione dello stato dell'istanza database di Amazon RDS

Lo stato di un'istanza database indica l'integrità dell'istanza db. Puoi utilizzare le seguenti procedure per visualizzare lo stato nella console Amazon RDS, nel AWS CLI comando o nell'operazione API.

> **Note**
>
> Amazon RDS usa anche un altro stato denominato stato di manutenzione, mostrato nella colonna Maintenance (Manutenzione) della console Amazon RDS. Questo valore indica lo stato delle patch di manutenzione da applicare a un'istanza database. Lo stato della manutenzione è indipendente dallo stato dell'istanza database. Per ulteriori informazioni sullo stato della manutenzione, consulta Applicazione di aggiornamenti a un'istanza database.

I valori di stato possibili per le Istanze DB nella tabella seguente. Nella tabella viene inoltre indicato se è prevista la fatturazione per l'istanza db e lo storage, o solo per lo storage oppure se la fatturazione non è prevista. Per tutti gli stati delle istanze database, l'utilizzo del backup viene inserito in fattura.

Stato istanza database	Fattura	Descrizione
Disponibilità	Fattura	L'istanza database è integra e disponibile.
Backup	Fattura	L'istanza database è attualmente sottoposta a backup.
C onfiguring-enhance d-monitoring	Fattura	Il monitoraggio avanzato per questa istanza database è in fase di abilitazione/disabilitazione.
C onfiguring-iam-dat abase-auth	Fattura	AWS Identity and Access Management (IAM) l'autenticazione del database è abilitata o disabilitata per questa istanza DB.
C onfiguring-log-exp orts	Fattura	La pubblicazione dei file di log su Amazon CloudWatch Logs è abilitata o disabilitata per questa istanza DB.
C onverting-to-vpc	Fattura	È in corso la conversione dell'istanza database da un'istanz a database che non si trova in un Amazon Virtual Private Cloud (Amazon VPC) a un'istanza database che si trova in un Amazon VPC.

Stato istanza database	Fattura	Descrizione
Creating (Creazione in corso)	Non fatturata	È in corso la creazione dell'istanza database. Non è possibile accedere all'istanza database mentre è in fase di creazione.
Elimina precontrollo	Non fatturata	Amazon RDS sta verificando se le repliche di lettura sono integre e sicure per l'eliminazione.
Deleting (Eliminazione in corso)	Non fatturata	È in corso l'eliminazione dell'istanza database.
Failed (Non riuscito)	Non fatturata	L'istanza database ha restituito un errore e Amazon RDS non è stato in grado di recuperarla. Esegui un point-in-time ripristin o all'ora di ripristino più recente dell'istanza DB per recuperare i dati.
I naccessible-encryp tion-credentials	Non fatturata	Non è possibile accedere o recuperare l'istanza DB AWS KMS key utilizzata per crittografare o decrittografare.
I naccessible-encryp tion-credentials-r ecoverable	Fattura per storage	Non è possibile accedere alla chiave KMS utilizzata per crittografare o decrittografare l'istanza database. Tuttavia, se la chiave KMS è attiva, il riavvio dell'istanza database può ripristin arla. Per ulteriori informazioni, consulta Crittografia di un'istanza database.
Incompatible-network (Rete incompatibile)	Non fatturata	Il tentativo di Amazon RDS di eseguire un'operazione di ripristin o su un'istanza database ha esito negativo perché il VPC si trova in uno stato che impedisce il completamento dell'oper azione. Questo stato si verifica ad esempio se tutti gli indirizzi IP di una sottorete sono in uso e Amazon RDS non è in grado di ottenere un indirizzo IP per l'istanza database.

Stato istanza database	Fattura	Descrizione
io ncompatible-option-group	Fattura	Amazon RDS ha tentato di applicare una modifica a un gruppo di opzioni, ma non può farlo e non può eseguire il rollback allo stato del gruppo di opzioni precedente. Per ulteriori informazioni, consulta l'elenco Recent Events (Eventi recenti) per l'istanza database. Questo stato si verifica ad esempio se il gruppo di opzioni contiene un'opzione come TDE e l'istanza database non contiene informazioni crittografate.
Incompatible-parameters (Parametri incompatibili)	Fattura	Amazon RDS non è in grado di avviare l'istanza database perché i parametri specificati nel gruppo di parametri database non sono compatibili con l'istanza database. È necessario annullare le modifiche apportate ai parametri o rendere tali parametri compatibili con l'istanza database per ottenere di nuovo l'accesso all'istanza database. Per ulteriori informazioni sui parametri incompatibili, consulta l'elenco Recent Events (Eventi recenti) per l'istanza database.
Incompatible-restore (Ripristino incompatibile)	Non fatturata	Amazon RDS non può eseguire un point-in-time ripristino. Tra le cause più comuni di questo stato è incluso l'uso di tabelle temporanee, , i tabelle MyISAM con MySQL di tabelle Aria con MariaDB.
Insufficient-capacity	Non fatturata	Amazon RDS non può creare l'istanza perché al momento non è disponibile una capacità sufficiente. Per creare l'istanza database nella stessa AZ con lo stesso tipo di istanza, elimina l'istanza database, attendi qualche ora e prova a crearla di nuovo. In alternativa, crea una nuova istanza utilizzando una classe di istanza o una AZ diversa.
Maintenance (Manutenzione)	Fattura	È in corso l'applicazione da parte di Amazon RDS di un aggiornamento di manutenzione all'istanza database. Questo stato viene utilizzato per la manutenzione a livello di istanza pianificata in anticipo da RDS.
Modifying (Modifica in corso)	Fattura	È in corso la modifica dell'istanza database in seguito alla richiesta da parte di un cliente.

Stato istanza database	Fattura	Descrizione
M oving-to-vpc	Fattura	È in corso lo spostamento dell'istanza database in un nuovo Amazon Virtual Private Cloud (Amazon VPC).
Reebooting (Riavvio in corso)	Fattura	È in corso il riavvio dell'istanza database a causa della richiesta di un cliente o perché è necessario per un processo Amazon RDS.
R esetting-master-cr edentials	Fattura	È in corso il ripristino delle credenziali master dell'istanza database in seguito alla richiesta da parte di un cliente.
Ridenominazione	Fattura	È in corso la ridenominazione dell'istanza database in seguito alla richiesta da parte di un cliente.
Restore-error (Errore ripristino)	Fattura	L'istanza DB ha rilevato un errore nel tentativo di ripristinare una point-in-time o da un'istantanea.
Avvio di	Fattura per storage	L'istanza database è in fase di avvio.
Arrestate	Fattura per storage	L'istanza database è stata arrestata.
Stopping (In arresto)	Fattura per storage	L'istanza database è in fase di arresto.
S torage-config-upgr ade	Fattura	La configurazione del file system di archiviazione dell'istanza database è in fase di aggiornamento. Questo stato si applica solo ai database verdi all'interno di un'implementazione blu/verde o alle repliche di lettura delle istanze database.

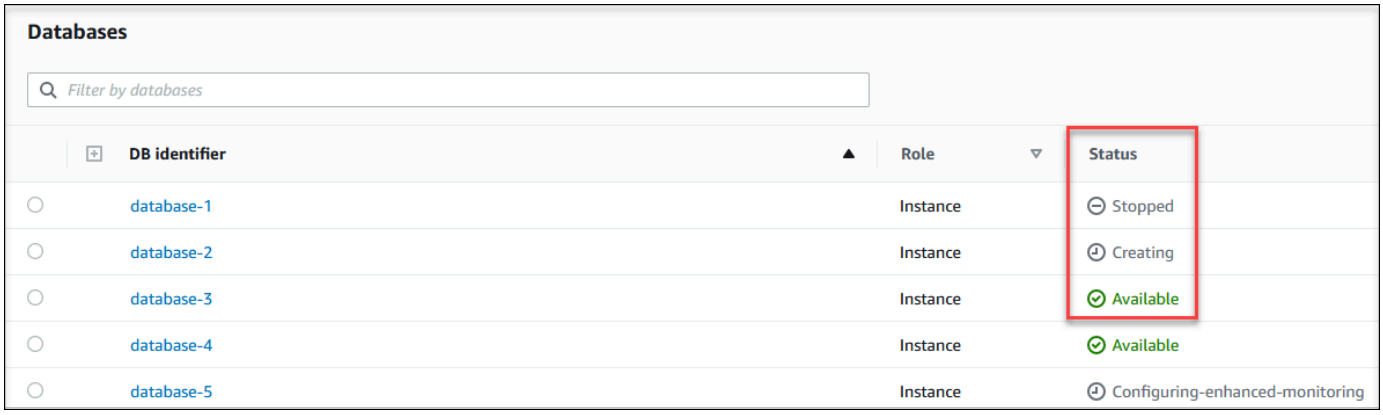
Stato istanza database	Fattura	Descrizione
Storage-full (Archiviazione piena)	Fattura	L'istanza database ha raggiunto la capacità di storage allocata. Si tratta di uno stato critico che richiede l'immediata risoluzione del problema. A tale scopo, è necessario aumentare la capacità di storage modificando l'istanza database. Per evitare questa situazione, imposta gli CloudWatch allarmi di Amazon per avvisarti quando lo spazio di archiviazione si sta esaurendo .
Storage-optimization (Ottimizzazione archiviazione)	Fattura	Amazon RDS sta ottimizzando lo storage dell'istanza database. L'istanza database è completamente operativa. Il processo di ottimizzazione dello storage è solitamente breve, ma a volte può richiedere oltre 24 ore.
Aggiornamento	Fattura	La versione del motore di database è in fase di aggiornamento.

Console

Per visualizzare lo stato di un'istanza database

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, scegli Databases (Database).

   La pagina Databases (Database) viene visualizzata con l'elenco delle istanze database. Per ogni istanza database , viene visualizzato il valore dello stato.



CLI

Per visualizzare l'istanza DB e le relative informazioni sullo stato utilizzando il AWS CLI, usa il describe-db-instancescomando. Ad esempio, il AWS CLI comando seguente elenca tutte le informazioni sulle istanze DB.

```
aws rds describe-db-instances
```

Per visualizzare un'istanza DB specifica e il relativo stato, chiamate il describe-db-instancescomando con l'opzione seguente:

- `DBInstanceIdentifier` – Il nome dell'istanza database.

```
aws rds describe-db-instances --db-instance-identifier mydbinstance
```

Per visualizzare solo lo stato di tutte le istanze DB, usa la seguente query in AWS CLI.

```
aws rds describe-db-instances --query 'DBInstances[*].
[DBInstanceIdentifier,DBInstanceStatus]' --output table
```

API

Per visualizzare lo stato dell'istanza database usando l'API Amazon RDS, chiama l'operazione DescribeDBInstances.

Visualizzazione e risposta ai consigli di RDS

Amazon RDS Aurora fornisce consigli automatici per le risorse di database, come istanze DB, repliche di lettura e gruppi di parametri DB. Queste raccomandazioni forniscono consigli sulle best practice analizzando la configurazione delle istanze database, l'utilizzo e i dati sulle prestazioni.

Amazon RDS Performance Insights monitora parametri specifici e crea automaticamente soglie analizzando quali livelli sono considerati potenzialmente problematici per una risorsa specifica. Quando i nuovi valori delle metriche superano una soglia predefinita in un determinato periodo di tempo, Performance Insights genera una raccomandazione proattiva. Questa raccomandazione aiuta a prevenire futuri impatti sulle prestazioni del database. Ad esempio, la raccomandazione «Idle In Transaction» viene generata per RDS per le istanze PostgreSQL Aurora PostgreSQL non svolgono attività attive, ma possono mantenere bloccate le risorse del database. Per ricevere consigli proattivi, devi attivare Performance Insights con un periodo di conservazione a pagamento. Per informazioni sull'attivazione di Performance Insights, consultaAttivazione e disattivazione di Performance Insights. Per informazioni sui prezzi e sulla conservazione dei dati per Performance Insights, vederePrezzi e conservazione dei dati per Performance Insights.

DevOpsGuru for RDS monitora determinate metriche per rilevare quando il comportamento della metrica diventa molto insolito o anomalo. Queste anomalie vengono segnalate come approfondimenti reattivi con raccomandazioni. Ad esempio, DevOps Guru for RDS potrebbe consigliarti di prendere in considerazione l'aumento della capacità della CPU o di analizzare gli eventi di attesa che contribuiscono al carico del DB. DevOpsGuru for RDS fornisce anche consigli proattivi basati su soglie. Per questi consigli, devi attivare DevOps Guru for RDS. Per informazioni sull'attivazione di DevOps Guru for RDS, consulta. Attivare DevOps Guru e specificare la copertura delle risorse

I consigli avranno uno dei seguenti stati: attivi, ignorati, in sospeso o risolti. I consigli risolti sono disponibili per 365 giorni.

È possibile visualizzare o ignorare i consigli. È possibile applicare immediatamente un consiglio attivo basato sulla configurazione, programmarlo nella finestra di manutenzione successiva o ignorarlo. Per i consigli proattivi basati sulla soglia e quelli reattivi basati sull'apprendimento automatico, è necessario esaminare la causa suggerita del problema e quindi eseguire le azioni consigliate per risolverlo.

Argomenti

- Visualizzazione dei suggerimenti Amazon RDS
- Risposta alle raccomandazioni Amazon RDS

Visualizzazione dei suggerimenti Amazon RDS

Amazon RDS genera suggerimenti per una risorsa quando questa viene creata o modificata.

I consigli basati sulla configurazione sono supportati nelle seguenti regioni:

- Stati Uniti orientali (Ohio)
- Stati Uniti orientali (Virginia settentrionale)
- Stati Uniti occidentali (California settentrionale)
- Stati Uniti occidentali (Oregon)
- Asia Pacifico (Mumbai)
- Asia Pacifico (Seoul)
- Asia Pacifico (Singapore)
- Asia Pacifico (Sydney)
- Asia Pacifico (Tokyo)
- Canada (Centrale)
- Europa (Francoforte)
- Europa (Irlanda)
- Europe (London)
- Europa (Parigi)
- Sud America (San Paolo)

Nella tabella seguente sono disponibili esempi di consigli basati sulla configurazione.

Type	Descrizione	Raccomandazione	Tempo di inattività richiest	Informazioni aggiuntive
Il volume magnetico è in uso	Le tue istanze DB utilizzano l'archiviazione magnetica. L'archiviazione	Scegli un tipo di archiviazione diverso: General Purpose	Sì	Volumi di generazione precedente nella documentazione di Amazon EC2.

Type	Descrizione	Raccomandazione	Tempo di inattività richiest	Informazioni aggiuntive
	magnetica non è consigliata per la maggior parte delle istanze DB. Scegli un tipo di storage diverso: General Purpose (SSD) o Provisioned IOPS.	(SSD) o Provisioned IOPS.		
I backup di Resource Automated sono disattivati	I backup automatici non sono attivati per le istanze DB. I backup automatici sono consigliati perché consentono il point-in-time ripristino delle istanze DB.	Attiva i backup automatici con un periodo di conservazione fino a 14 giorni.	Sì	Abilitazione dei backup automatici Demistificazione dei costi dello storage di backup di Amazon RDS sul Database Blog AWS
È richiesto l'aggiornamento della versione secondaria del motore	Le risorse del database non eseguono l'ultima versione secondaria del motore DB. L'ultima versione secondaria contiene le ultime correzioni di sicurezza e altri miglioramenti.	Esegui l'aggiornamento alla versione più recente del motore.	Sì	Aggiornamento della versione del motore di un'istanza database

Type	Descrizione	Raccomandazione	Tempo di inattività richiest	Informazioni aggiuntive
Il monitoraggio avanzato è disattivato	Il monitoraggio avanzato non è attivato per le risorse del database. Il monitoraggio avanzato offre i parametri del sistema operativo in tempo reale per il monitoraggio e la risoluzione dei problemi.	Attiva il monitoraggio avanzato.	No	Monitoraggio dei parametri del sistema operativo con il monitoraggio avanzato

Type	Descrizione	Raccomandazione	Tempo di inattività richiest	Informazioni aggiuntive
La crittografia dello storage è disattivata	Amazon RDS supporta la crittografia a riposo per tutti i motori di database utilizzando le chiavi gestite in AWS Key Management Service (AWS KMS). Su un'istanza DB attiva con crittografia Amazon RDS, i dati archiviati a riposo nello storage sono crittografati, in modo simile ai backup automatici, alle repliche di lettura e alle istantanee. Se la crittografia non è attivata durante la creazione di un'istanza DB, dovrai creare e ripristinare una copia crittografata dello snapshot decrittografato dell'istanza DB prima di attivare la crittografia.	Attiva la crittografia dei dati inattivi per la tua istanza DB.	Sì	Sicurezza in Amazon RDS Copia di una snapshot DB.

Type	Descrizione	Raccomandazione	Tempo di inattività richiesto	Informazioni aggiuntive
Performance Insights è disattivato	Performance Insights monitora il carico dell'istanza DB per aiutarti ad analizzare e risolvere i problemi di prestazioni del database. Ti consigliamo di attivare Performance Insights.	Attivare Performance Insights.	No	Monitoraggio del carico DB con Performance Insights su Amazon RDS
La scalabilità automatica dello storage sulle istanze DB è disattivata	La scalabilità automatica dello storage non è attivata per l'istanza DB. Quando il carico di lavoro del database aumenta, la scalabilità automatica dello storage RDS ridimensiona automaticamente la capacità di archiviazione senza tempi di inattività.	Attiva la scalabilità automatica dello storage Amazon RDS con una soglia di storage massima specificata	No	Gestione della capacità automaticamente con Auto Scaling dello storage Amazon RDS

Type	Descrizione	Raccomandazione	Tempo di inattività richiest	Informazioni aggiuntive
È richiesto l'aggiornamento delle versioni principali delle risorse RDS	I database con la versione principale corrente per il motore DB non saranno supportati. Ti consigliamo di eseguire l'aggiornamento alla versione principale più recente che include nuove funzionalità e miglioramenti.	Esegui l'aggiornamento alla versione principale più recente per il motore DB.	Sì	Aggiornamento della versione del motore di un'istanza database Utilizzo delle implementazioni blu/verde Amazon RDS per gli aggiornamenti del database
È richiesto l'aggiornamento della classe di istanza delle risorse RDS	L'istanza DB esegue una classe di istanza DB di generazione precedente. Abbiamo sostituito le classi di istanze DB di una generazione precedente con classi di istanze DB con costi e prestazioni migliori o entrambi. Ti consigliamo di eseguire l'istanza DB con una classe di istanza DB di nuova generazione.	Aggiorna la classe di istanza DB.	Sì	Motori DB supportati per classi di istanza database

Type	Descrizione	Raccomandazione	Tempo di inattività richiesto	Informazioni aggiuntive
Risorse RDS che utilizzano l'edizione End of Support Engine con licenza inclusa	Ti consigliamo di aggiornare la versione principale all'ultima versione del motore supportata da Amazon RDS per continuare con il supporto della licenza corrente. La versione del motore del database non sarà supportata con la licenza corrente.	Ti consigliamo di aggiornare il database all'ultima versione supportata in Amazon RDS per continuare a utilizzare il modello con licenza.	Sì	Aggiornamenti a una versione principale Oracle

Type	Descrizione	Raccomandazione	Tempo di inattività richiest	Informazioni aggiuntive
Istanze DB che non utilizzano l'implementazione Multi-AZ	Consigliamo di usare l'implementazione multi-AZ. Le implementazioni multi-AZ migliorano la disponibilità e la durabilità dell'istanza database.	Configura Multi-AZ per le istanze DB interessate	No Durante questa modifica non si verifica un'interruzione. Tuttavia è possibile riscontrare un impatto sulle prestazioni. Per ulteriori informazioni, consulta Trasformazione di un'istanza database in	Prezzi per Amazon RDS Multi-AZ

Type	Descrizione	Raccomandazione	Tempo di inattività richiest	Informazioni aggiuntive
			implementazione d'istanza database Multi-AZ	
I parametri di memoria DB sono diversi da quelli predefiniti	I parametri di memoria delle istanze DB sono significativamente diversi dai valori predefiniti. Queste impostazioni possono influire sulle prestazioni e causare errori. Si consiglia di ripristinare i parametri di memoria personalizzati per l'istanza DB ai valori predefiniti nel gruppo di parametri DB.	Reimposta i parametri di memoria ai valori predefiniti.	No	Le migliori pratiche per la configurazione dei parametri prestazionali per Amazon RDS for MySQL sul database blog AWS

Type	Descrizione	Raccomandazione	Tempo di inattività richiest	Informazioni aggiuntive
`InnoDB_Change_Buffering` parame che utilizza un valore inferiore a quello ottimale	Il buffering delle modifiche consente a un'istanza DB MySQL di posticipare alcune scritture, necessarie per mantenere gli indici secondari. Questa funzionalità era utile in ambienti con dischi lenti. La modifica della configurazione del buffering ha migliorato leggermente le prestazioni del DB, ma ha causato un ritardo nel ripristino in caso di arresto anomalo e lunghi tempi di spegnimento durante l'aggiornamento.	Imposta `InnoDB_Change_Buffering` il valore del parametro su `NONE` nei gruppi di parametri del database.	No	Le migliori pratiche per la configurazione dei parametri prestazionali per Amazon RDS for MySQL sul database blog AWS

Type	Descrizione	Raccomandazione	Tempo di inattività richiest	Informazioni aggiuntive
Il parametro della cache delle query è attivato	Quando le modifiche richiedono l'eliminazione della cache delle query, l'istanza DB sembrerà bloccarsi. La maggior parte dei carichi di lavoro non beneficia della cache delle query. La cache delle query è stata rimossa da MySQL versione 8.0. Ti consigliamo di impostare il parametro query_cache_type su 0.	Imposta il valore del `query_cache_type` parametro su nei gruppi di parametri del database`0`.	Sì	Le migliori pratiche per la configurazione dei parametri prestazionali per Amazon RDS for MySQL sul database blog AWS

Type	Descrizione	Raccomandazione	Tempo di inattività richiest	Informazioni aggiuntive
`log_output` il parametro è impostato su tabella	Quando `log_output` è impostato su`TABLE`, viene utilizzato più spazio di archiviazione rispetto a quando `log_output` è impostato su`FILE`. Si consiglia di impostare il parametro su`FILE`, per evitare di raggiungere il limite di dimensione di archiviazione.	Imposta il valore del `log_output` parametro su `FILE` nei gruppi di parametri del database.	No	File di log del database MySQL
Gruppi di parametri che non utilizzano pagine enormi	Le pagine di grandi dimensioni possono aumentare la scalabilità del database, ma l'istanza DB non utilizza pagine di grandi dimensioni. Ti consigliamo di impostare il valore del `use_large_pages` parametro su `ONLY` nel gruppo di parametri DB per la tua istanza DB.	Imposta il valore del `use_large_pages` parametro su `ONLY` nei tuoi gruppi di parametri DB.	Sì	Attivazione di HugePages per un'istanza RDS per Oracle

Type	Descrizione	Raccomandazione	Tempo di inattività richiest	Informazioni aggiuntive
`autovacuum` il parametro è disattivato	Il parametro autovacuum è disattivato per le istanze DB. La disattivazione dell'autovacuum aumenta il volume della tabella e dell'indice e influisce sulle prestazioni. Ti consigliamo di attivare l'autovacuum nei gruppi di parametri del database.	Attiva il parametro autovacuum nei gruppi di parametri DB.	No	Informazioni sull'autovacuum negli ambienti Amazon RDS for PostgreSQL nel Database Blog AWS
`synchronous_commit` il parametro è disattivato	Quando `synchronous_commit` il parametro è disattivato, i dati possono andare persi in caso di arresto anomalo del database. La durabilità del database è a rischio. Consigliamo di attivare il parametro `synchronous_commit` .	Attiva i `synchronous_commit` parametri nei gruppi di parametri del database.	Sì	Parametri PostgreSQL di Amazon Aurora: replica, sicurezza e registrazione nel blog del database AWS

Type	Descrizione	Raccomandazione	Tempo di inattività richiest	Informazioni aggiuntive
`track_counts` il parametro è disattivato	Quando il `track_counts` parametro è disattivato, il database non raccoglie le statistiche sull'attività del database. La funzione di autovacuum richiede che queste statistiche funzionino correttamente. Consigliamo di impostare il parametro `track_counts` su 1.	Imposta `track_counts` il parametro su`1`.	No	Statistiche di runtime per PostgreSQL
`enable_indexonlyscan` il parametro è disattivato	Il pianificatore o l'ottimizzatore delle query non possono utilizzare il tipo di piano di scansione basato solo sull'indice quando è disattivato. Si consiglia di impostare il valore del `enable_indexonlyscan` parametro su. 1	Imposta il valore del `enable_indexonlyscan` parametro su`1`.	No	Configurazione del metodo Planner per PostgreSQL

Type	Descrizione	Raccomandazione	Tempo di inattività richiest	Informazioni aggiuntive
`enable_indexscan` il parametro è disattivato	Il pianificatore o l'ottimizzatore delle query non possono utilizzare il tipo di piano di scansione dell'indice quando è disattivato. Si consiglia di impostare il `enable_indexscan` valore su. 1	Imposta il valore del `enable_indexscan` parametro su1.	No	Configurazione del metodo Planner per PostgreSQL
`innodb_flush_log_at_trx` il parametro è disattivato	Il valore del `innodb_flush_log_at_trx` parametro dell'istanza DB non è un valore sicuro. Questo parametro controlla la persistenza delle operazioni di commit su disco. Consigliamo di impostare il parametro `innodb_flush_log_at_trx` su 1.	Imposta il valore del `innodb_flush_log_at_trx` parametro su1.	No	Le migliori pratiche per la configurazione dei parametri prestazionali per Amazon RDS for MySQL sul database blog AWS

Type	Descrizione	Raccomandazione	Tempo di inattività richiest	Informazioni aggiuntive
`sync_binlog` il parametro è disattivato	La sincronizzazione del registro binario con il disco non viene applicata prima che i commit delle transazioni vengano riconosciuti nell'istanza DB. Si consiglia di impostare il valore del `sync_binlog` parametro su. 1	Imposta il valore del `sync_binlog` parametro su1.	No	Le migliori pratiche per la configurazione dei parametri di replica per Amazon RDS for MySQL sul database blog AWS

Type	Descrizione	Raccomandazione	Tempo di inattività richiest	Informazioni aggiuntive
`innodb_stats_persistent` il parametro è disattivato	L'istanza database non è configurata per memorizzare le statistiche InnoDB sul disco. Quando le statistiche non vengono archiviate, vengono ricalcolate ogni volta che l'istanza si riavvia e si accede alla tabella. Ciò porta a variazioni nel piano di esecuzione e delle query. Puoi modificare il valore di questo parametro globale a livello di tabella. Si consiglia di impostare il valore del `innodb_stats_persistent` parametro su`ON`.	Imposta il valore del `innodb_stats_persistent` parametro su`ON`.	No	Le migliori pratiche per la configurazione dei parametri prestazionali per Amazon RDS for MySQL sul database blog AWS

Type	Descrizione	Raccomandazione	Tempo di inattività richiest	Informazioni aggiuntive
`innodb_open_files` il parametro è basso	Il `innodb_open_files` parametro controlla il numero di file che InnoDB può aprire contemporaneamente. InnoDB apre tutti i file di log e di tablespace di sistema quando mysqld è in esecuzione. Il valore del numero massimo di file dell'istanza database che InnoDB può aprire contemporaneamente non è sufficiente. Consigliamo di impostare il parametro `innodb_open_files` almeno sul valore 65.	Imposta il `innodb_open_files` parametro su un valore minimo di. 65	Sì	InnoDB apre i file per MySQL

Type	Descrizione	Raccomandazione	Tempo di inattività richiesto	Informazioni aggiuntive
`max_user_connections` il parametro è basso	Il valore del numero massimo di connessioni simultanee per ogni account di database dell'istanza database non è sufficiente. Si consiglia di impostare il `max_user_connections` parametro su un numero maggiore di5.	Aumentate il valore del `max_user_connections` parametro portandolo a un numero maggiore di5.	Sì	Impostazione dei limiti delle risorse dell'account per MySQL
Le repliche di lettura sono aperte in modalità scrivibile	L'istanza DB ha una replica di lettura in modalità scrivibile, che consente gli aggiornamenti dai client. Ti consigliamo di impostare il `read_only` parametro su in `TrueIfReplica` modo che le repliche di lettura non siano in modalità scrivibile.	Imposta il valore del `read_only` parametro su. `TrueIfReplica`	No	Le migliori pratiche per la configurazione dei parametri di replica per Amazon RDS for MySQL sul database blog AWS

Type	Descrizione	Raccomandazione	Tempo di inattività richiest	Informazioni aggiuntive
`innodb_default_row_format` l'impostazione dei parametri non è sicura	L'istanza DB presenta un problema noto: una tabella creata in una versione di MySQL precedente alla 8.0.26 con `COMPACT` o `REDUNDANT` sarà inaccessibile e irrecuperabile quando l'`row_format` indice supera i 767 byte. Si `innodb_default_row_format` consiglia di impostare il valore del parametro su. `DYNAMIC`	Imposta il valore del `innodb_default_row_format` parametro su`DYNAMIC`.	No	Modifiche in MySQL 8.0.26

Type	Descrizione	Raccomandazione	Tempo di inattività richiest	Informazioni aggiuntive
`general_logging` il parametro è attivato	La registrazione generale è attivata per l'istanza DB. Questa impostazione è utile per la risoluzione dei problemi del database. Tuttavia, l'attivazione della registrazione generale aumenta la quantità di operazioni di I/O e lo spazio di archiviazione allocato, il che potrebbe causare conflitti e un peggioramento delle prestazioni. Verifica i tuoi requisiti per l'utilizzo generale della registrazione. Si consiglia di impostare il valore del `general_logging` parametro su`0`.	Verifica i tuoi requisiti per l'utilizzo generale della registrazione. Se non è obbligatorio, ti consigliamo di impostare il valore del `general_logging` parametro su`0`.	No	Panoramica dei registri di database RDS per MySQL

Type	Descrizione	Raccomandazione	Tempo di inattività richiest	Informazioni aggiuntive
L'istanza RDS non dispone di risorse sufficienti per la capacità di memoria di sistema	Si consiglia di ottimizzare le query in modo da utilizzare meno memoria o utilizzare un tipo di istanza DB con una maggiore quantità di memoria allocata. Quando la memoria dell'istanza sta esaurendo, le prestazioni del database ne risentono.	Utilizza un'istanza DB con una maggiore capacità di memoria	Sì	Scalabilità verticale e orizzontale dell'istanza Amazon RDS sul database Blog AWS Tipi di istanze Amazon RDS Prezzi di Amazon SQS
L'istanza RDS non dispone di risorse sufficienti per la capacità della CPU di sistema	Ti consigliamo di ottimizzare le query per utilizzare meno CPU o di modificare l'istanza DB per utilizzare una classe di istanze DB con vCPU con allocazione più elevata. Le prestazioni del database potrebbero diminuire quando un'istanza DB sta esaurendo la CPU.	Utilizza un'istanza DB con una maggiore capacità della CPU	Sì	Scalabilità verticale e orizzontale dell'istanza Amazon RDS sul database Blog AWS Tipi di istanze Amazon RDS Prezzi di Amazon SQS

Type	Descrizione	Raccomandazione	Tempo di inattività richiesto	Informazioni aggiuntive
Le risorse RDS non utilizzano correttamente il pool di connessioni	Ti consigliamo di abilitare Amazon RDS Proxy per raggruppare e condividere in modo efficiente le connessioni al database esistenti. Se stai già utilizzando un proxy per il tuo database, configuralo correttamente per migliorare il pool di connessioni e il bilanciamento del carico su più istanze DB. RDS Proxy può aiutare a ridurre il rischio di esaurimento della connessione e i tempi di inattività, migliorando al contempo la disponibilità e la scalabilità.	Abilita RDS Proxy o modifica la configurazione proxy esistente	No	Scalabilità verticale e orizzontale dell'istanza Amazon RDS sul database Blog AWS Utilizzo del proxy Amazon RDS Prezzi del proxy Amazon RDS

Type	Descrizione	Raccomandazione	Tempo di inattività richiest	Informazioni aggiuntive
Le istanze RDS stanno creando oggetti temporanei eccessivi	Ti consigliamo di ottimizzare il carico di lavoro per evitare la creazione di oggetti temporanei eccessivi o di passare a classi di istanze RDS che supportano letture ottimizzate. RDS Optimized Reads migliora le prestazioni del database per i carichi di lavoro che coinvolgono un gran numero di oggetti temporanei e/o oggetti temporanei di grandi dimensioni. Valuta il carico di lavoro per determinare se l'utilizzo di un'istanza con RDS Optimized Reads avvantaggia il carico di lavoro del database.	Utilizza un tipo di istanza DB con RDS Optimized Reads	Sì	Tipi di istanze Amazon RDS Miglioramento delle prestazioni delle query per RDS for MySQL con Amazon RDS Optimized Reads Miglioramento delle prestazioni delle query per RDS per MariaDB con Amazon RDS Optimized Reads Miglioramento delle prestazioni delle query per RDS per PostgreSQL con Amazon RDS Optimized Reads

Type	Descrizione	Raccomandazione	Tempo di inattività richiesto	Informazioni aggiuntive
Le istanze RDS sono sottodimensionate per la capacità IOPS del sistema	Ti consigliamo di utilizzare una classe di istanza con un limite di IOPS predefinito più elevato, poiché hai fornito più IOPS su Amazon EBS di quanti ne possa supportare la tua classe di istanza attuale. L'utilizzo di una classe di istanza con un limite di IOPS supportato inferiore agli IOPS di Amazon EBS forniti impedisce di sfruttare appieno il potenziale degli IOPS di Amazon EBS forniti.	Utilizza un tipo di istanza DB con limiti IOPS predefiniti più elevati	Sì	Tipi di istanze Amazon RDS Storage di istanze database Amazon RDS Caricamento database

Utilizzando la console Amazon RDS, puoi visualizzare i consigli di Amazon RDS per le tue risorse di database.

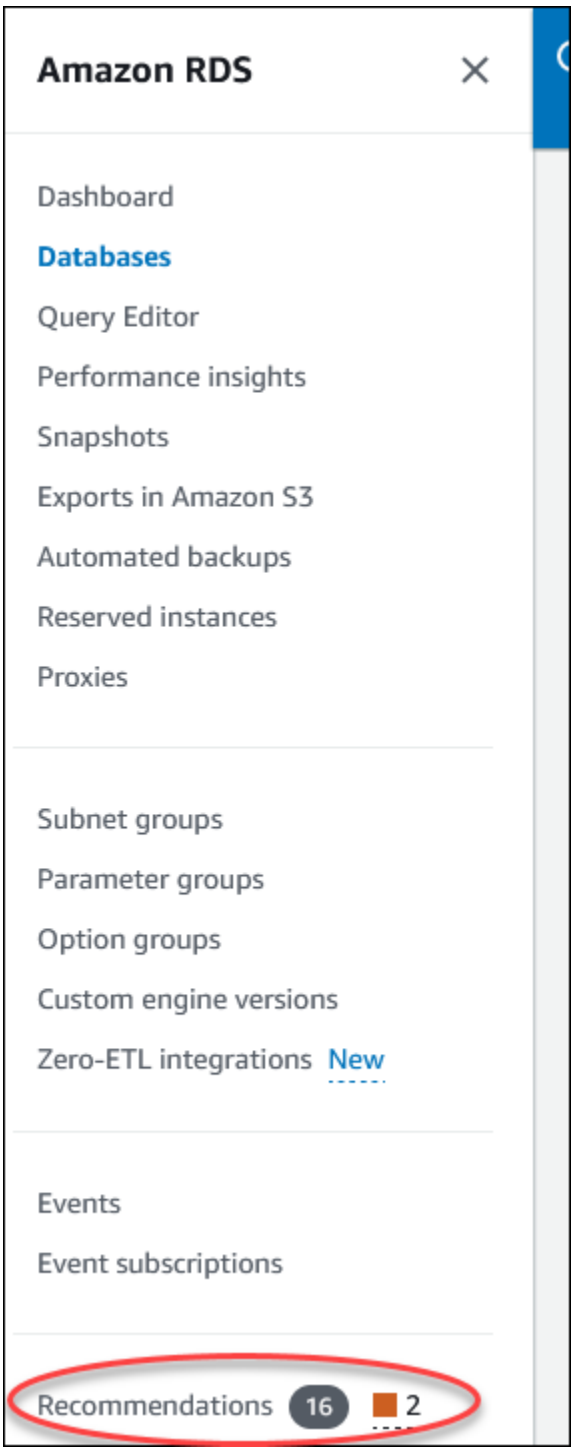
Console

Per visualizzare i consigli di Amazon RDS

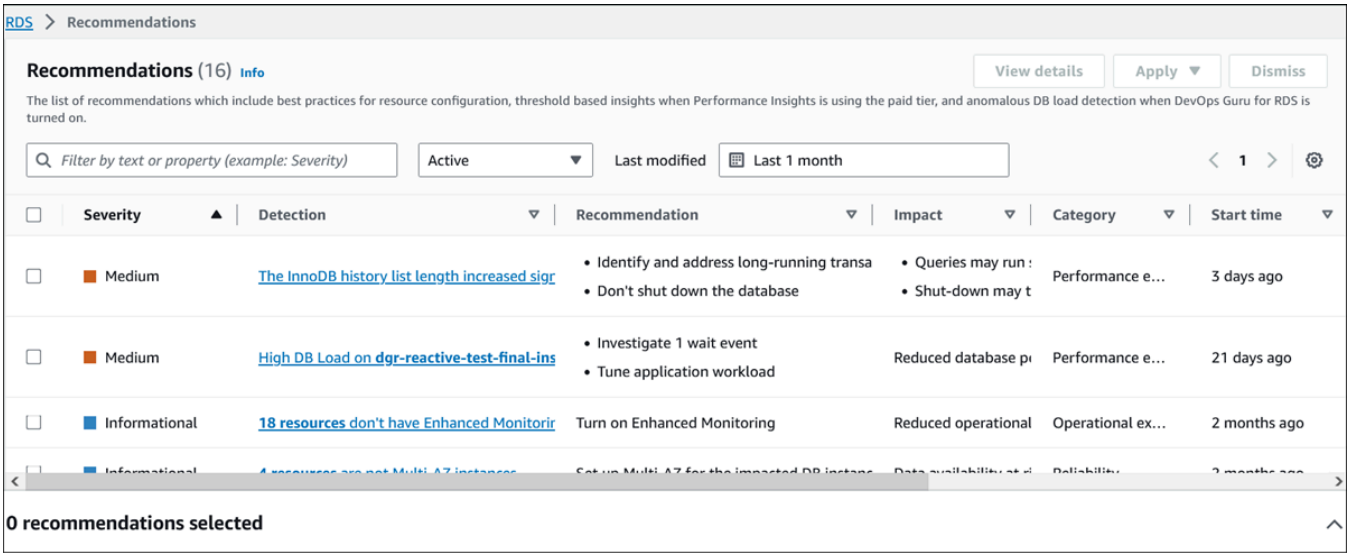
1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.

2. Nel pannello di navigazione, esegui una delle seguenti operazioni:

   - Scegli Consigli. Il numero di consigli attivi per le tue risorse e il numero di consigli con la massima severità generati nell'ultimo mese sono disponibili accanto a Consigli. Per trovare il numero di consigli attivi per ogni gravità, scegli il numero che mostra la gravità più alta.

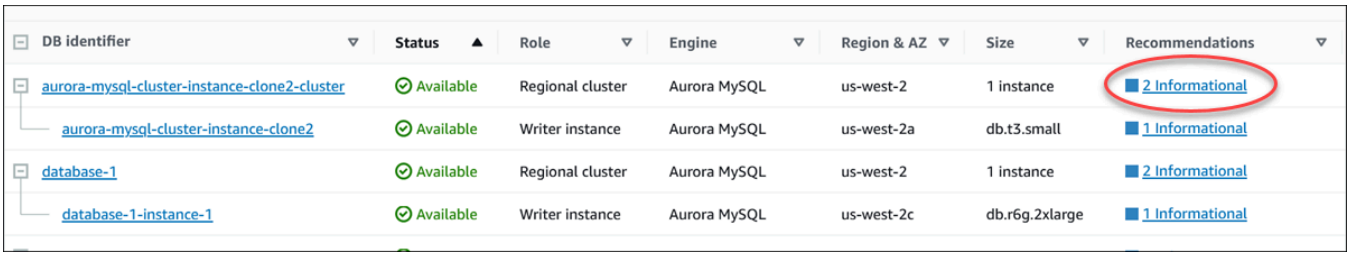


Per impostazione predefinita, la pagina Consigli mostra un elenco di nuovi consigli nell'ultimo mese. Amazon RDS Aurora fornisce consigli per tutte le risorse del tuo account e li ordina in base alla loro gravità.

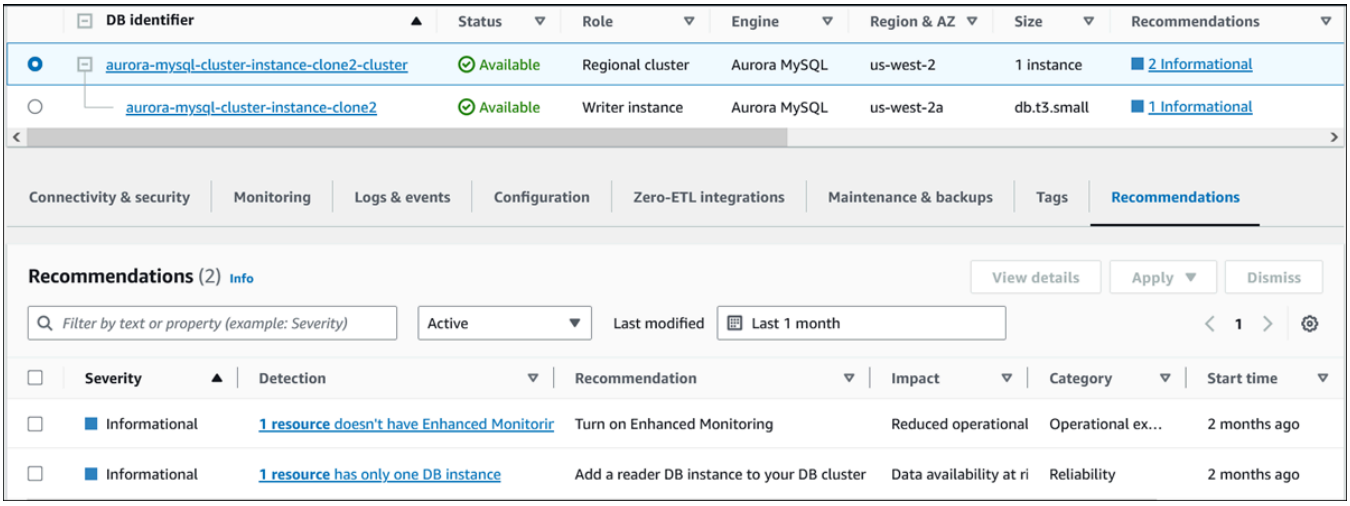


Puoi scegliere un consiglio per visualizzare una sezione nella parte inferiore della pagina che contiene le risorse interessate e i dettagli su come verrà applicata la raccomandazione.

- Nella pagina Database, scegli Consigli per una risorsa.



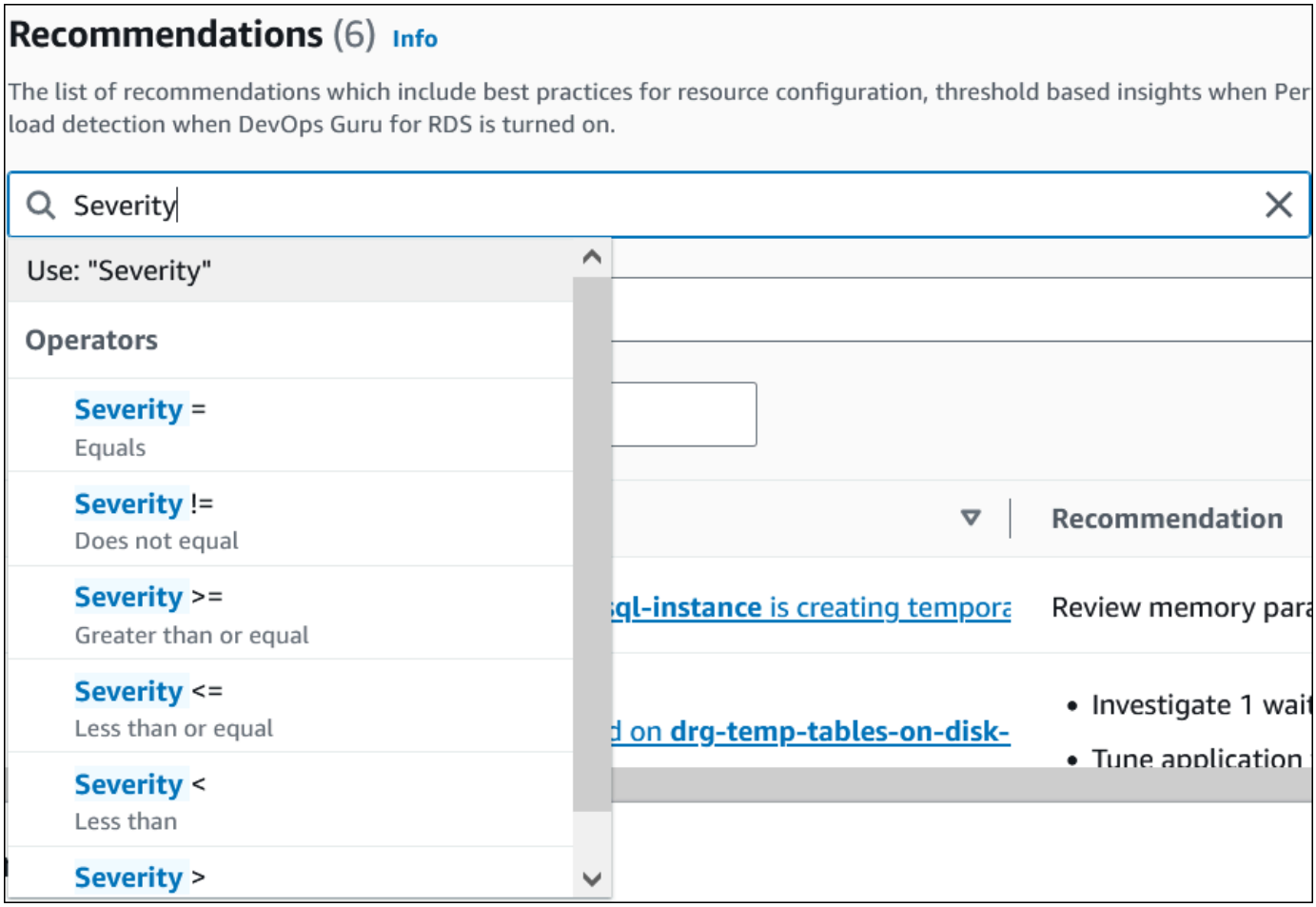
La scheda Consigli mostra i consigli e i relativi dettagli per la risorsa selezionata.



I seguenti dettagli sono disponibili per i consigli:

- Gravità: il livello di implicazione del problema. I livelli di gravità sono Alto, Medio, Basso e Informativo.
- Rilevamento: il numero di risorse interessate e una breve descrizione del problema. Scegli questo link per visualizzare la raccomandazione e i dettagli dell'analisi.
- Raccomandazione: una breve descrizione dell'azione consigliata da applicare.
- Impatto: una breve descrizione del possibile impatto quando la raccomandazione non viene applicata.
- Categoria: il tipo di raccomandazione. Le categorie sono Efficienza delle prestazioni, Sicurezza, Affidabilità, Ottimizzazione dei costi, Eccellenza operativa e Sostenibilità.
- Stato: lo stato attuale della raccomandazione. Gli stati possibili sono Tutti, Attivo, Ignorato, Risolto e In sospeso.
- Ora di inizio: l'ora in cui è iniziato il problema. Ad esempio, 18 ore fa.
- Ultima modifica: l'ora in cui il consiglio è stato aggiornato l'ultima volta dal sistema a causa di una modifica della severità o l'ora in cui hai risposto al consiglio. Ad esempio, 10 ore fa.
- Ora di fine: l'ora in cui il problema è terminato. L'ora non verrà visualizzata per eventuali problemi persistenti.
- Identificatore di risorsa: il nome di una o più risorse.

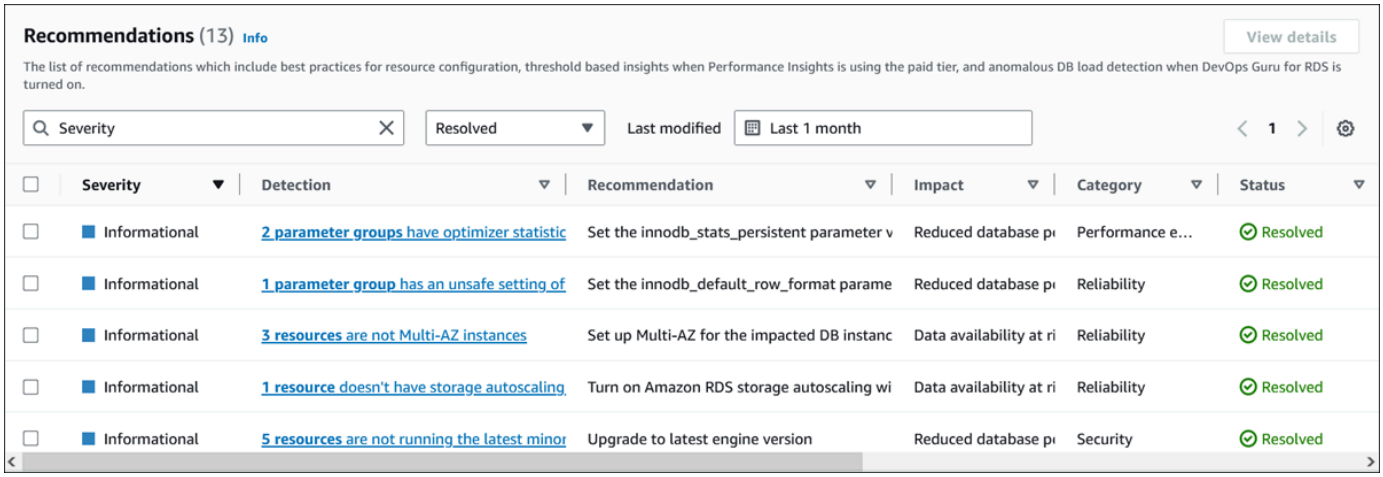
3. (Facoltativo) Scegliete gli operatori di severità o categoria nel campo per filtrare l'elenco dei consigli.



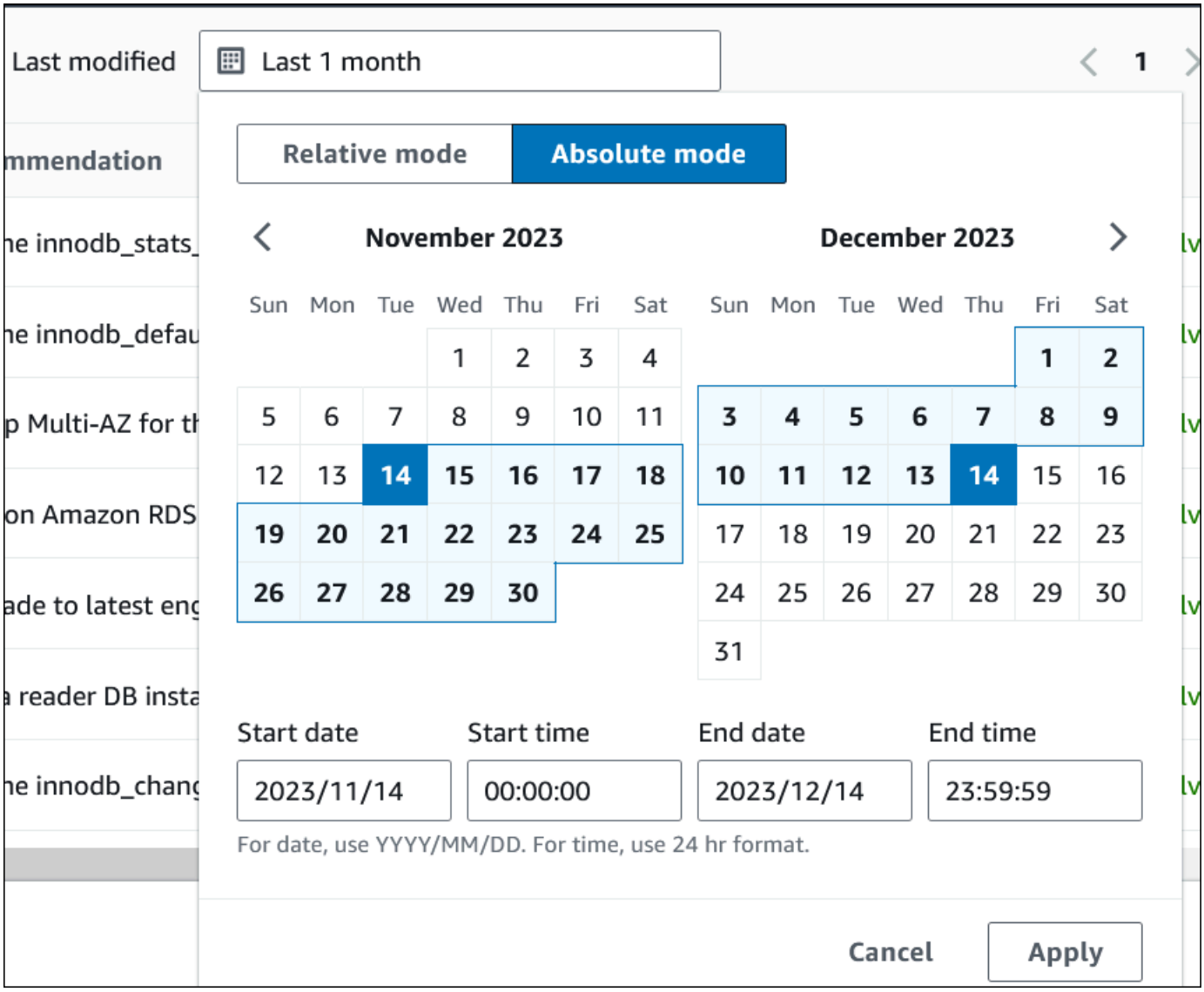
Vengono visualizzati i consigli per l'operazione selezionata.

4. (Facoltativo) Scegliete uno dei seguenti stati di raccomandazione:

   - Attivo (impostazione predefinita): mostra i consigli correnti che è possibile applicare, programmarli per la finestra di manutenzione successiva o ignorarli.
   - Tutti: mostra tutti i consigli con lo stato corrente.
   - Ignorato: mostra i consigli ignorati.
   - Risolto: mostra i consigli che sono stati risolti.
   - In sospeso: mostra i consigli le cui azioni consigliate sono in corso o pianificate per la finestra di manutenzione successiva.



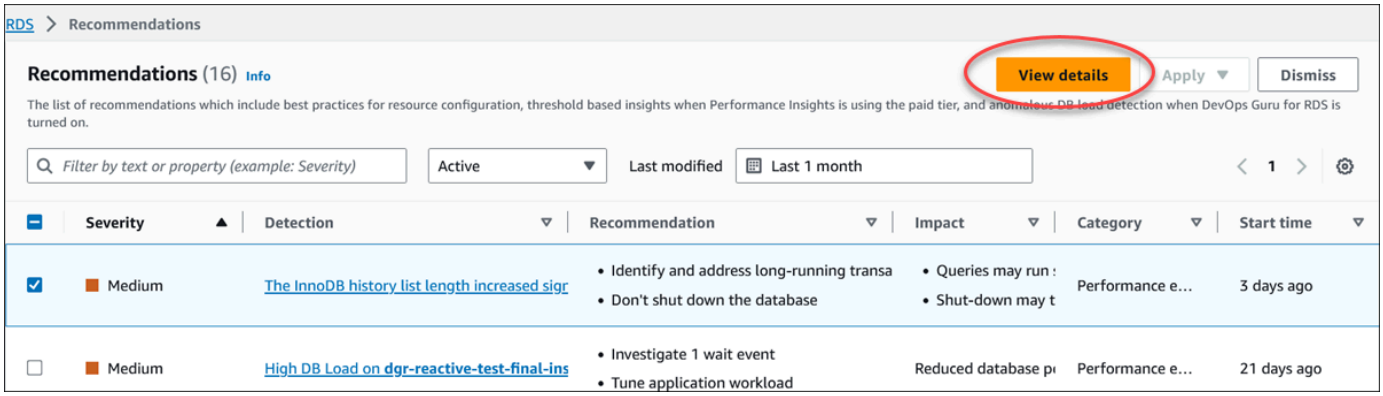
5. (Facoltativo) Scegliete la modalità relativa o la modalità Assoluta in Ultima modifica per modificare il periodo di tempo. La pagina Consigli mostra i consigli generati nel periodo di tempo. Il periodo di tempo predefinito è l'ultimo mese. In modalità Assoluta, puoi scegliere il periodo di tempo o inserire l'ora nei campi Data di inizio e Data di fine.



Vengono visualizzati i consigli per il periodo di tempo impostato.

Tieni presente che puoi visualizzare tutti i consigli relativi alle risorse del tuo account impostando l'intervallo su Tutti.

6. (Facoltativo) Scegli Preferenze a destra per personalizzare i dettagli da visualizzare. Puoi scegliere una dimensione di pagina, disporre le righe del testo e consentire o nascondere le colonne.
7. (Facoltativo) Scegli un consiglio, quindi scegli Visualizza dettagli.



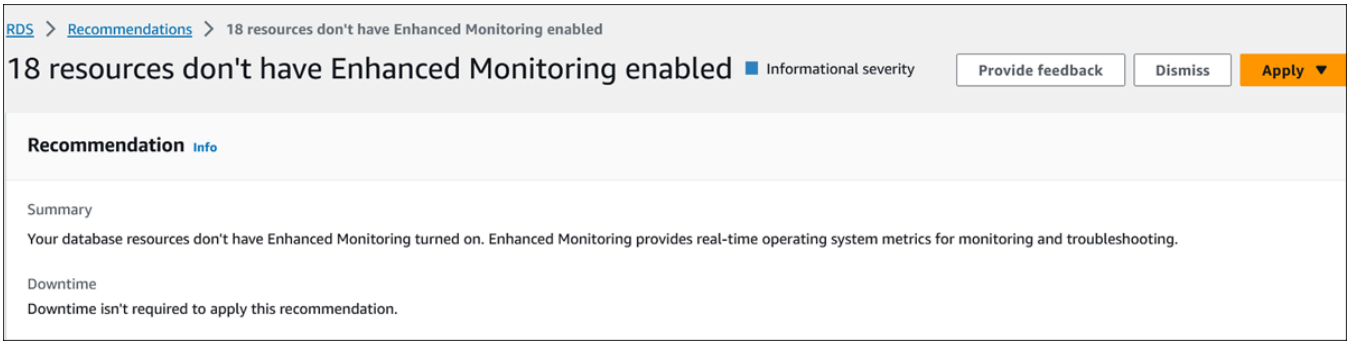
Viene visualizzata la pagina dei dettagli dei consigli. Il titolo fornisce il conteggio totale delle risorse con il problema rilevato e la gravità.

Per informazioni sui componenti nella pagina dei dettagli per una raccomandazione reattiva basata sulle anomalie, consulta la sezione Visualizzazione delle anomalie reattive nella Amazon DevOps Guru User Guide.

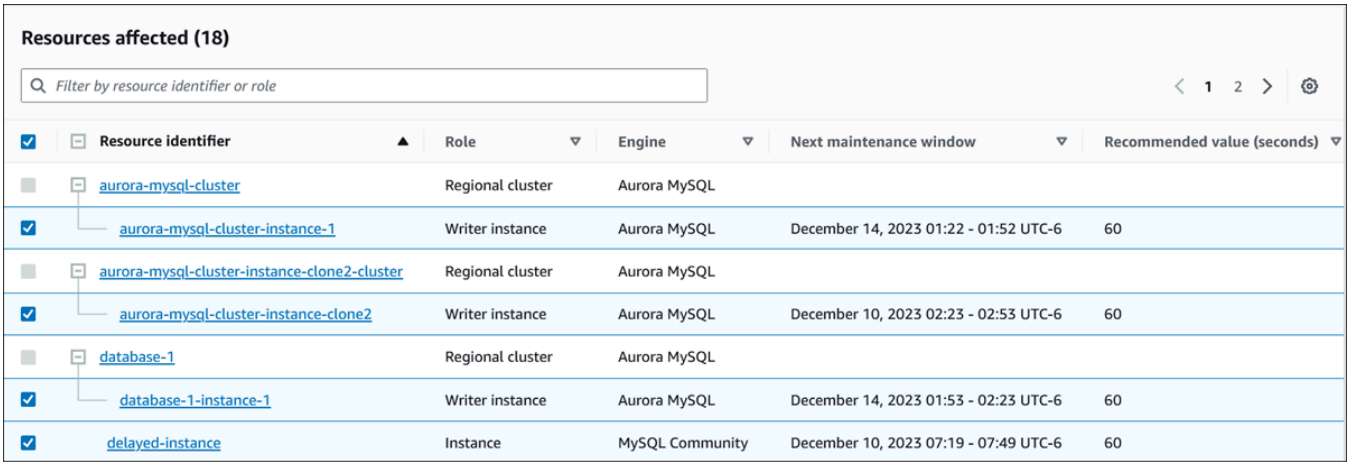
Per informazioni sui componenti nella pagina dei dettagli per una raccomandazione proattiva basata su una soglia, consulta. Visualizzazione dei consigli proattivi di Performance Insights

Gli altri consigli automatici mostrano i seguenti componenti nella pagina dei dettagli dei consigli:

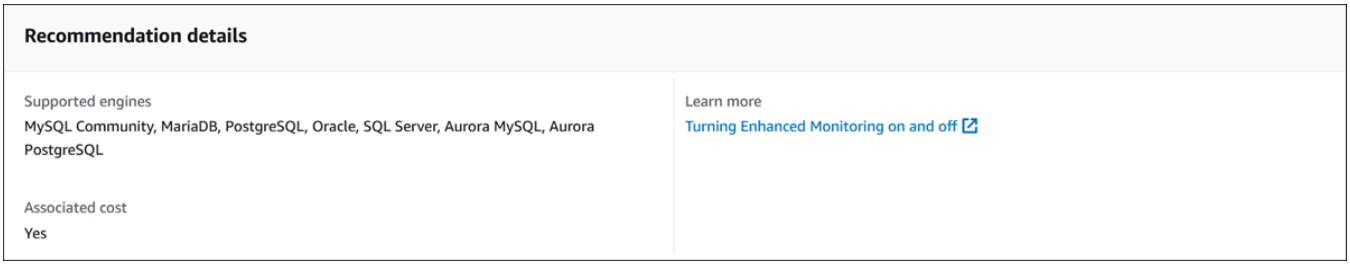
- Raccomandazione: un riepilogo della raccomandazione e indica se sono necessari tempi di inattività per applicarla.



- Risorse interessate: dettagli delle risorse interessate.



- Dettagli sui consigli: informazioni sul motore supportato, eventuali costi associati necessari per applicare il consiglio e link alla documentazione per saperne di più.



CLI

Per visualizzare le raccomandazioni di Amazon RDS relative alle istanze , usa il seguente comando in. AWS CLI

```
aws rds describe-db-recommendations
```

API RDS

Per visualizzare i consigli di Amazon RDS utilizzando l'API Amazon RDS, utilizza l'operazione DescribEdbRecommendations.

Risposta alle raccomandazioni Amazon RDS

Dall'elenco dei consigli di RDS , puoi:

- Applicare immediatamente un consiglio basato sulla configurazione o rimandarlo alla finestra di manutenzione successiva.
- Ignora uno o più consigli.

- Sposta uno o più consigli ignorati in consigli attivi.

Applicazione di un consiglio Amazon RDS

Utilizzando la console Amazon RDS, seleziona un consiglio basato sulla configurazione o una risorsa interessata nella pagina dei dettagli e applica immediatamente il consiglio o pianificalo per la finestra di manutenzione successiva. Potrebbe essere necessario riavviare la risorsa per rendere effettiva la modifica. Per alcuni consigli sui gruppi di parametri DB, potrebbe essere necessario riavviare le risorse.

I consigli proattivi basati sulla soglia o reattivi basati sulle anomalie non avranno l'opzione Applica e potrebbero richiedere un'ulteriore revisione.

Console

Per applicare un consiglio basato sulla configurazione

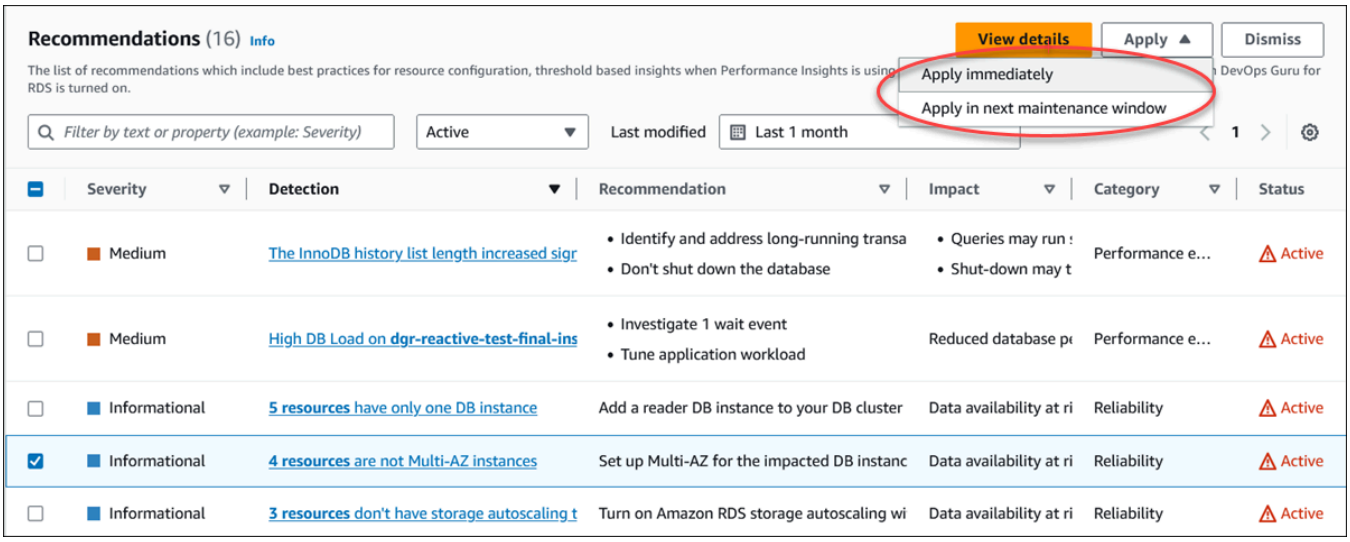
1. Accedi alla AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, effettuate una delle seguenti operazioni:
   - Scegli Consigli.

     Viene visualizzata la pagina Consigli con l'elenco di tutti i consigli.
   - Scegli Database, quindi scegli Consigli per una risorsa nella pagina dei database.

     I dettagli vengono visualizzati nella scheda Consigli per il consiglio selezionato.
   - Scegli Rilevamento per un consiglio attivo nella pagina Consigli o nella scheda Consigli nella pagina Database.

     Viene visualizzata la pagina dei dettagli dei consigli.
3. Scegli un consiglio o una o più risorse interessate nella pagina dei dettagli del consiglio ed esegui una delle seguenti operazioni:
   - Scegli Applica, quindi scegli Applica immediatamente per applicare immediatamente il consiglio.
   - Scegli Applica, quindi scegli Applica nella finestra di manutenzione successiva per programmarla nella finestra di manutenzione successiva.

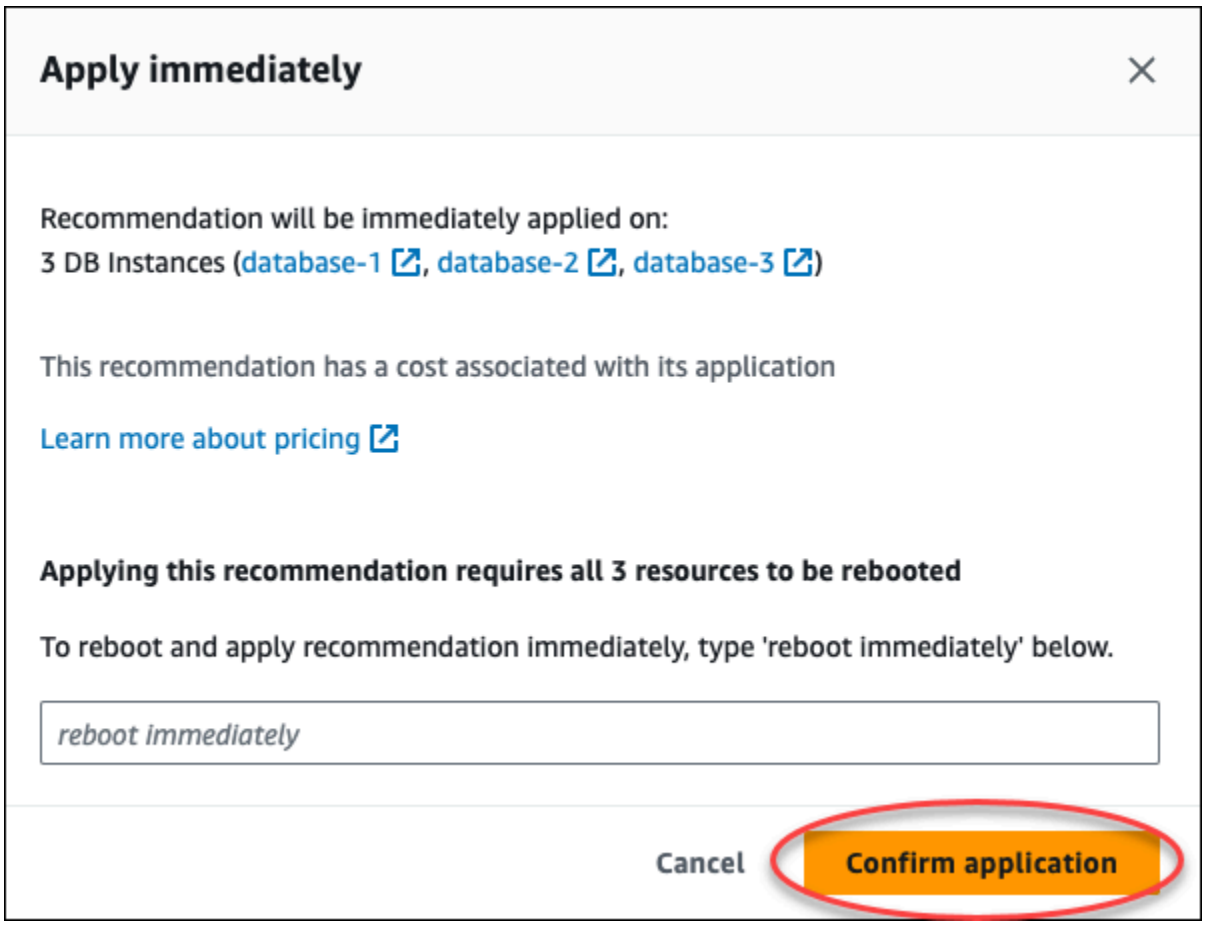
Lo stato del consiglio selezionato viene aggiornato e impostato in sospeso fino alla finestra di manutenzione successiva.



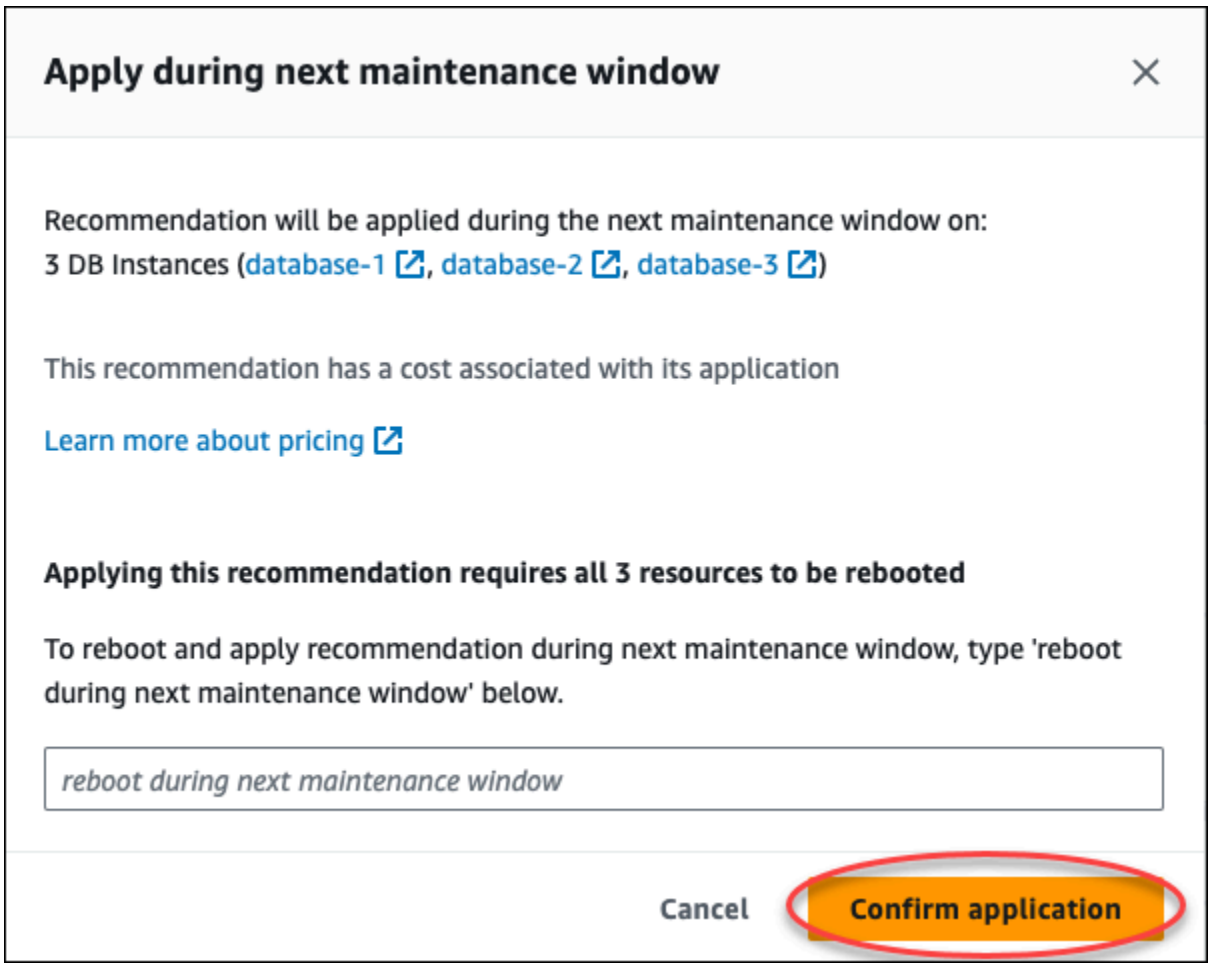
Viene visualizzata una finestra di conferma.

4. Scegli Conferma applicazione per applicare il consiglio. Questa finestra conferma se le risorse necessitano di un riavvio automatico o manuale per rendere effettive le modifiche.

   L'esempio seguente mostra la finestra di conferma per applicare immediatamente il consiglio.

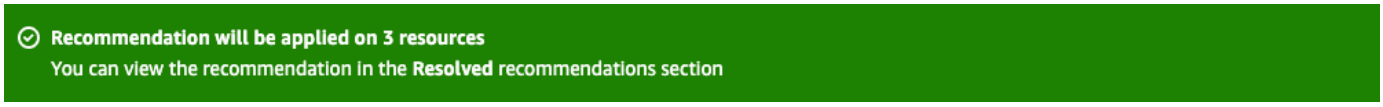


L'esempio seguente mostra la finestra di conferma per pianificare l'applicazione del consiglio nella finestra di manutenzione successiva.



Un banner mostra un messaggio quando il consiglio applicato ha esito positivo o negativo.

L'esempio seguente mostra il banner con il messaggio di successo.



L'esempio seguente mostra il banner con il messaggio di errore.

API RDS

Per applicare una raccomandazione RDS basata sulla configurazione utilizzando l'API Amazon RDS

1. Usa l'operazione DescribEdbRecommendations. `RecommendedActions`Nell'output possono essere presenti una o più azioni consigliate.
2. Usa l'RecommendedActionoggetto per ogni azione consigliata dal passaggio 1. L'output contiene `Operation` e`Parameters`.

   L'esempio seguente mostra l'output con un'azione consigliata.

```
   "RecommendedActions": [
       {
           "ActionId": "0b19ed15-840f-463c-a200-b10af1b552e3",
           "Title": "Turn on auto backup", // localized
           "Description": "Turn on auto backup for my-mysql-instance-1", //
localized
           "Operation": "ModifyDbInstance",
           "Parameters": [
               {
                   "Key": "DbInstanceIdentifier",
                   "Value": "my-mysql-instance-1"
               },
               {
                   "Key": "BackupRetentionPeriod",
                   "Value": "7"
               }
           ],
           "ApplyModes": ["immediately", "next-maintenance-window"],
           "Status": "applied"
       },
       ... // several others
   ],
```

3. Usa il `operation` per ogni azione consigliata dall'output nel passaggio 2 e inserisci i `Parameters` valori.
4. Una volta completata l'operazione nel passaggio 2, utilizzate l'operazione ModifyDBRecommendation per modificare lo stato del consiglio.

Ignorare i consigli di Amazon RDS per Amazon

Puoi ignorare uno o più consigli.

Console

Per ignorare uno o più consigli

1. Accedi alla AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, effettuate una delle seguenti operazioni:

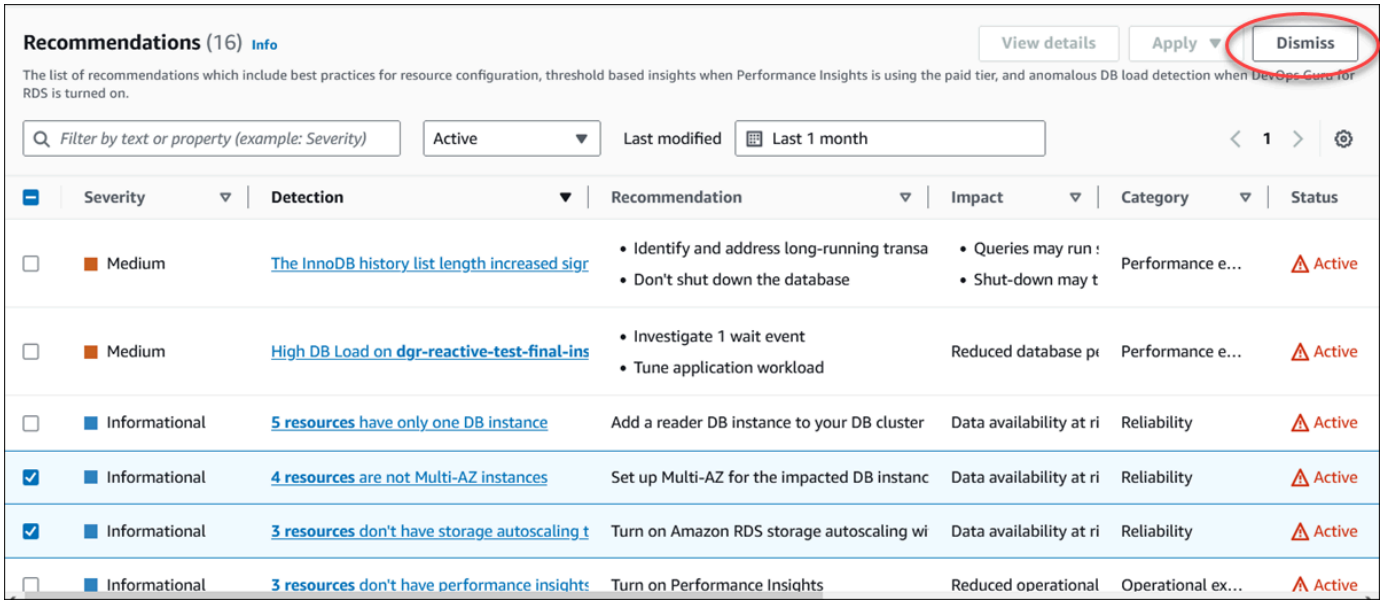
   - Scegli Consigli.

     Viene visualizzata la pagina Consigli con l'elenco di tutti i consigli.
   - Scegli Database, quindi scegli Consigli per una risorsa nella pagina dei database.

     I dettagli vengono visualizzati nella scheda Consigli per il consiglio selezionato.
   - Scegli Rilevamento per un consiglio attivo nella pagina Consigli o nella scheda Consigli nella pagina Database.

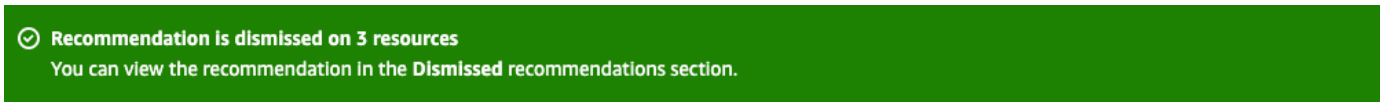
     La pagina dei dettagli dei consigli mostra l'elenco delle risorse interessate.
3. Scegli uno o più consigli o una o più risorse interessate nella pagina dei dettagli del consiglio, quindi scegli Ignora.

   L'esempio seguente mostra la pagina Consigli con più consigli attivi selezionati per essere ignorati.



Un banner mostra un messaggio quando uno o più consigli selezionati vengono ignorati.

L'esempio seguente mostra il banner con il messaggio di successo.



L'esempio seguente mostra il banner con il messaggio di errore.

CLI

Per ignorare un RDS una raccomandazione il AWS CLI

1. Esegui il comando `aws rds describe-db-recommendations --filters "Name=status,Values=active"`.

   L'output fornisce un elenco di raccomandazioni in corso. `active`
2. `recommendationId`Individua il consiglio che desideri eliminare dal passaggio 1.
3. Esegui il comando `>aws rds modify-db-recommendation --status dismissed --recommendationId <ID>` `recommendationId` dal passaggio 2 per ignorare il consiglio.

API RDS

Per ignorare una raccomandazione RDS di utilizzando l'API Amazon RDS, utilizza l'operazione ModifyDBRemendation.

Modifica delle raccomandazioni Amazon RDS Amazon Aurora ignorate in

Puoi spostare uno o più consigli ignorati in consigli attivi.

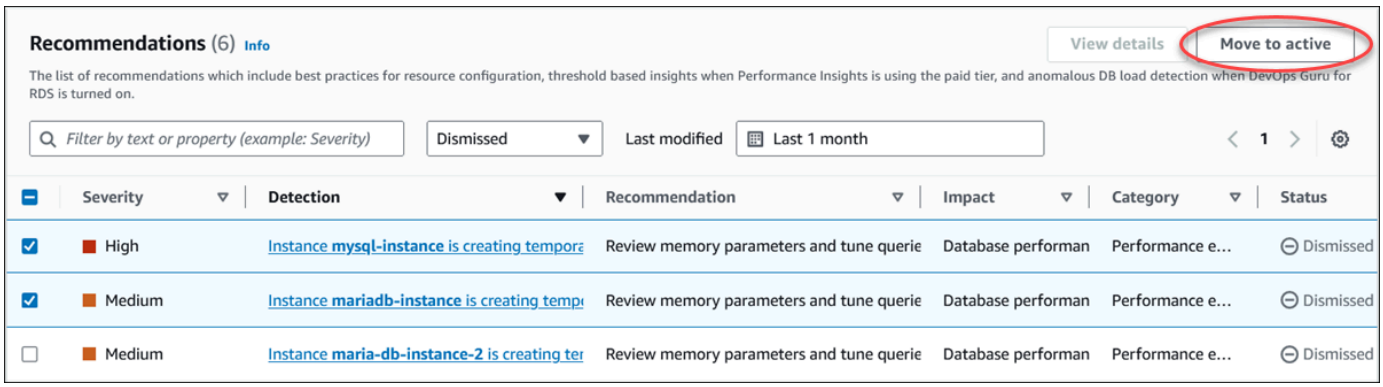
Console

Per spostare uno o più consigli ignorati in consigli attivi

1. Accedi alla AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, esegui una delle seguenti operazioni:
   - Scegli Consigli.

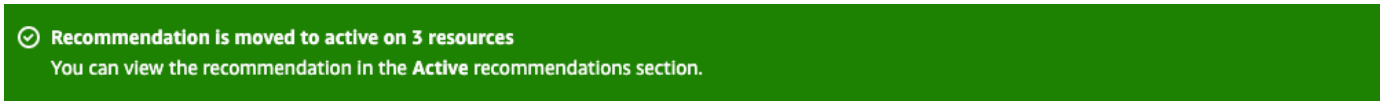
     La pagina Consigli mostra un elenco di consigli ordinati in base alla gravità per tutte le risorse del tuo account.
   - Scegli Database, quindi scegli Consigli per una risorsa nella pagina dei database.

     La scheda Consigli mostra i consigli e i relativi dettagli per la risorsa selezionata.
3. Scegli uno o più consigli ignorati dall'elenco, quindi scegli Sposta su attivo.



Un banner mostra un messaggio di successo o di fallimento quando si spostano i consigli selezionati dallo stato ignorato a quello attivo.

L'esempio seguente mostra il banner con il messaggio di successo.



L'esempio seguente mostra il banner con il messaggio di errore.

CLI

Per modificare una raccomandazione RDS Aurora utilizzando il AWS CLI

1. Esegui il comando `aws rds describe-db-recommendations --filters "Name=status,Values=dismissed"`.

   L'output fornisce un elenco di consigli in corso. `dismissed`
2. Trova `recommendationId` il suggerimento di cui desideri modificare lo stato dal passaggio 1.
3. Esegui `>aws rds modify-db-recommendation --status active --recommendationId <ID>` il comando `recommendationId` dal passaggio 2 per passare alla raccomandazione attiva.

API RDS

Per modificare una raccomandazione RDS ignorata in una raccomandazione attiva utilizzando l'API Amazon RDS, utilizza l'operazione ModifyDBRemendation.

Visualizzazione dei parametri nella console Amazon RDS

Amazon RDS si integra con Amazon CloudWatch per visualizzare una varietà di parametri per le istanze database RDS nella console RDS. Per le descrizioni di questi parametri, consulta Riferimento per i parametri per Amazon RDS.

Per l'istanza database, vengono monitorate le seguenti categorie di metriche:

- CloudWatch: mostra le metriche Amazon CloudWatch per RDS a cui è possibile accedere nella console RDS. Puoi visualizzare tali parametri anche nella console CloudWatch. Ogni parametro include un grafico che mostra il parametro monitorato in un periodo di tempo specifico. Per un elenco completo dei parametri CloudWatch, consulta CloudWatch Parametri Amazon per Amazon RDS.
- Enhanced monitoring (Monitoraggio avanzato): mostra un riepilogo dei parametri del sistema operativo quando l'istanza database RDS ha attivato il monitoraggio avanzato. RDS fornisce i parametri del monitoraggio avanzato al tuo account Amazon CloudWatch Logs. Ogni parametro del sistema operativo include un grafico che visualizza il parametro monitorato in un periodo di tempo specifico. Per una panoramica, consulta Monitoraggio dei parametri del sistema operativo con il monitoraggio avanzato. Per un elenco di parametri di monitoraggio avanzato, consulta Parametri del sistema operativo nel monitoraggio avanzato.
- OS Process list (Elenco processi sistema operativo): mostra i dettagli di ogni processo in esecuzione nell'istanza database.
- Performance Insights: apre il pannello di controllo di Amazon RDS Performance Insights per un'istanza database . Per una panoramica su Performance Insights, consulta Monitoraggio del carico DB con Performance Insights su Amazon RDS. Per un elenco di parametri di Performance Insights, consulta CloudWatch Metriche Amazon per Performance Insights.

Amazon RDS ora fornisce una visualizzazione consolidata delle metriche di Performance Insights e CloudWatch nel pannello di controllo di Performance Insights. Performance Insights deve essere attivato affinché l'istanza database possa utilizzare questa visualizzazione. È possibile scegliere la nuova visualizzazione di monitoraggio nella scheda Monitoraggio o Performance Insights nel pannello di navigazione. Per visualizzare le istruzioni per scegliere questa visualizzazione, consultare Visualizzazione delle metriche combinate nella console Amazon RDS.

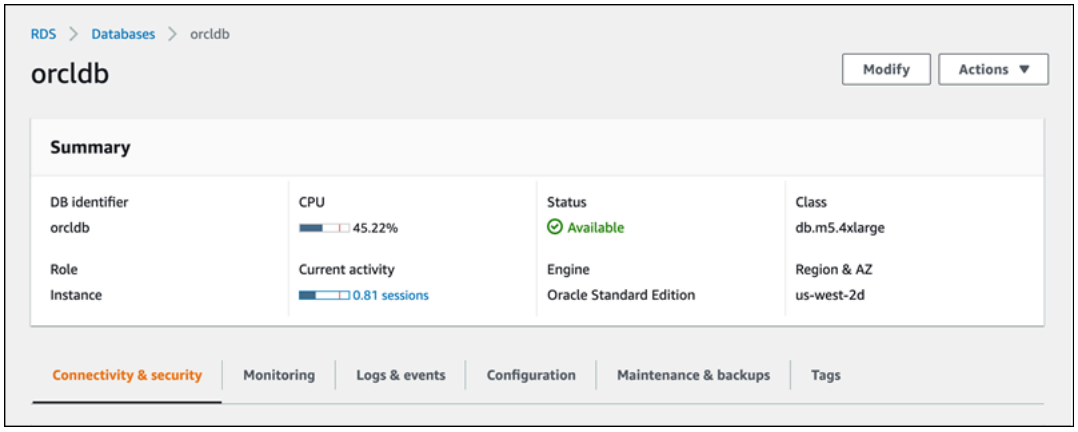
Per continuare a utilizzare la visualizzazione di monitoraggio legacy, continuare con questa procedura.

> **Note**
>
> La visualizzazione di monitoraggio legacy non sarà più disponibile a partire dal 15 dicembre 2023.

Per visualizzare le metriche per l'istanza database nella visualizzazione di monitoraggio legacy:

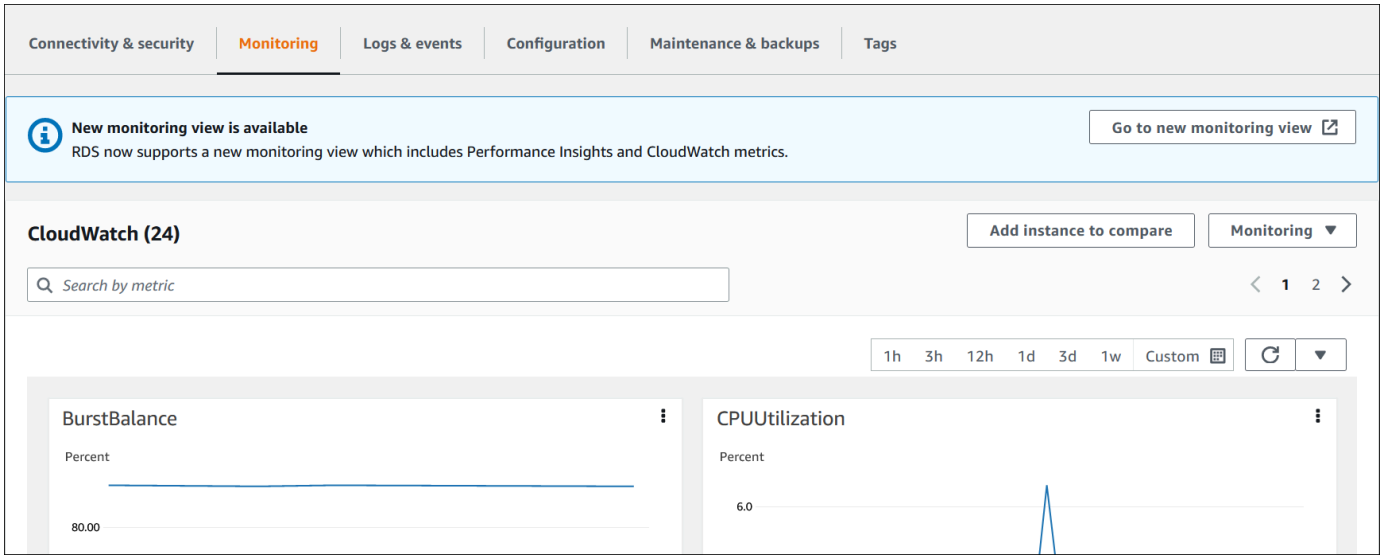
1. Accedi alla AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel pannello di navigazione, scegliere Databases (Database).
3. Scegliere il nome del di database che si desidera monitorare.

   Verrà visualizzata la pagina Databases (Database). L'esempio seguente mostra un database Oracle denominato `orclb`.

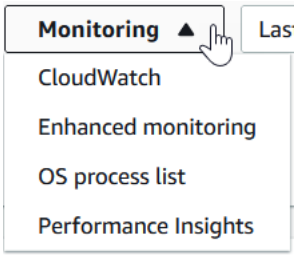


4. Scorri verso il basso e seleziona Monitoring (Monitoraggio).

   Viene visualizzata la sezione di monitoraggio. Di default, sono visualizzati tutti i parametri CloudWatch. Per una descrizione di questi parametri, consulta CloudWatch Parametri Amazon per Amazon RDS.



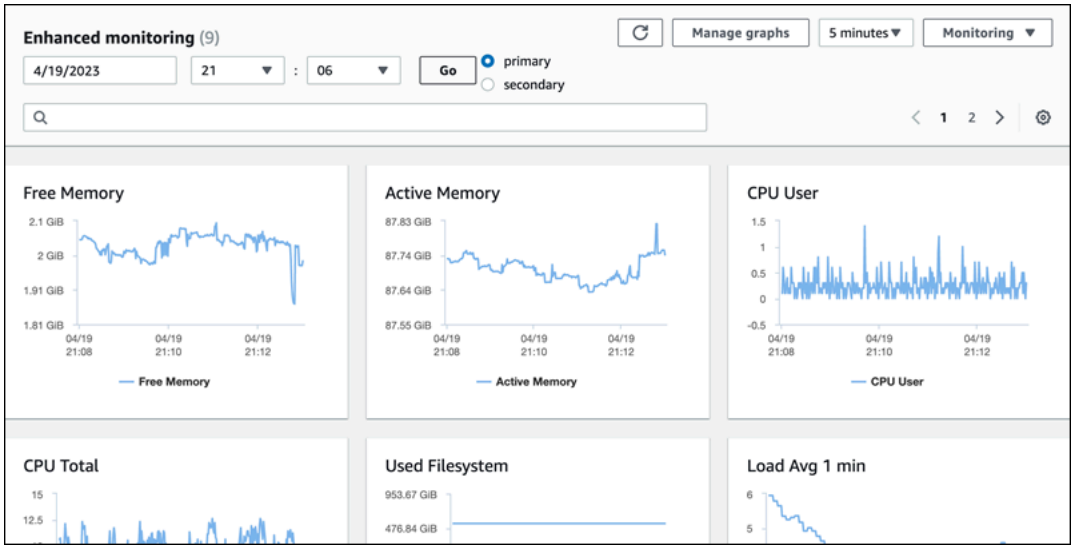
5. Scegli Monitoring (Monitoraggio) per vedere le categorie dei parametri.



6. Scegli la categoria di parametri da visualizzare.

   L'esempio seguente mostra i parametri di monitoraggio avanzato. Per una descrizione di questi parametri, consulta Parametri del sistema operativo nel monitoraggio avanzato.

> **Note**
>
> Attualmente, la visualizzazione dei parametri del sistema operativo per una replica di standby Multi-AZ non è supportata per le istanze database MariaDB.



> **Tip**
>
> Per scegliere l'intervallo di tempo dei parametri rappresentati dai grafici, puoi utilizzare l'elenco di intervalli di tempo.
>
> Puoi selezionare un grafico per ottenere una visualizzazione più dettagliata. Puoi anche applicare filtri specifici per i parametri ai dati.

Visualizzazione delle metriche combinate nella console Amazon RDS

Amazon RDS ora fornisce una visualizzazione consolidata delle metriche di Performance Insights e CloudWatch per l'istanza database nel pannello di controllo di Performance Insights. È possibile utilizzare il pannello di controllo preconfigurato o crearne uno personalizzato. Il pannello di controllo preconfigurato fornisce le metriche più comunemente utilizzate per diagnosticare i problemi relativi alle prestazioni di un motore di database. In alternativa, è possibile creare un pannello di controllo personalizzato contenente le metriche di un motore di database che soddisfi i requisiti di analisi definiti. Sarà quindi possibile utilizzare questo pannello di controllo per tutte le istanze database di tale tipo di motore di database nell'account AWS in uso.

È possibile scegliere la nuova visualizzazione di monitoraggio nella scheda Monitoraggio o Performance Insights nel pannello di navigazione. Quando si accede alla pagina Performance Insights, vengono visualizzate le opzioni per scegliere tra la nuova visualizzazione di monitoraggio e la visualizzazione legacy. L'opzione scelta viene salvata come visualizzazione predefinita.

Performance Insights deve essere attivato affinché l'istanza database possa visualizzare le metriche combinate nel pannello di controllo di Performance Insights. Per ulteriori informazioni sull'attivazione di Performance Insights, consultare Attivazione e disattivazione di Performance Insights.

> **Note**
>
> Si consiglia di scegliere la nuova visualizzazione di monitoraggio. È possibile continuare a utilizzare la visualizzazione di monitoraggio legacy fino alla sua dismissione in data 15 dicembre 2023.

Scelta della nuova visualizzazione di monitoraggio nella scheda Monitoraggio

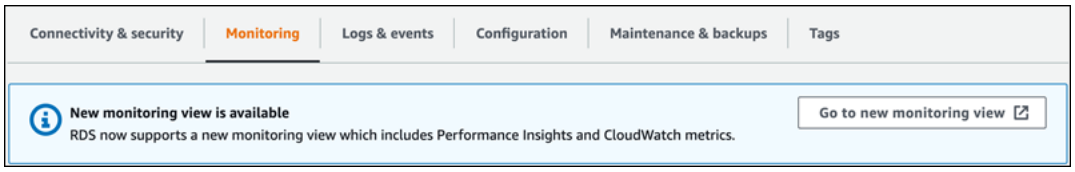
Per scegliere la nuova visualizzazione di monitoraggio nella scheda Monitoraggio:

1. Accedi alla AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel pannello di navigazione a sinistra, scegliere Database.
3. Scegliere l'istanza database che si desidera monitorare.

Verrà visualizzata la pagina Databases (Database).

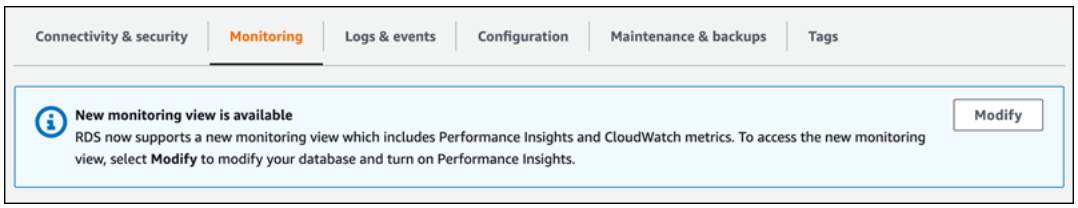
4. Scorrere verso il basso e selezionare Monitoraggio.

   Viene visualizzato un banner con l'opzione che consente di scegliere la nuova visualizzazione di monitoraggio. Nell'esempio seguente è rappresentato il banner per scegliere la nuova visualizzazione di monitoraggio.



5. Scegliere Vai alla nuova visualizzazione di monitoraggio per aprire il pannello di controllo di Performance Insights con le metriche di Performance Insights e CloudWatch per l'istanza database.
6. (Facoltativo) Se Performance Insights è disattivato per l'istanza database, viene visualizzato un banner con l'opzione per modificare il cluster database e attivare Performance Insights.

   L'esempio seguente mostra il banner per modificare il cluster database nella scheda Monitoraggio.



   Scegliere Modifica per modificare il cluster database e attivare Performance Insights. Per ulteriori informazioni sull'attivazione di Performance Insights, consultare Attivazione e disattivazione di Performance Insights.

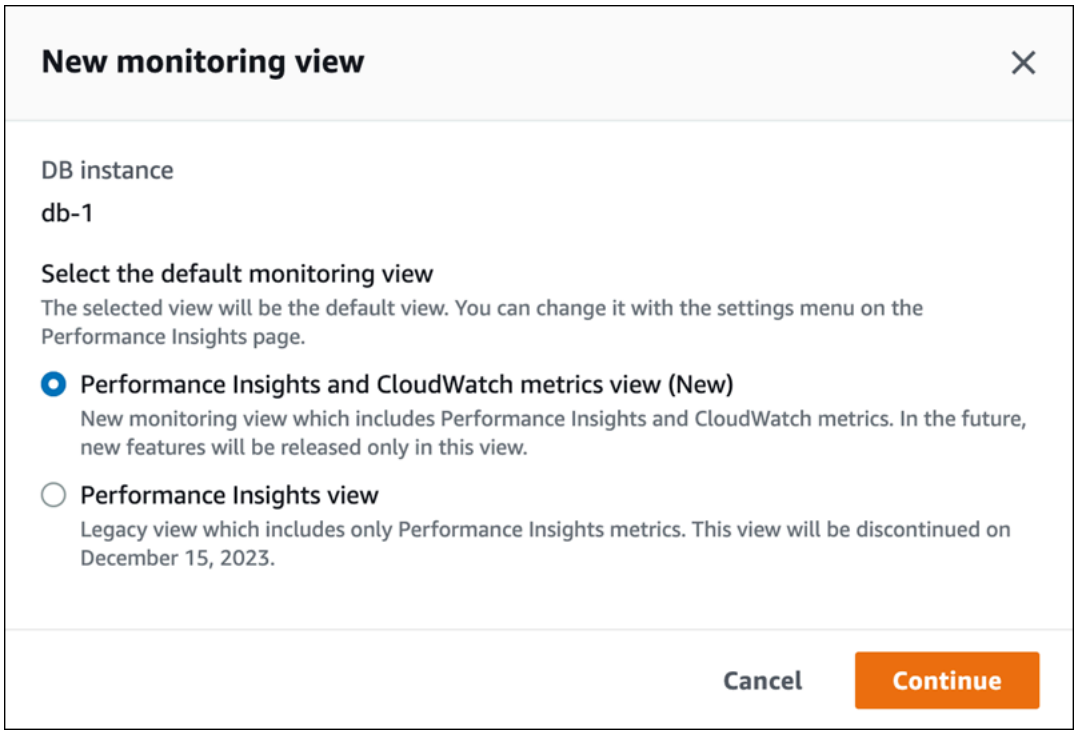
Scelta della nuova visualizzazione di monitoraggio con Performance Insights nel pannello di navigazione

Per scegliere la nuova visualizzazione di monitoraggio con Performance Insights nel pannello di navigazione:

1. Accedi alla AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.

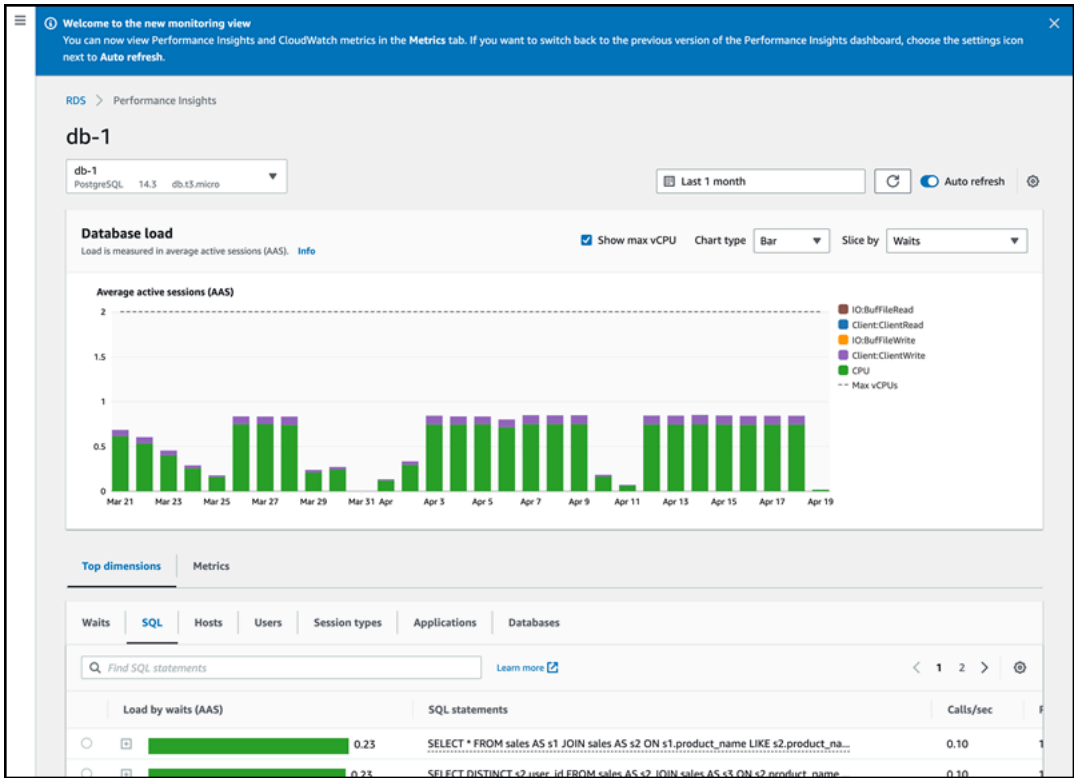
2. Nel pannello di navigazione scegli Performance Insights.
3. Scegliere un'istanza database per aprire una finestra con le opzioni per la visualizzazione del monitoraggio.

   Nell'esempio seguente viene illustrata la finestra con le opzioni per la visualizzazione del monitoraggio.



4. Selezionare l'opzione Visualizzazione metriche di Performance Insights e CloudWatch (Nuova) e scegliere Continua.

   Ora è possibile visualizzare il pannello di controllo di Performance Insights in cui sono visualizzate le metriche sia di Performance Insights che quelle di CloudWatch per l'istanza database. L'esempio seguente mostra i parametri di Performance Insights e CloudWatch nel pannello di controllo.



Scelta della visualizzazione legacy con Performance Insights nel pannello di navigazione

È possibile scegliere la visualizzazione di monitoraggio legacy per visualizzare solo le metriche di Performance Insights per un'istanza database specifica.

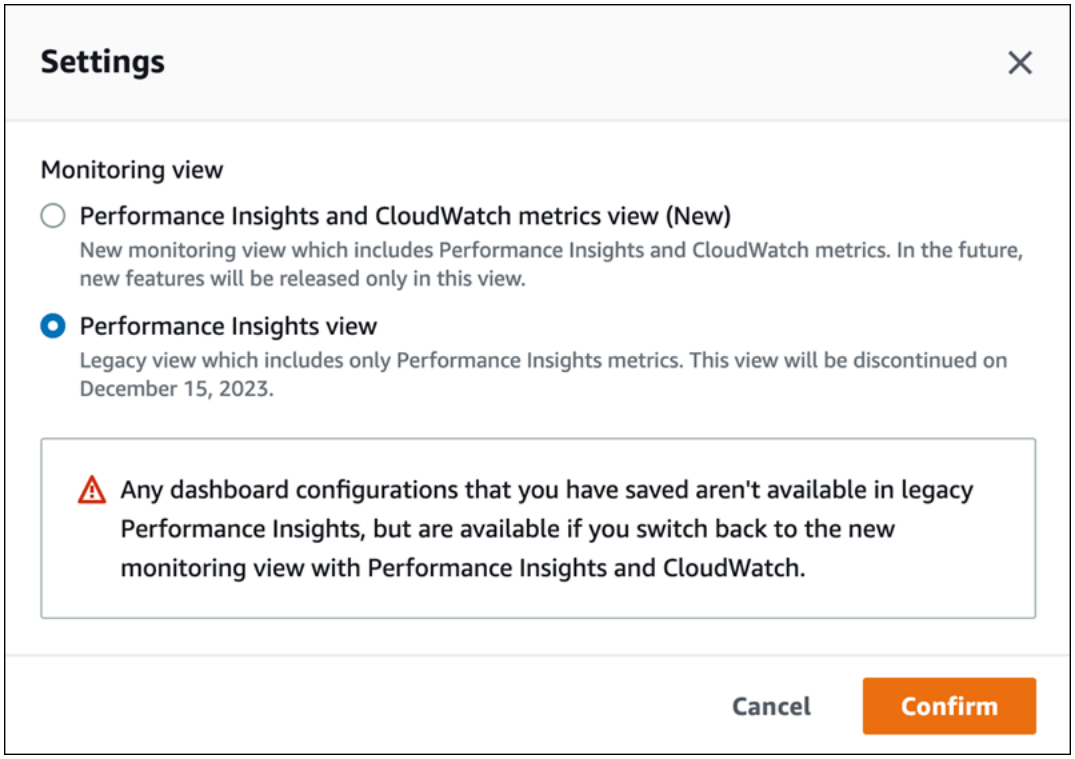
> **Note**
>
> Questa visualizzazione non sarà più disponibile a partire dal 15 dicembre 2023.

Per scegliere la visualizzazione di monitoraggio legacy con Performance Insights nel pannello di navigazione:

1. Accedi alla AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel pannello di navigazione scegli Performance Insights.
3. Scegli un'istanza database.
4. Scegliere l'icona delle impostazioni nel pannello di controllo di Performance Insights.

Ora è possibile visualizzare la finestra Impostazioni che mostra l'opzione per scegliere la visualizzazione legacy di Performance Insights.

Nell'esempio seguente viene illustrata la finestra con l'opzione per la visualizzazione del monitoraggio legacy.



5. Selezionare l'opzione Visualizzazione Performance Insights e scegliere Continua.

   Viene visualizzato un avviso. Eventuali configurazioni del pannello di controllo precedentemente salvate non saranno disponibili in questa visualizzazione.

6. Scegliere Conferma per passare alla visualizzazione legacy di Performance Insights.

   Ora è possibile visualizzare il pannello di controllo di Performance Insights in cui sono visualizzate solo le metriche di Performance Insights per l'istanza database.

Creazione di un pannello di controllo personalizzato con Performance Insights nel pannello di navigazione

Nella nuova visualizzazione di monitoraggio, è possibile creare un pannello di controllo personalizzato con le metriche necessarie per soddisfare gli specifici requisiti di analisi.

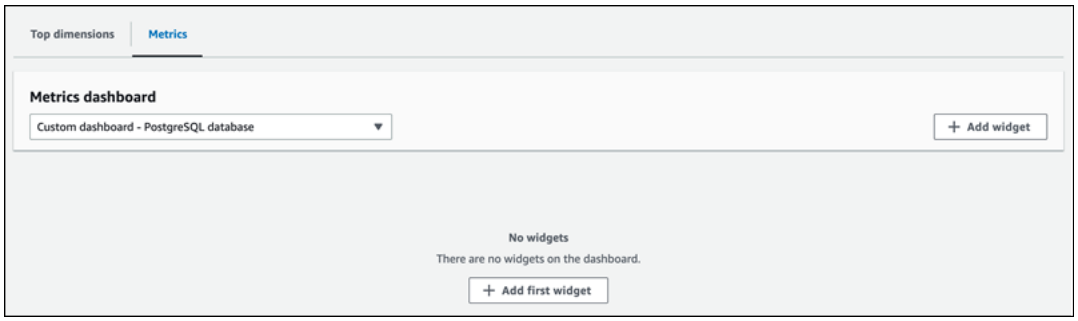
È possibile creare un pannello di controllo personalizzato selezionando Performance Insights e le metriche di CloudWatch per l'istanza database specifica. È possibile utilizzare questo pannello di controllo personalizzato per altre istanze database dello stesso tipo di motore di database nell'account AWS in uso.

> **Note**
>
> Il pannello di controllo personalizzato supporta fino a 50 metriche.

Usare il menu delle impostazioni del widget per modificare o eliminare il pannello di controllo e spostare o ridimensionare la finestra del widget.

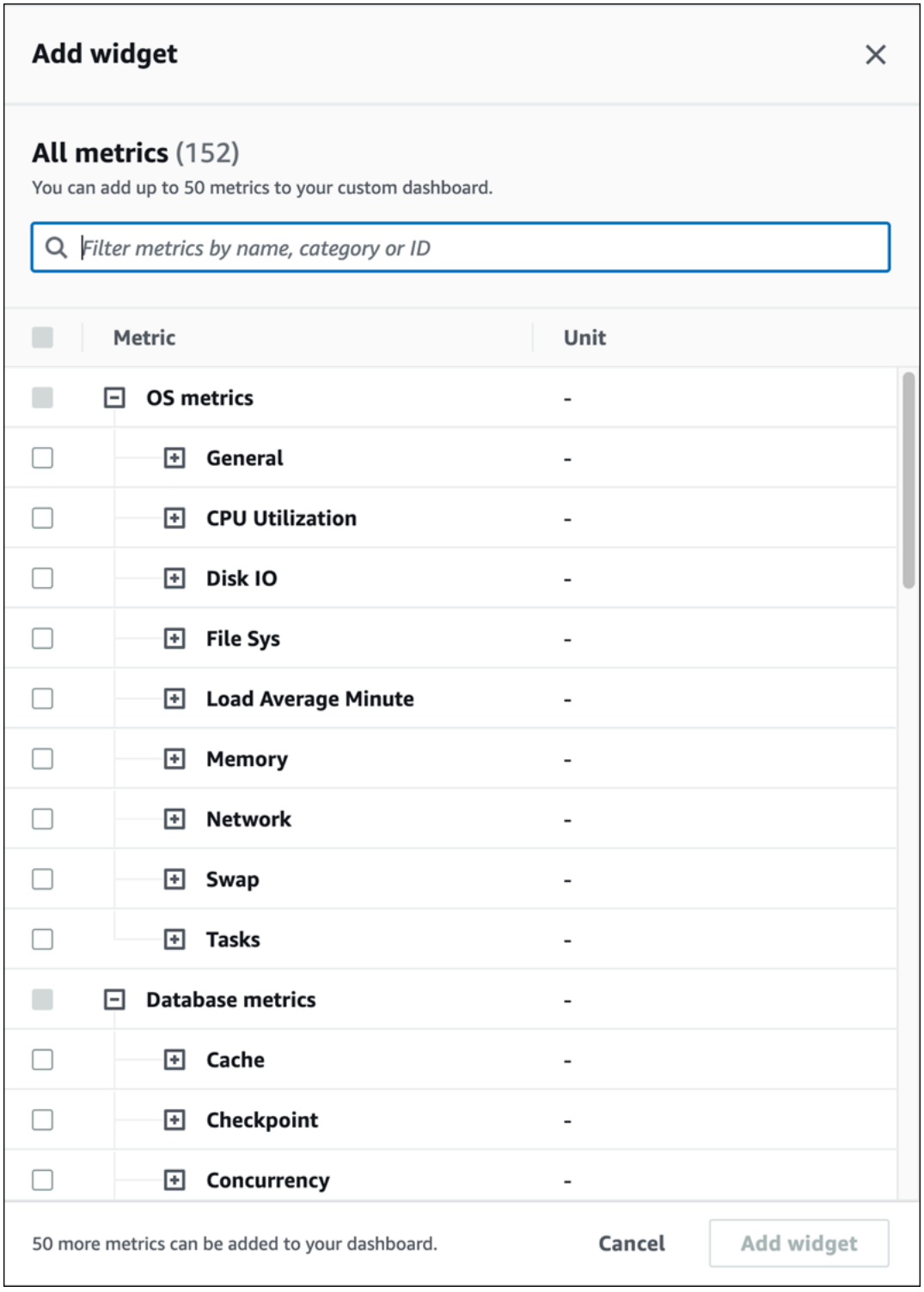
Per creare un pannello di controllo personalizzato con Performance Insights nel pannello di navigazione:

1. Accedi alla AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel pannello di navigazione scegli Performance Insights.
3. Scegli un'istanza database.
4. Scorrere verso il basso fino alla scheda Metriche nella finestra.
5. Selezionare il pannello di controllo personalizzato nell'elenco a discesa. Nell'esempio seguente viene illustrata la creazione del pannello di controllo personalizzato.



6. Scegli Aggiungi widget per aprire la finestra Aggiungi widget. Nella finestra è possibile aprire e visualizzare le metriche disponibili per il sistema operativo (SO), per il database e per CloudWatch.

   L'esempio seguente mostra la finestra Aggiungi widget con le metriche.



7. Selezionare le metriche da visualizzare nel pannello di controllo e scegliere Aggiungi widget. È possibile utilizzare il campo di ricerca per cercare una metrica specifica.

   Le metriche selezionate vengono visualizzate nel pannello di controllo.

8. (Facoltativo) Per modificare o eliminare il pannello di controllo, scegliere l'icona delle impostazioni in alto a destra del widget, quindi selezionare una delle seguenti azioni nel menu.
   - Modifica: consente di modificare l'elenco delle metriche nella finestra. Scegliere Aggiorna widget dopo aver selezionato le metriche da visualizzare nel pannello di controllo.
   - Elimina: consente di eliminare il widget. Nella finestra di conferma scegliere Elimina.

Scelta del pannello di controllo preconfigurato con Performance Insights nel pannello di navigazione

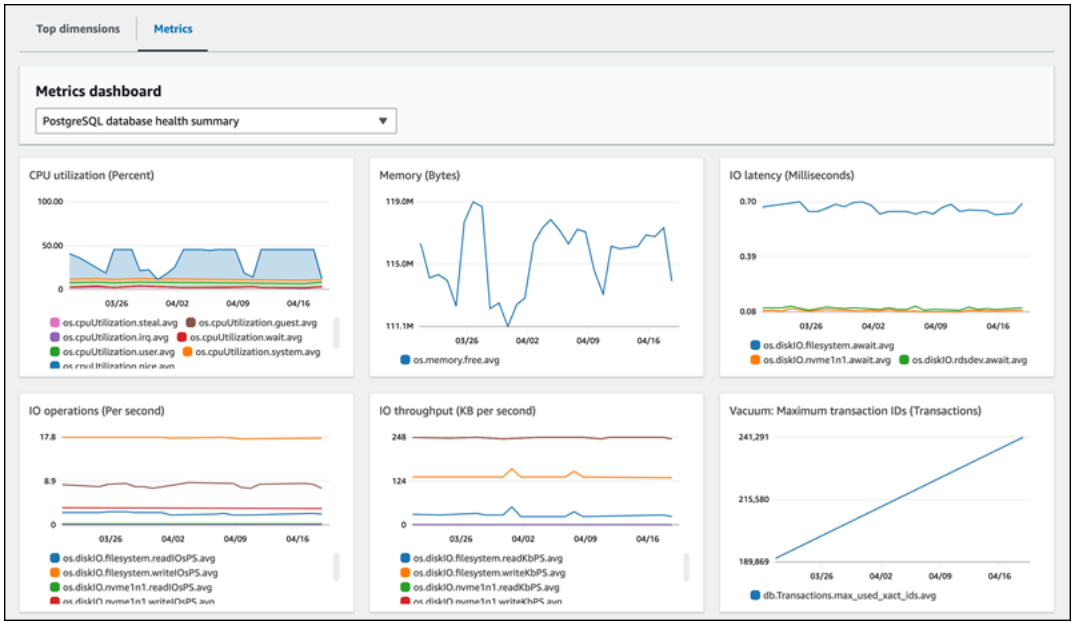
Nel pannello di controllo preconfigurato è possibile visualizzare le metriche di più frequente utilizzo. Questo pannello di controllo aiuta a diagnosticare i problemi di prestazioni di un motore di database e a ridurre il tempo medio di ripristino da ore a minuti.

> **Note**
>
> Questo pannello di controllo non può essere modificato.

Per scegliere il pannello di controllo preconfigurato con Performance Insights nel riquadro di navigazione:

1. Accedi alla AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel pannello di navigazione scegli Performance Insights.
3. Scegli un'istanza database.
4. Scorrere verso il basso fino alla scheda Metriche nella finestra.
5. Selezionare un pannello di controllo preconfigurato nell'elenco a discesa.

   Nel pannello di controllo è possibile visualizzare le metriche per l'istanza database. Nell'esempio seguente viene illustrato un pannello di controllo preconfigurato con le metriche.


Top dimensions
Metrics
Metrics dashboard
PostgreSQL database health summary
CPU utilization (Percent)
100.00
50.00
0
03/26
04/02
04/09
04/16
os.cpuUtilization.steal.avg
os.cpuUtilization.guest.avg
os.cpuUtilization.irq.avg
os.cpuUtilization.wait.avg
os.cpuUtilization.user.avg
os.cpuUtilization.system.avg
Memory (Bytes)
119.0M
115.0M
111.1M
os.memory.free.avg
IO latency (Milliseconds)
0.70
0.39
0.08
os.diskIO.filesystem.await.avg
os.diskIO.nvme1n1.await.avg
os.diskIO.rdsdev.await.avg
IO operations (Per second)
17.8
8.9
os.diskIO.filesystem.readIOsPS.avg
os.diskIO.filesystem.writeIOsPS.avg
os.diskIO.nvme1n1.readIOsPS.avg
IO throughput (KB per second)
248
124
os.diskIO.filesystem.readKbPS.avg
os.diskIO.filesystem.writeKbPS.avg
os.diskIO.nvme1n1.readKbPS.avg
Vacuum: Maximum transaction IDs (Transactions)
241,291
215,580
189,869
db.Transactions.max_used_xact_ids.avg

Monitoraggio dei parametri di Amazon RDS con Amazon CloudWatch

Amazon CloudWatch è un repository di parametri. Il repository raccoglie ed elabora i dati non elaborati da Amazon RDS in parametri leggibili quasi in tempo reale. Per l'elenco completo dei parametri di Amazon RDS inviati a CloudWatch, consulta Guida di riferimento per i parametri di Amazon RDS.

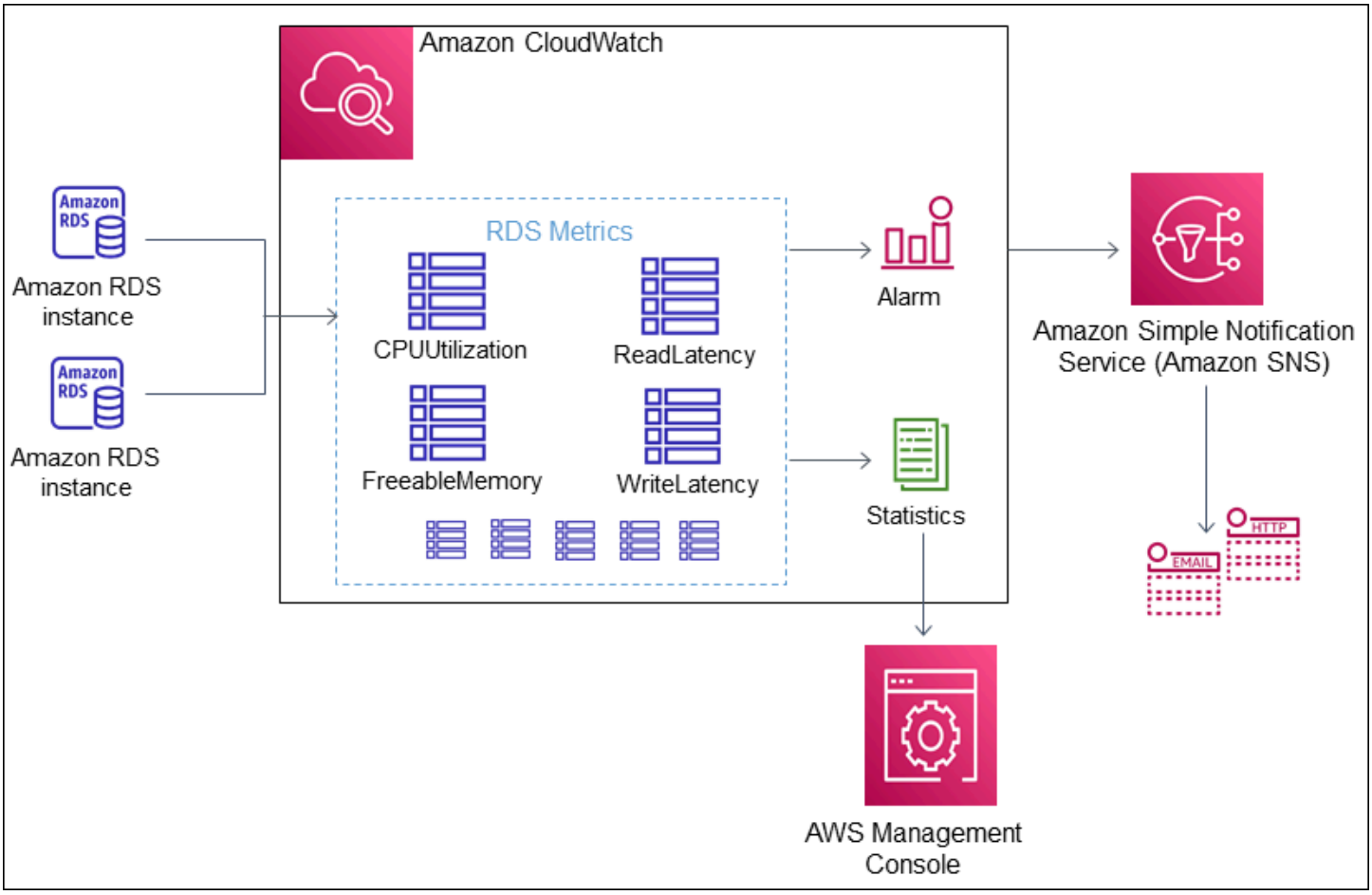
Argomenti

Panoramica di Amazon RDS e Amazon CloudWatch

Per impostazione predefinita, Amazon RDS invia automaticamente i dati dei parametri a CloudWatch a intervalli di 1 minuto. Ad esempio, il parametro `CPUUtilization` registra la percentuale di utilizzo della CPU per un'istanza database nel tempo. I punti di dati con un periodo di 60 secondi (1 minuto) sono disponibili per 15 giorni. Ciò significa che è possibile accedere alle informazioni della cronologia e visualizzare le prestazioni del servizio o dell'applicazione Web.

Ora puoi esportare le dashboard dei parametri di Approfondimenti sulle prestazioni da Amazon RDS ad Amazon CloudWatch. Puoi esportare le dashboard dei parametri preconfigurate o personalizzate come nuove dashboard o aggiungerle a una dashboard CloudWatch esistente. La dashboard esportata è disponibile per la visualizzazione nella console CloudWatch. Per ulteriori informazioni su come esportare le dashboard dei parametri di Approfondimenti sulle prestazioni su CloudWatch, consulta Esportazione delle metriche di Performance Insights in CloudWatch.

Come illustrato nel seguente diagramma, è possibile impostare gli allarmi per i parametri di CloudWatch. Ad esempio, puoi creare un allarme che segnali quando l'utilizzo della CPU per un'istanza è superiore al 70%. È possibile configurare Amazon Simple Notification Service in modo da ricevere un messaggio e-mail quando viene superata la soglia.



Amazon RDS pubblica i seguenti tipi di parametri in Amazon CloudWatch:

- Parametri per le istanze database RDS

  Per una tabella di questi parametri, consulta CloudWatch Parametri Amazon per Amazon RDS.
- Parametri Performance Insights

  Per una tabella di questi parametri, consulta CloudWatch Metriche Amazon per Performance Insights e Parametri contatore di Performance Insights.
- Parametri di Monitoraggio avanzato (pubblicati in Amazon CloudWatch Logs)

  Per una tabella di questi parametri, consulta Parametri del sistema operativo nel monitoraggio avanzato.
- Parametri di utilizzo per le quote di servizio Amazon RDS nell'Account AWS

  Per una tabella di questi parametri, consulta . Per ulteriori informazioni sulle quote di Amazon RDS, consulta Quote e vincoli per Amazon RDS.

Per ulteriori informazioni su CloudWatch, consulta Che cos'è Amazon CloudWatch? nella Guida per l'utente di Amazon CloudWatch. Per ulteriori informazioni sulla conservazione dei parametri di CloudWatch, consulta Conservazione dei parametri.

Visualizzazione delle metriche di istanze DB nella CloudWatch console e AWS CLI

Di seguito, puoi trovare dettagli su come visualizzare le metriche per la tua istanza DB utilizzando. CloudWatch Per informazioni sul monitoraggio delle metriche per il sistema operativo dell'istanza DB in tempo reale tramite CloudWatch Logs, consulta. Monitoraggio dei parametri del sistema operativo con il monitoraggio avanzato

Quando usi le risorse Amazon RDS Aurora, Amazon RDS Amazon parametri e dimensioni ad Amazon ogni minuto. CloudWatch

Ora puoi esportare i dashboard dei parametri di Performance Insights da Amazon RDS ad Amazon CloudWatch e visualizzarli nella console. CloudWatch Per ulteriori informazioni su come esportare i dashboard delle metriche di Performance Insights in CloudWatch, consulta. Esportazione delle metriche di Performance Insights in CloudWatch

Utilizza le seguenti procedure per visualizzare i parametri per Amazon RDS Amazon nella CloudWatch console e nella CLI.

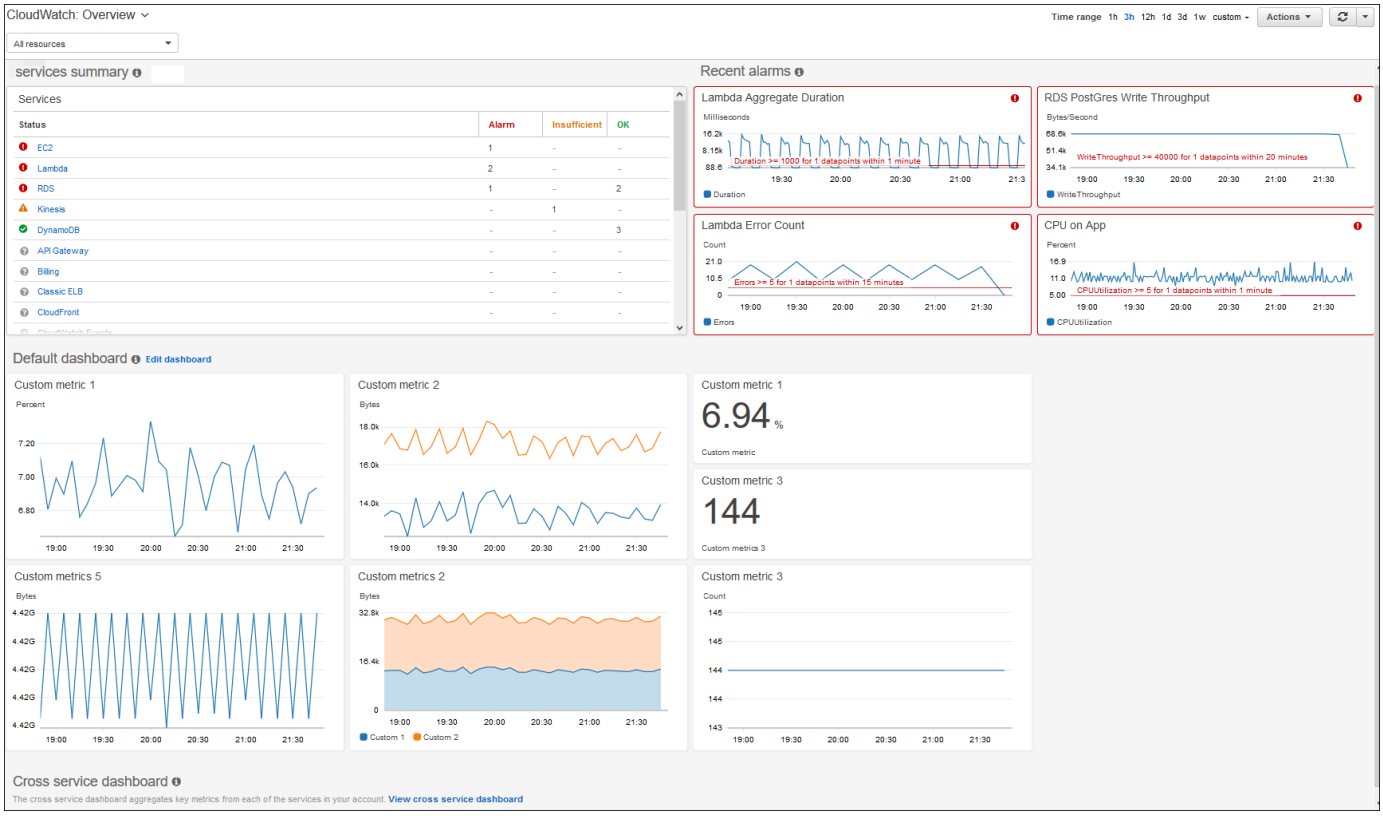
Console

Per visualizzare le metriche utilizzando la console Amazon CloudWatch

I parametri vengono raggruppati prima in base allo spazio dei nomi del servizio e successivamente in base alle diverse combinazioni di dimensioni all'interno di ogni spazio dei nomi.

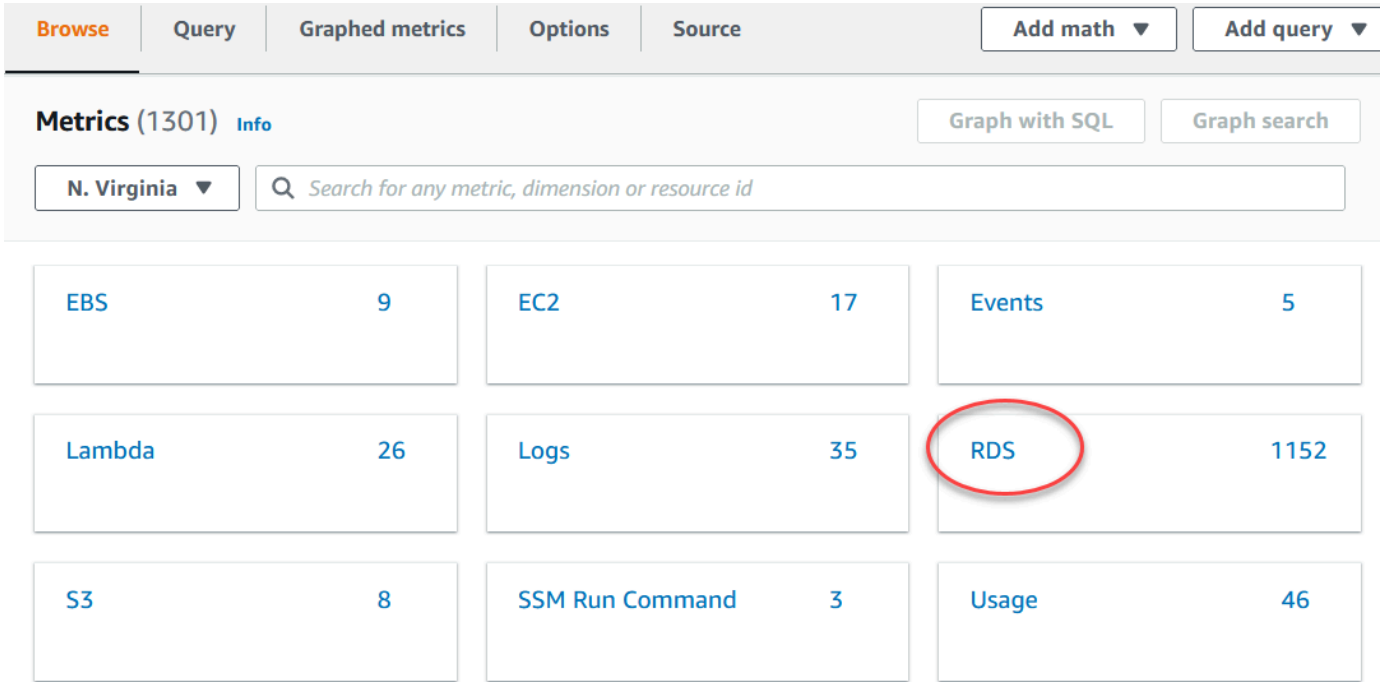
1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.

   Viene visualizzata la home page di CloudWatch panoramica.



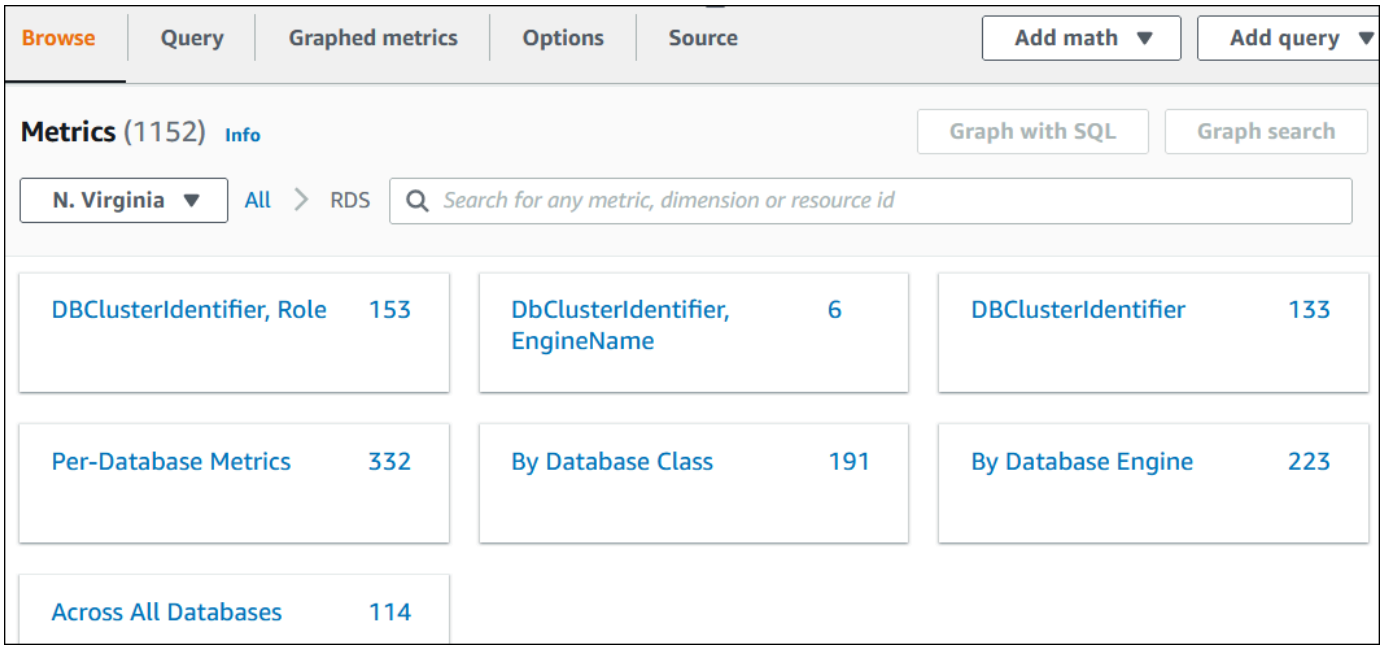
2. Se necessario, modifica Regione AWS. Dalla barra di navigazione, seleziona la Regione AWS in cui si trovano le risorse AWS. Per ulteriori informazioni, consulta Regioni ed endpoint.

3. Nel pannello di navigazione, scegli Metrics (Parametri), quindi scegli All metrics (Tutti i parametri).

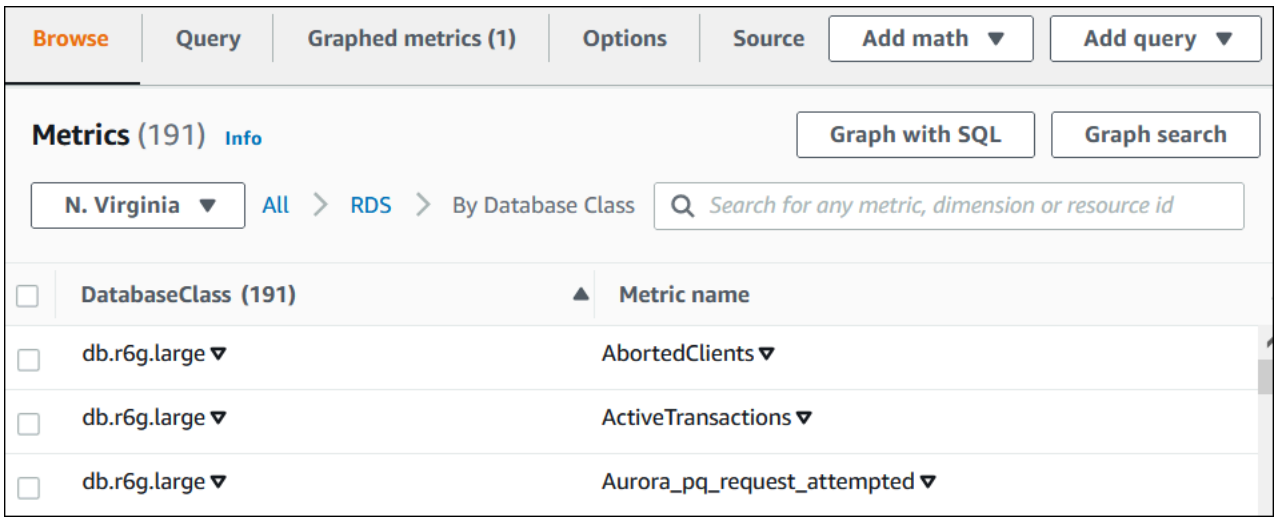


4. Scorri verso il basso e scegli il parametro namespace RDS.

   La pagina mostra le dimensioni Amazon RDS. Per una descrizione di queste dimensioni, consulta Le dimensioni di Amazon CloudWatch per Amazon RDS.



5. Scegli una dimensione di parametro, ad esempio By Database Class (Per classe di database).

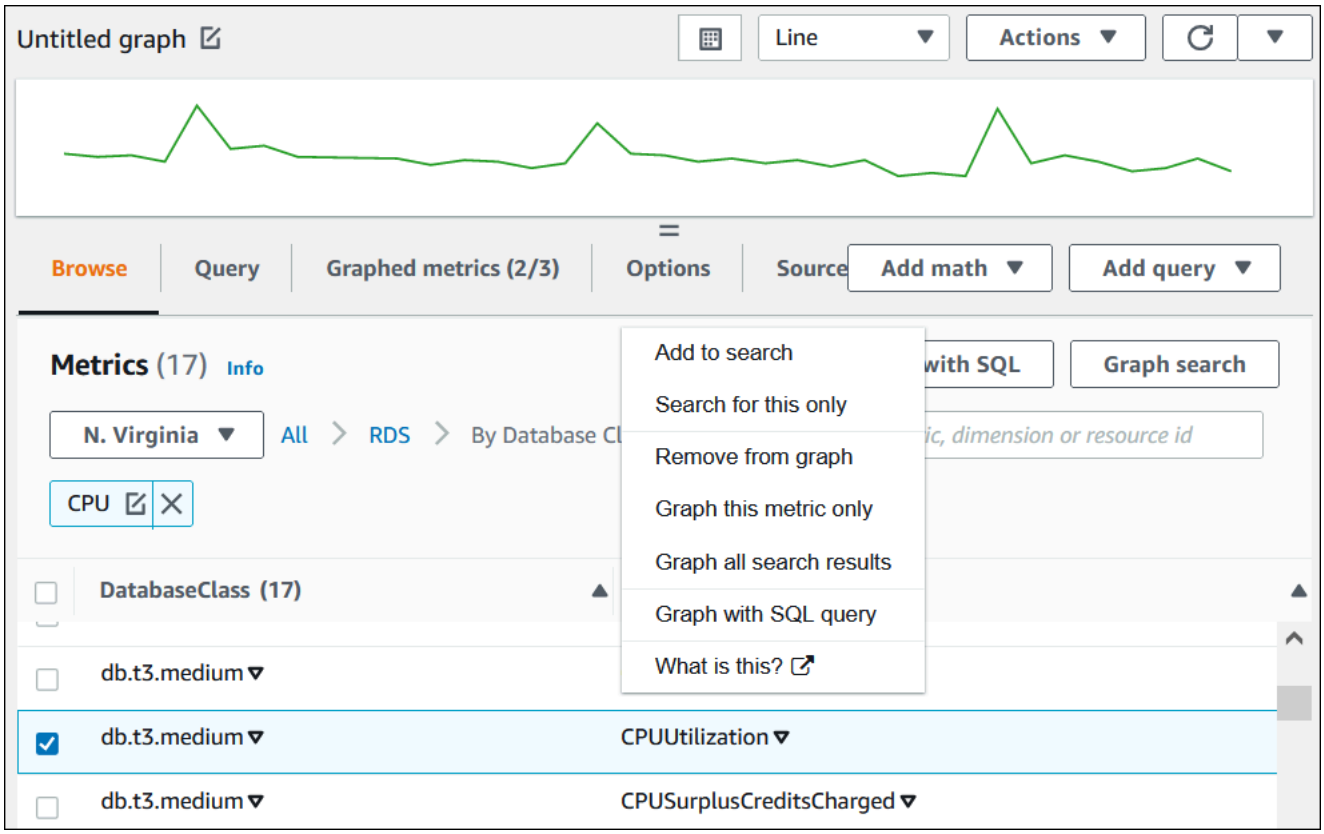


6. Effettua una delle seguenti operazioni:

   - Per ordinare i parametri, utilizza l'intestazione della colonna.
   - Per creare il grafico di un parametro, seleziona la casella di controllo accanto al parametro.
   - Per filtrare in base a una risorsa, scegli l'ID della risorsa e quindi Add to search (Aggiungi alla ricerca).

- Per filtrare in base a un parametro, scegli il nome del parametro e quindi Add to search (Aggiungi alla ricerca).

Il seguente esempio filtra sui grafici e sulla classe db.t3.medium il parametro CPUUtilization.



AWS CLI

Per ottenere informazioni metriche utilizzando il, usa il comando. AWS CLI CloudWatch `list-metrics` Nell'esempio seguente, vengono elencati tutti i parametri nello spazio dei nomi `AWS/RDS`.

```
aws cloudwatch list-metrics --namespace AWS/RDS
```

Per ottenere dati metrici, utilizzate il comando. `get-metric-data`

L'esempio seguente ottiene `CPUUtilization` statistiche, ad esempio, `my-instance` su un periodo di 24 ore specifico, con una granularità di 5 minuti.

Crea un file JSON `CPU_metric.json` con i seguenti contenuti.

```
{
```

```
   "StartTime" : "2023-12-25T00:00:00Z",
   "EndTime" : "2023-12-26T00:00:00Z",
   "MetricDataQueries" : [{
     "Id" : "cpu",
     "MetricStat" : {
    "Metric" : {
        "Namespace" : "AWS/RDS",
        "MetricName" : "CPUUtilization",
        "Dimensions" : [{ "Name" : "DBInstanceIdentifier" , "Value" : my-instance}]
    },
       "Period" : 360,
       "Stat" : "Minimum"
     }
   }]
}
```

Example

PerLinux, omacOS: Unix

```
aws cloudwatch get-metric-data \
    --cli-input-json file://CPU_metric.json
```

Per Windows:

```
aws cloudwatch get-metric-data ^
     --cli-input-json file://CPU_metric.json
```

L'output di esempio viene visualizzato come segue:

```
{
    "MetricDataResults": [
        {
            "Id": "cpu",
            "Label": "CPUUtilization",
            "Timestamps": [
                "2023-12-15T23:48:00+00:00",
                "2023-12-15T23:42:00+00:00",
                "2023-12-15T23:30:00+00:00",
                "2023-12-15T23:24:00+00:00",
                ...
            ],
            "Values": [
```

```
                13.299778337027714,
                13.677507543049558,
                14.24976250395827,
                13.02521708695145,
                ...
            ],
            "StatusCode": "Complete"
        }
    ],
    "Messages": []
}
```

Per ulteriori informazioni, consulta Ottenere statistiche per una metrica nella Amazon CloudWatch User Guide.

Esportazione delle metriche di Performance Insights in CloudWatch

Performance Insights ti consente di esportare il pannello di controllo delle metriche preconfigurato o personalizzato per la tua istanza DB su Amazon. CloudWatch Puoi esportare la dashboard delle metriche come nuova dashboard o aggiungerla a una dashboard esistente. CloudWatch Quando scegli di aggiungere la dashboard a una CloudWatch dashboard esistente, puoi creare un'etichetta di intestazione in modo che le metriche vengano visualizzate in una sezione separata della dashboard. CloudWatch

Puoi visualizzare la dashboard delle metriche esportate nella console. CloudWatch Se aggiungi nuove metriche a una dashboard delle metriche di Performance Insights dopo averla esportata, devi esportare nuovamente questa dashboard per visualizzare le nuove metriche nella console. CloudWatch

Puoi anche selezionare un widget metrico nella dashboard di Performance Insights e visualizzare i dati delle metriche nella CloudWatch console.

Per ulteriori informazioni sulla visualizzazione delle metriche nella CloudWatch console, consulta. Visualizzazione delle metriche di istanze DB nella CloudWatch console e AWS CLI

Esportazione delle metriche di Performance Insights come nuova dashboard in CloudWatch

Scegli una dashboard delle metriche preconfigurata o personalizzata dalla dashboard di Performance Insights ed esportala come nuova dashboard in. CloudWatch Puoi visualizzare la dashboard esportata nella console. CloudWatch

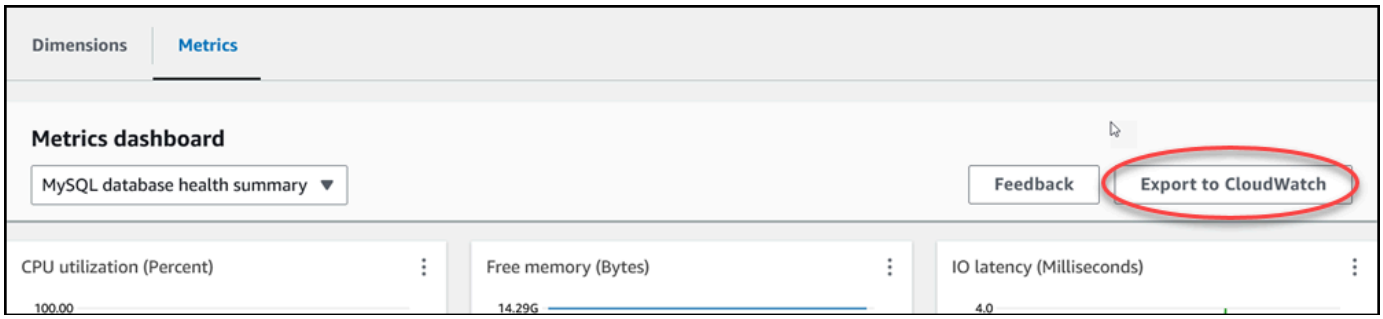
Per esportare una dashboard metrica di Performance Insights come nuova dashboard in CloudWatch

1. Apri la console di Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel pannello di navigazione scegli Performance Insights.
3. Scegli un'istanza database.

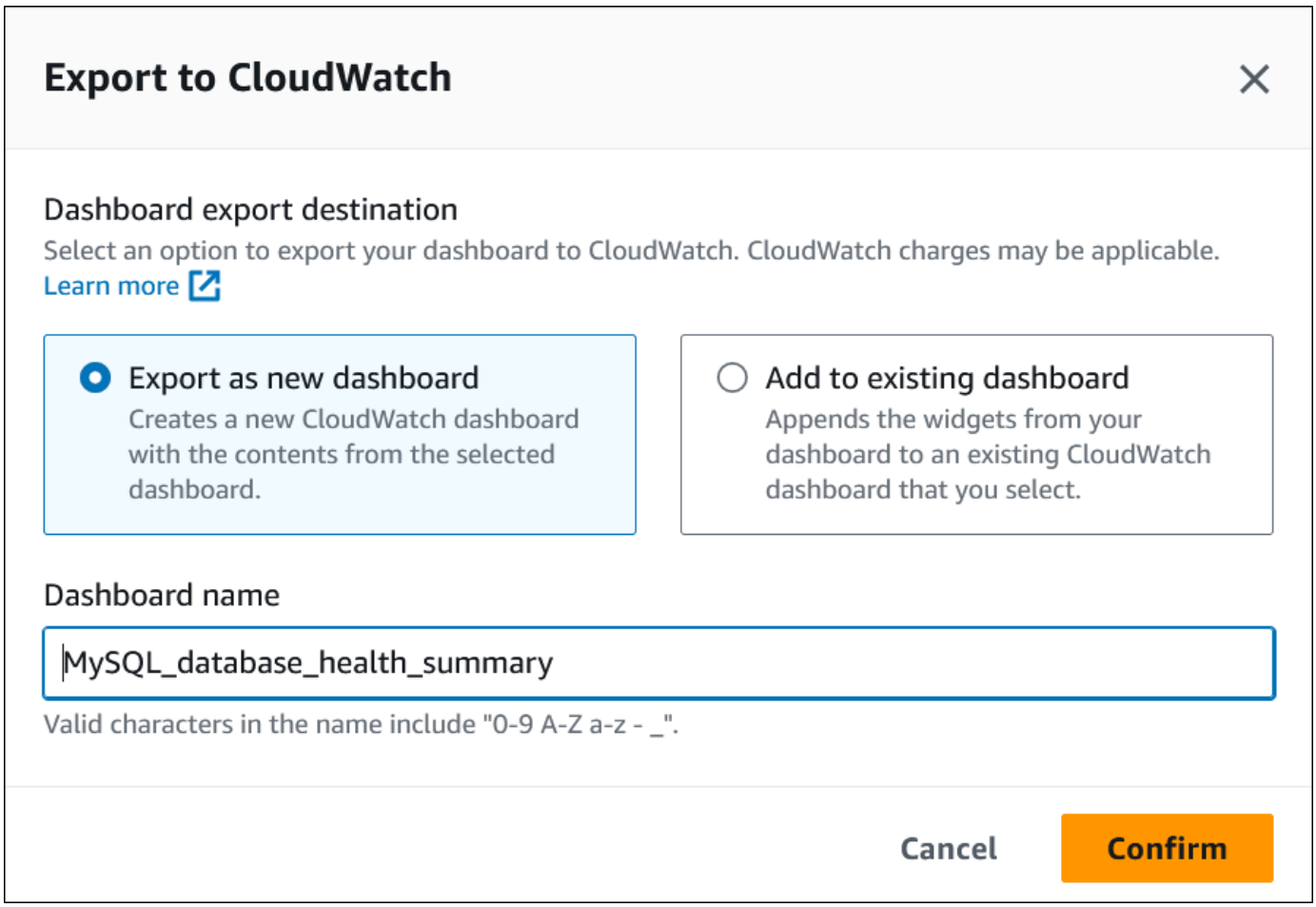
   Viene visualizzato il pannello di controllo di Approfondimenti sulle prestazioni per l'istanza database.
4. Scorri verso il basso e scegli Parametri.

   Per impostazione predefinita, viene visualizzata la dashboard preconfigurata con i parametri di Approfondimenti sulle prestazioni.
5. Scegli una dashboard preconfigurata o personalizzata, quindi scegli Esporta in. CloudWatch

   Viene visualizzata la CloudWatch finestra Esporta in.

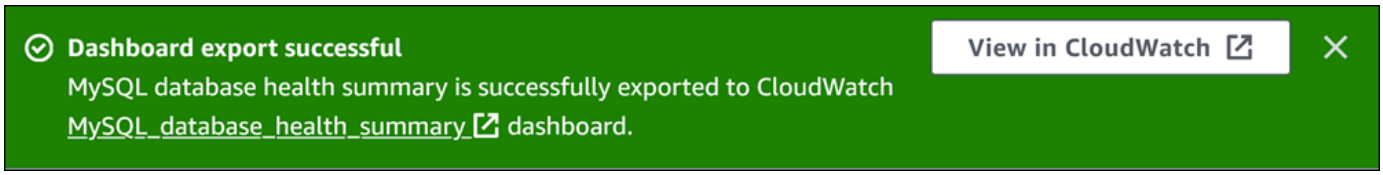


6. Scegli Esporta come nuova dashboard.



7. Immetti un nome per la nuova dashboard nel campo Nome della dashboard e scegli Conferma.

   Un banner mostra un messaggio al completamento dell'esportazione della dashboard.



8. Scegli il link o Visualizza CloudWatch nel banner per visualizzare la dashboard delle metriche nella CloudWatch console.

Aggiungere metriche di Performance Insights a una dashboard esistente CloudWatch

Aggiungi un pannello di controllo delle metriche preconfigurato o personalizzato a un pannello di controllo esistente. CloudWatch Puoi aggiungere un'etichetta alla dashboard delle metriche da visualizzare in una sezione separata della dashboard. CloudWatch

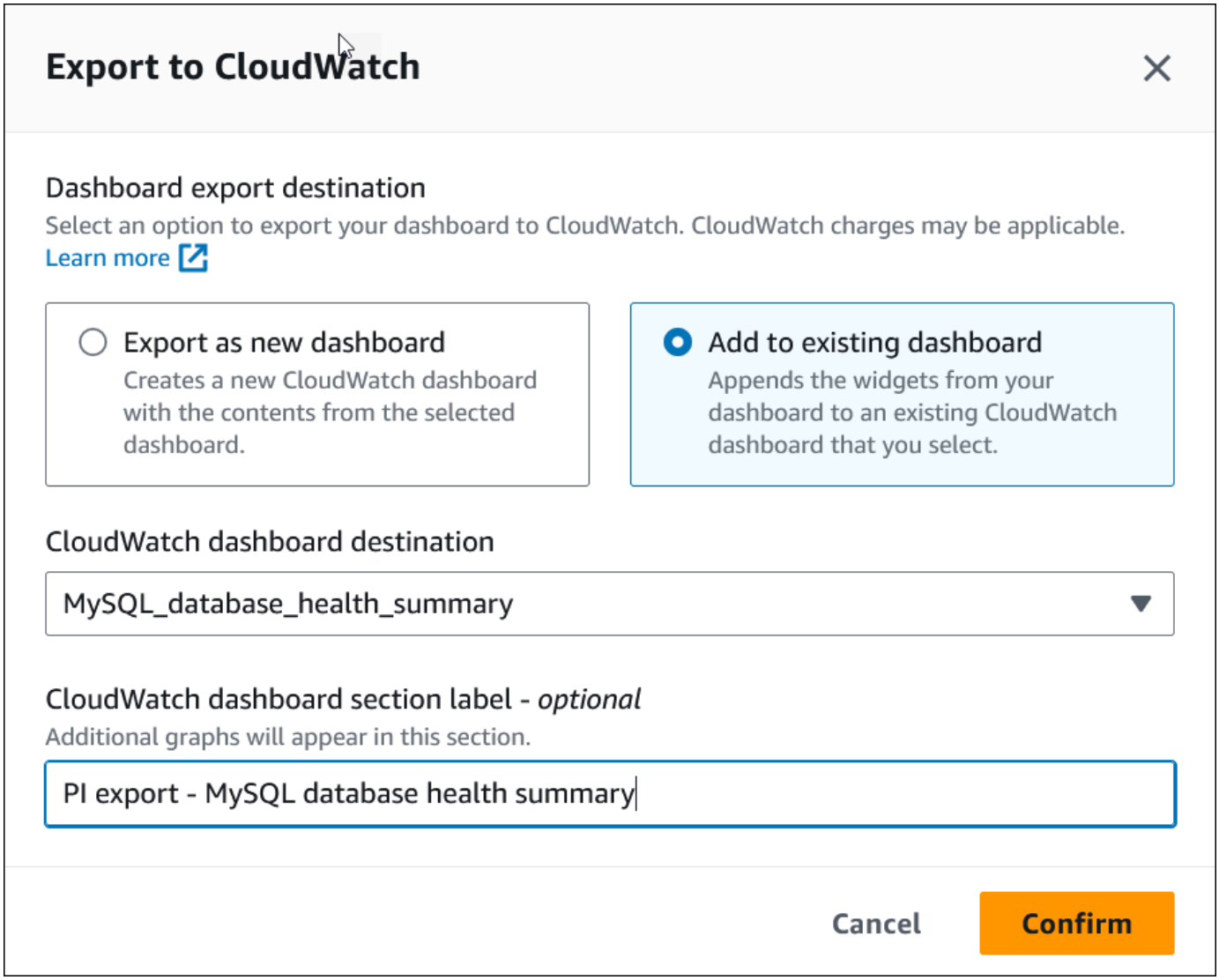
Per esportare le metriche in una dashboard esistente CloudWatch

1. Apri la console di Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel pannello di navigazione scegli Performance Insights.
3. Scegli un'istanza database.

   Viene visualizzato il pannello di controllo di Approfondimenti sulle prestazioni per l'istanza database.
4. Scorri verso il basso e scegli Parametri.

   Per impostazione predefinita, viene visualizzata la dashboard preconfigurata con i parametri di Approfondimenti sulle prestazioni.
5. Scegli la dashboard preconfigurata o personalizzata, quindi scegli Esporta in. CloudWatch

   Viene visualizzata la CloudWatch finestra Esporta in.
6. Scegli Aggiungi alla dashboard esistente.



7. Specifica la destinazione e l'etichetta della dashboard, quindi scegli Conferma.

   - CloudWatch destinazione del pannello di controllo: scegli un CloudWatch pannello di controllo esistente.
   - CloudWatch etichetta della sezione dashboard - opzionale - Inserisci un nome per le metriche di Performance Insights da visualizzare in questa sezione del CloudWatch dashboard.

   Un banner mostra un messaggio al completamento dell'esportazione della dashboard.

8. Scegli il link o Visualizza CloudWatch nel banner per visualizzare la dashboard delle metriche nella CloudWatch console.

Visualizzazione di un widget metrico Performance Insights in CloudWatch

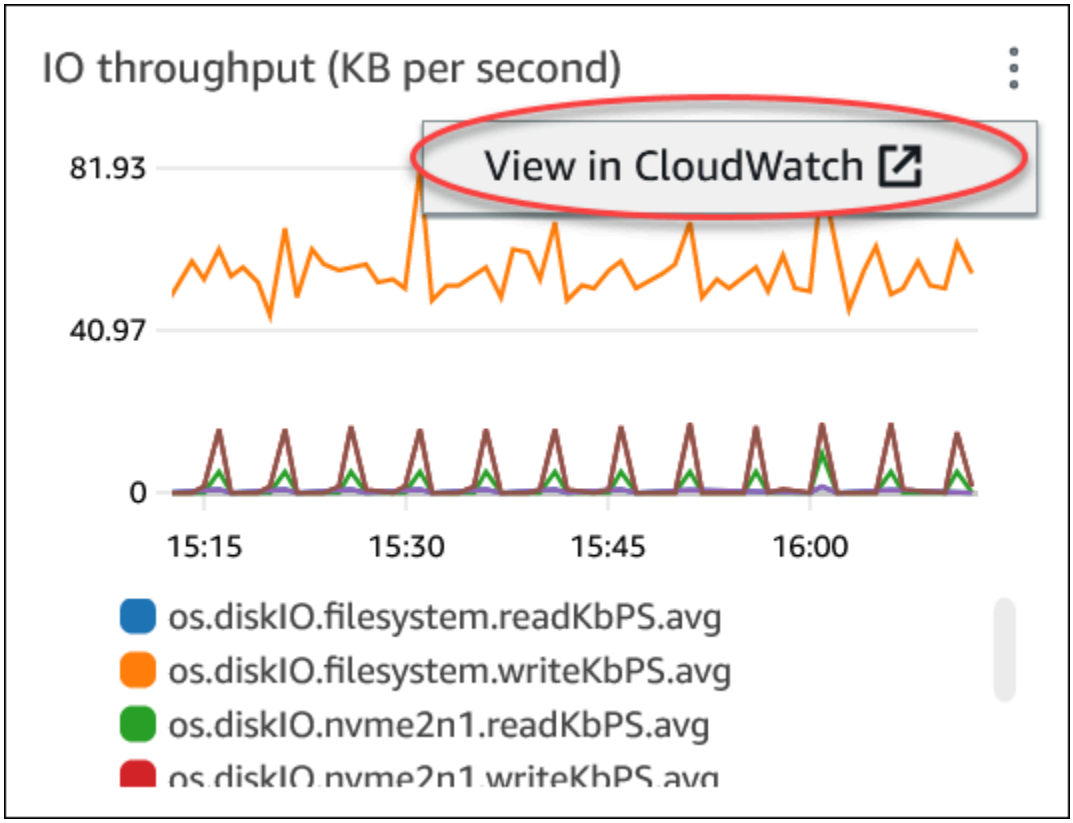
Seleziona un widget metrico Performance Insights nella dashboard di Amazon RDS Performance Insights e visualizza i dati delle metriche nella console. CloudWatch

Per esportare un widget metrico e visualizzare i dati delle metriche nella console CloudWatch

1. Apri la console di Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel pannello di navigazione scegli Performance Insights.
3. Scegli un'istanza database.

   Viene visualizzato il pannello di controllo di Approfondimenti sulle prestazioni per l'istanza database.
4. Scorri verso il basso fino a Parametri.

   Per impostazione predefinita, viene visualizzata la dashboard preconfigurata con i parametri di Approfondimenti sulle prestazioni.
5. Scegli un widget metrico, quindi scegli Visualizza CloudWatch nel menu.



   I dati metrici vengono visualizzati nella CloudWatch console.

Creazione di allarmi CloudWatch per il monitoraggio di Amazon RDS

Puoi creare un avviso CloudWatch che invia un messaggio Amazon SNS quando l'avviso cambia stato. Un allarme monitora un singolo parametro per un periodo di tempo specificato. L'allarme può anche eseguire una o più operazioni basate sul valore del parametro relativo a una soglia prestabilita per un certo numero di periodi. L'operazione corrisponde all'invio di una notifica a un argomento Amazon SNS o a una policy Amazon EC2 Auto Scaling.

Gli allarmi richiamano operazioni solo per le modifiche di stato prolungate. Gli allarmi CloudWatch non richiamano le operazioni semplicemente perché si trovano in uno stato particolare. È necessario che lo stato cambi e rimanga costante per un periodo specificato

Puoi utilizzare la funzione matematica composta da parametri DB_PERF_INSIGHTS nella console CloudWatch per eseguire query su Amazon RDS per i parametri contatore di Performance Insights. La funzione DB_PERF_INSIGHTS include anche la metrica DBLoad a intervalli inferiori al minuto. Puoi impostare gli allarmi CloudWatch sui questi parametri.

Per maggiori dettagli su come creare un allarme, consulta Creazione di un allarme sui parametri contatore di Performance Insights da un database AWS.

Per impostare un allarme mediante AWS CLI

- Chiamare `put-metric-alarm`. Per ulteriori informazioni, consulta il Riferimento ai comandi AWS CLI.

Per impostare un allarme mediante l'API di CloudWatch

- Chiamare `PutMetricAlarm`. Per maggiori informazioni, consulta la Documentazione di riferimento delle API di Amazon CloudWatch.

Per ulteriori informazioni sull'impostazione degli argomenti di Amazon SNS e sulla creazione degli allarmi, consulta Utilizzo degli allarmi di Amazon CloudWatch.

Tutorial: creazione di un allarme Amazon CloudWatch per il ritardo di replica del cluster di database Multi-AZ

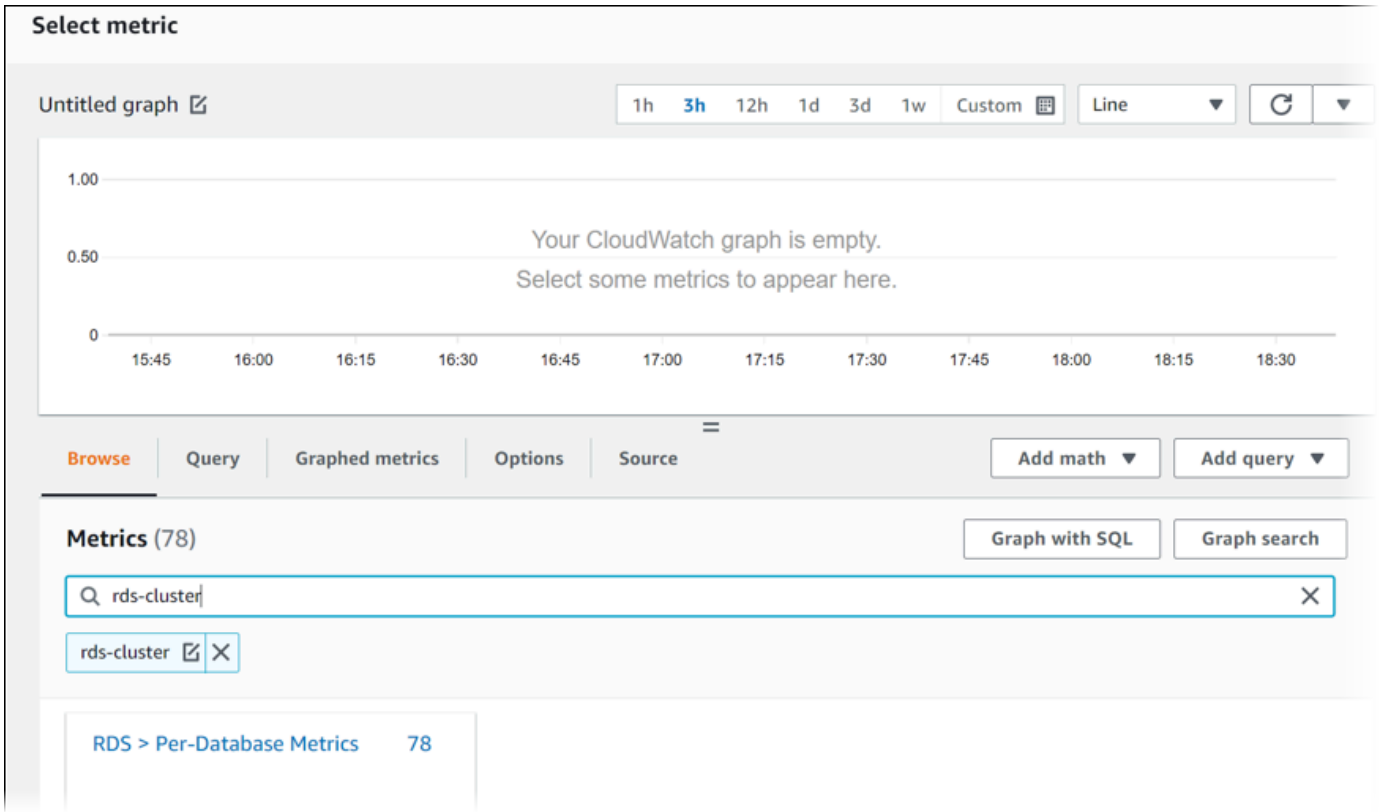
Puoi creare un allarme Amazon CloudWatch che invia un messaggio di Amazon SNS quando il ritardo di replica per un cluster di database Multi-AZ ha superato una soglia. Un allarme monitora il

parametro `ReplicaLag` per il periodo di tempo specificato. L'operazione corrisponde all'invio di una notifica a un argomento Amazon SNS o a una policy Amazon EC2 Auto Scaling.

Per impostare un allarme CloudWatch per il ritardo di replica del cluster di database Multi-AZ

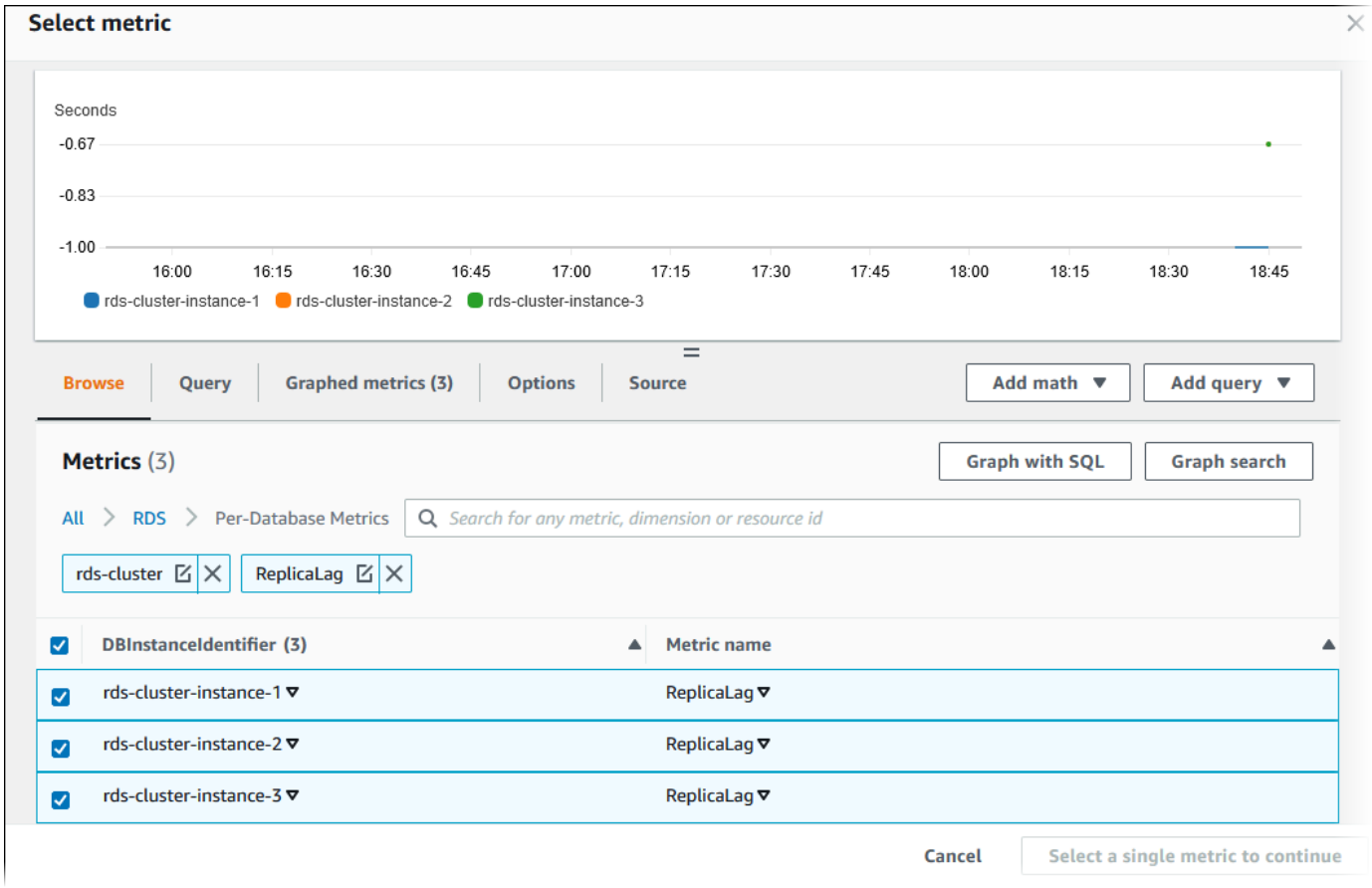
1. Accedi alla AWS Management Console e apri la console CloudWatch all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel pannello di navigazione, scegli Alarms (Allarmi), All alarms (Tutti gli allarmi).
3. Scegli Create Alarm (Crea allarme).
4. Nella pagina Specify metric and conditions (Specifica parametro e condizioni), scegliere Select metric (Seleziona parametro).
5. Nella casella di ricerca inserisci il nome del cluster di database Multi-AZ e premi Invio.

   L'immagine che segue mostra la pagina Select metric (Seleziona parametro) con inserito un cluster di database Multi-AZ denominato `rds-cluster`.



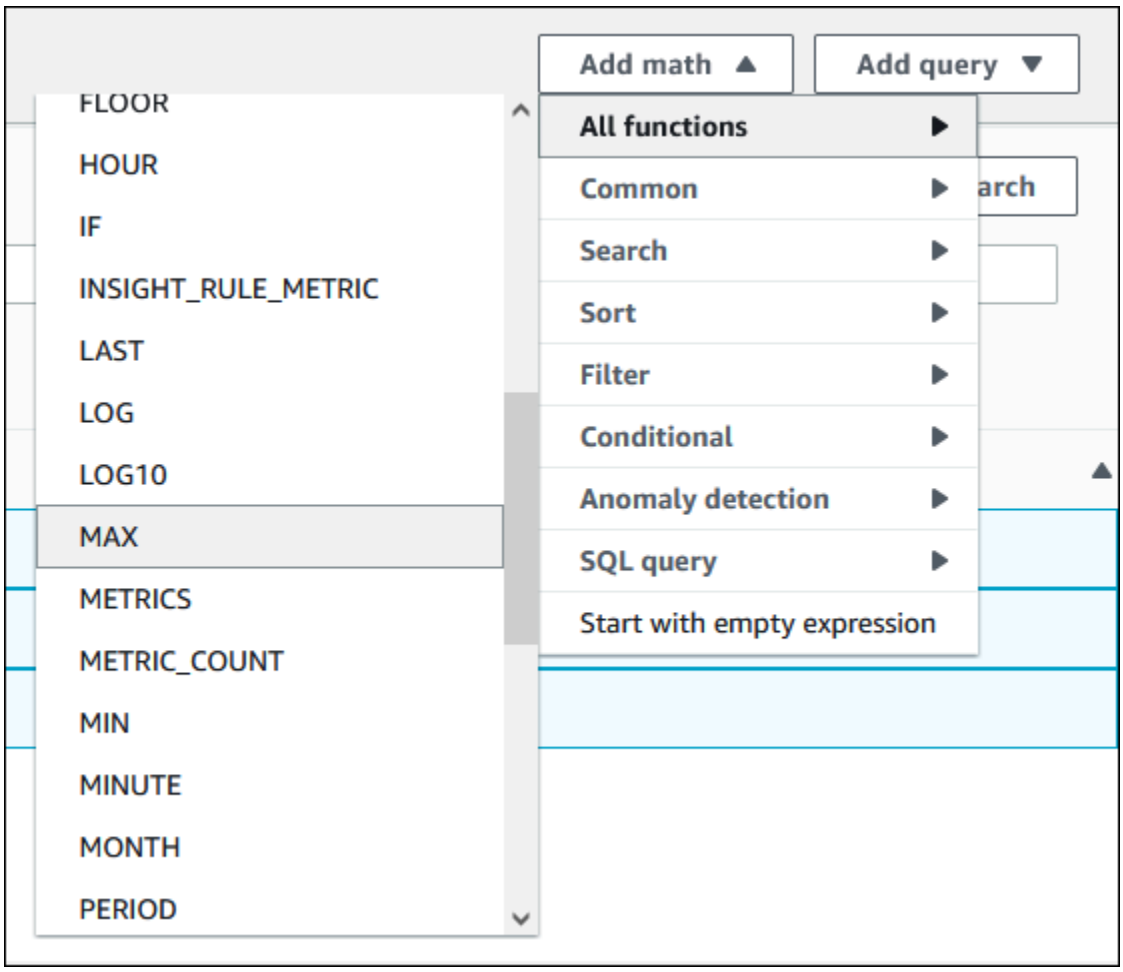
6. Scegli RDS, Per-Database Metrics (Parametri per database).
7. Nella casella di ricerca inserisci **`ReplicaLag`** e premi Invio, quindi seleziona ciascuna istanza database nel cluster di database.

L'immagine che segue mostra la pagina Select metric (Seleziona parametro) con le istanze database selezionate per il parametro ReplicaLag.



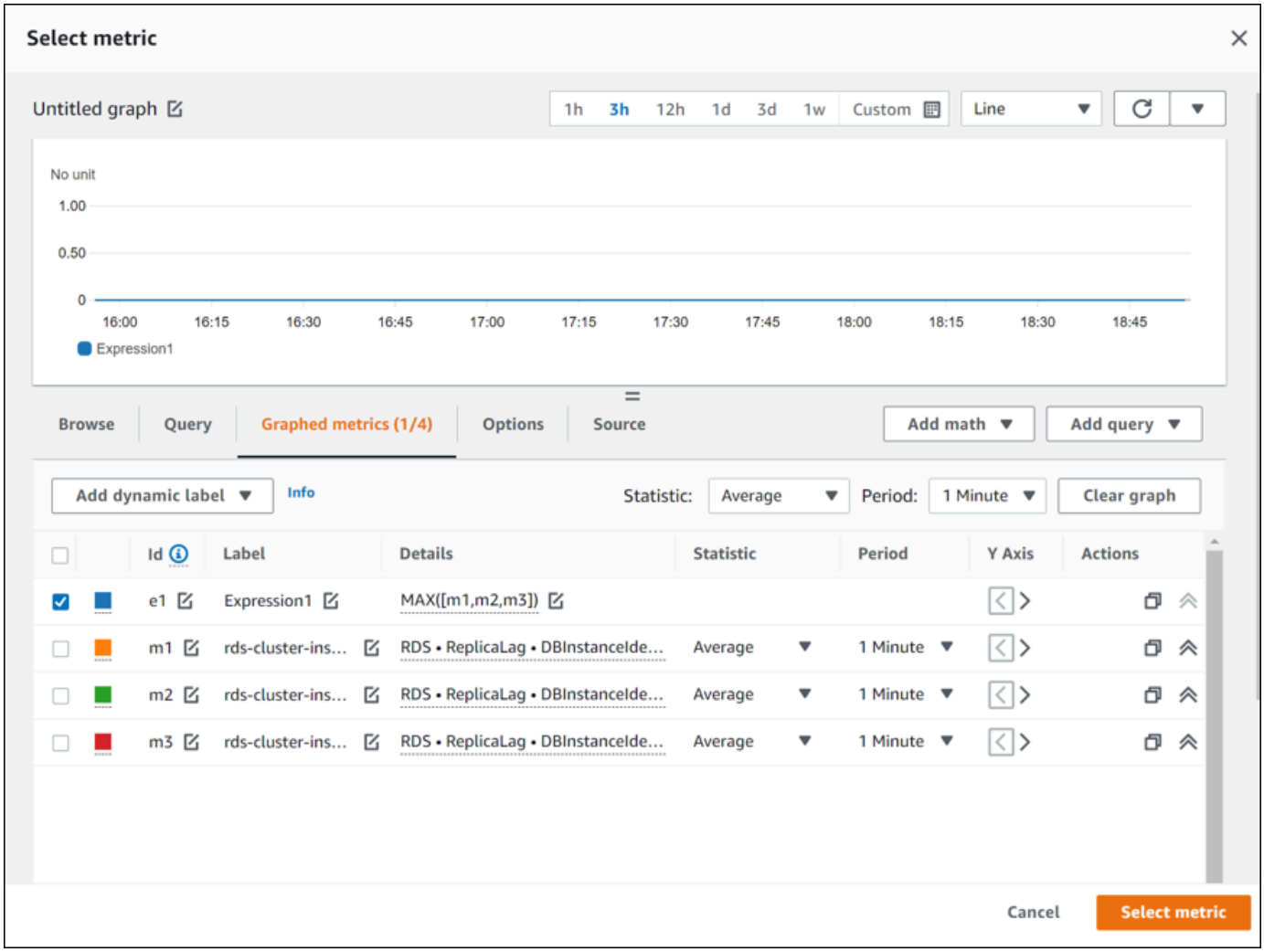
Questo allarme considera il ritardo di replica per tutte e tre le istanze database nel cluster di database Multi-AZ. L'allarme risponde quando una qualsiasi istanza database supera la soglia. Utilizza un'espressione matematica che restituisce il valore massimo dei tre parametri. Inizia ordinando in base al nome parametro, quindi scegli tutti e tre i parametri ReplicaLag.

8. In Add math (Aggiungi matematica), scegli All functions (Tutte le funzioni), MAX.



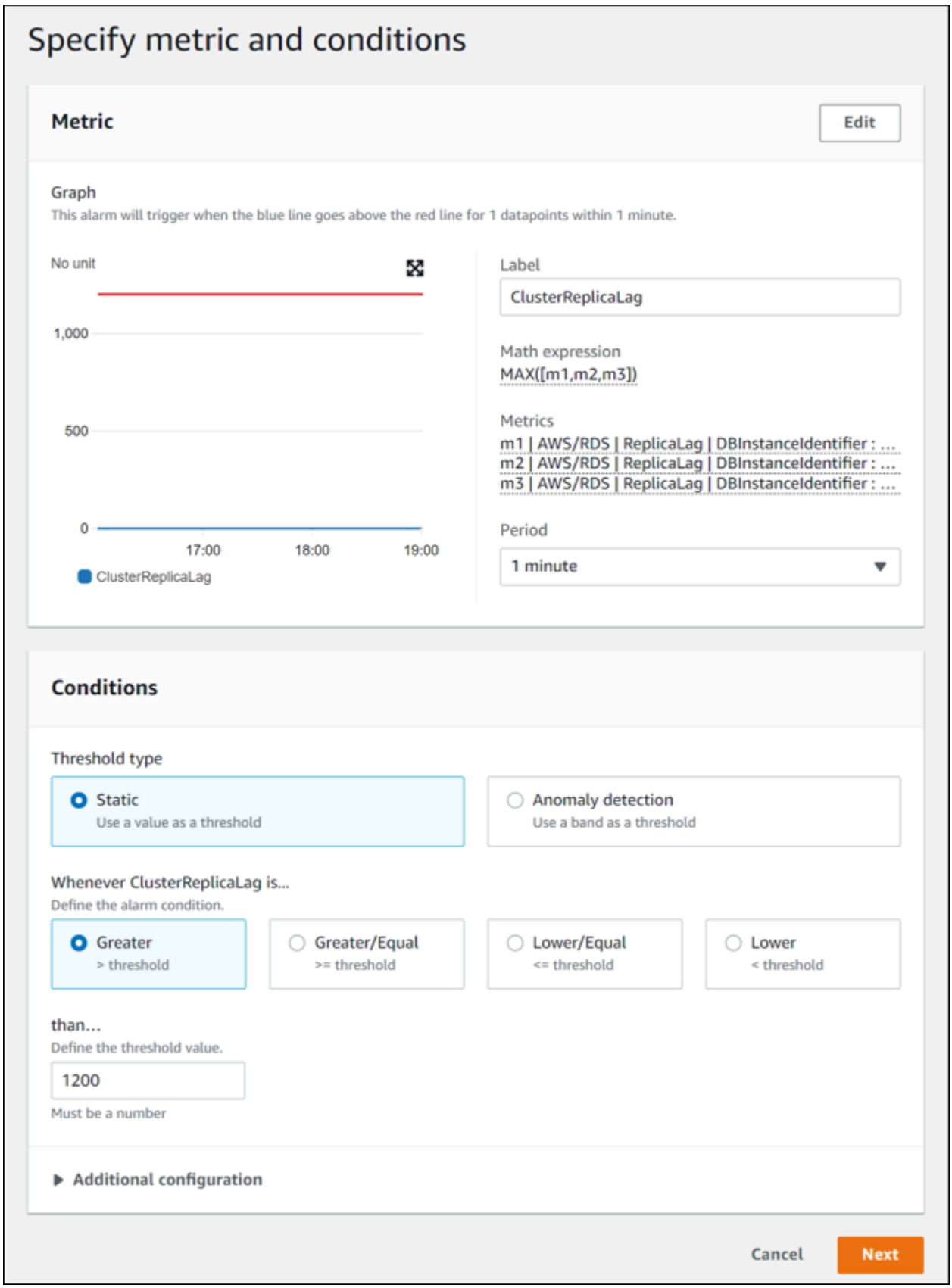
9. Seleziona la scheda Graphed metrics (Parametri nel grafico), quindi modifica i dettagli per Expression1 in **`MAX([m1,m2,m3])`**.
10. Per tutti e tre i parametri ReplicaLag, cambia il Period (Periodo) in 1 minute (1 minuto).
11. Cancella la selezione da tutti i parametri tranne che per Expression1.

    La pagina Select metric (Seleziona parametro) dovrebbe apparire simile alla seguente immagine.



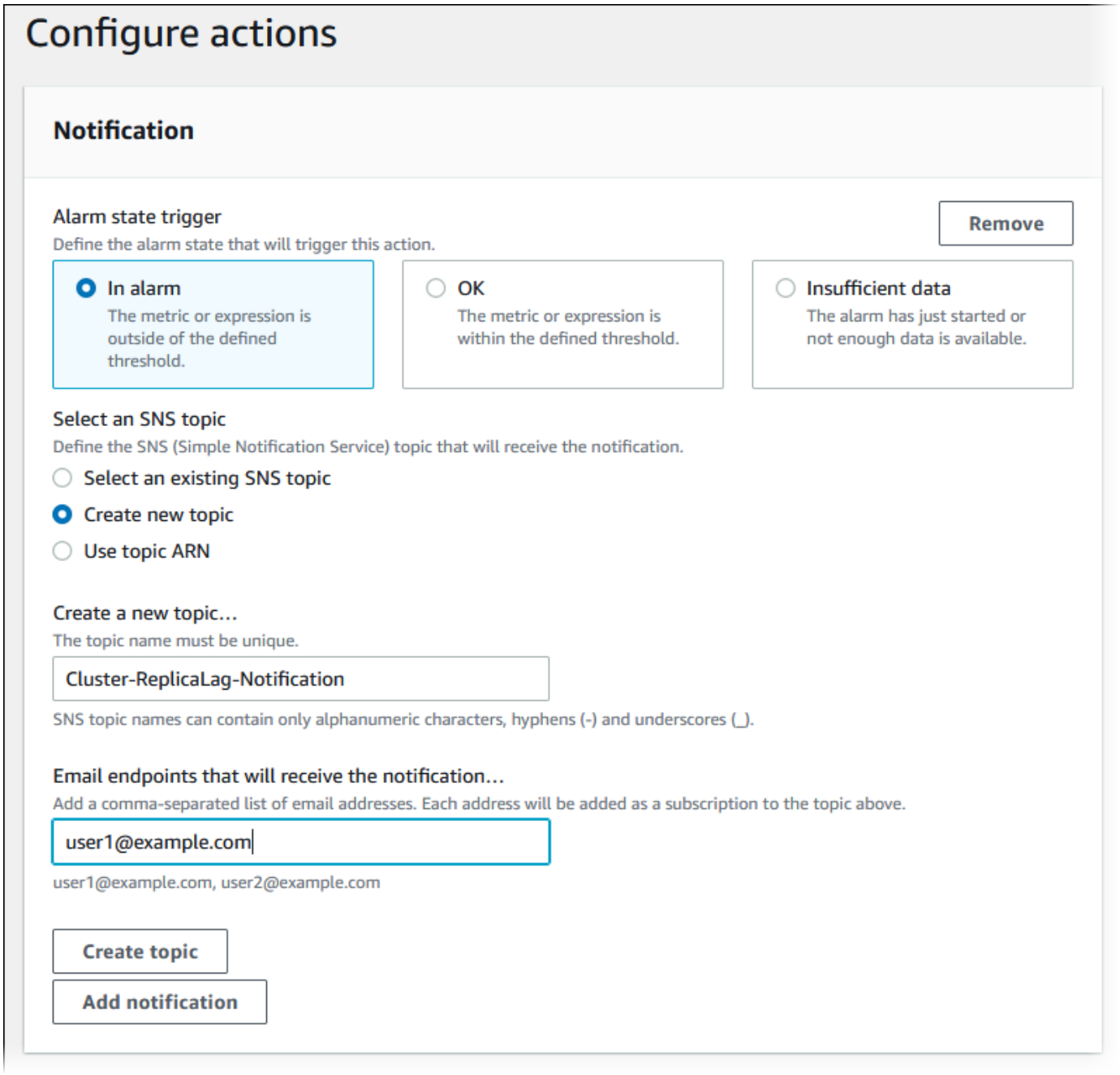
12. Scegli Select Metric (Seleziona parametro).
13. Nella pagina Specify metric and conditions (Specifica parametro e condizioni), modifica l'etichetta con un nome significativo, ad esempio **`ClusterReplicaLag`** e inserisci un numero di secondi in Define the threshold value (Definisci il valore di soglia. Per questo tutorial, seleziona **`1200`** secondi (20 minuti). È possibile modificare questo valore in base ai requisiti del carico di lavoro.

    La pagina Specify metric and conditions (Specifica parametro e condizioni) dovrebbe apparire simile alla seguente immagine.

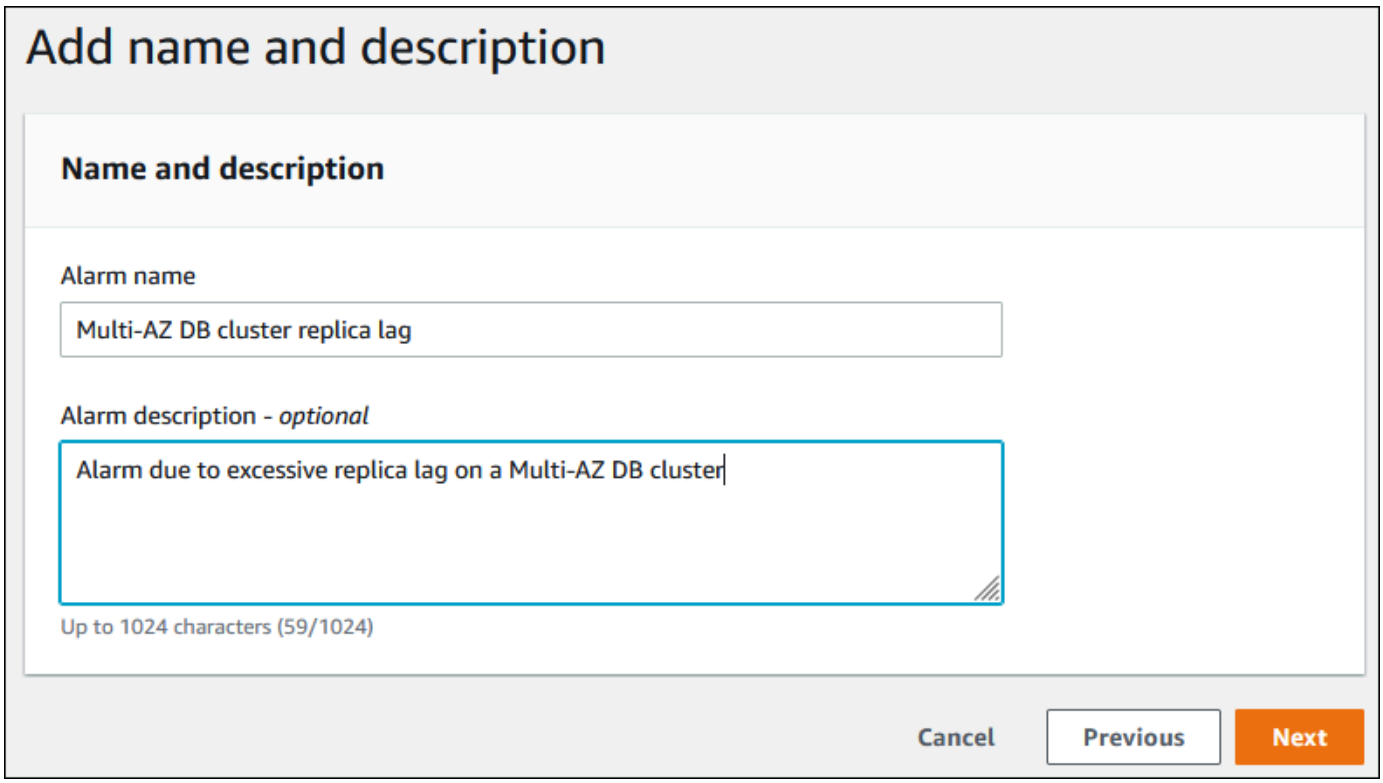


14. Scegli Next (Avanti) e viene visualizzata la pagina Configure actions (Configura azioni).

15. Mantieni In alarm (In allarme) selezionato, scegli Create new topic (Crea nuovo argomento) e inserisci il nome dell'argomento e un indirizzo e-mail valido.



16. Sceglie Create topic (Crea argomento), quindi seleziona Next (Avanti).
17. Nella pagina Add name and description (Aggiungi nome e descrizione), inserisci Alarm name (Nome dell'allarme) e Alarm description (Descrizione dell'allarme) e scegli Next (Successivo).



18. Visualizza in anteprima l'avviso che stai per creare nell'area Preview and create (Visualizza anteprima e crea), quindi scegli Create alarm (Crea avviso).

Monitoraggio del carico DB con Performance Insights su Amazon RDS

Performance Insights si espande sulle caratteristiche di monitoraggio esistenti di Amazon RDS per illustrare e aiutare ad analizzare le prestazioni del database. Con il pannello di controllo di Performance Insights, puoi visualizzare il carico del database sull'istanza database Amazon RDS e filtrare il carico in base alle attese, alle istruzioni SQL, agli host o agli utenti. Per informazioni sull'uso di Performance Insights con Amazon DocumentDB, consulta Guida per gli sviluppatori di Amazon DocumentDB.

Argomenti

Panoramica di Performance Insights su Amazon RDS

Per impostazione predefinita, RDS abilita Performance Insights nella procedura guidata di creazione della console per tutti i motori Amazon RDS. Se in un'istanza database sono presenti più database, Performance Insights aggrega i dati sulle prestazioni.

Puoi trovare una panoramica di Performance Insights per Amazon RDS nel seguente video.

Uso di Performance Insights per analizzare le prestazioni di Amazon Aurora PostgreSQL

> **Important**
>
> I seguenti argomenti descrivono l'utilizzo di Amazon RDS Performance Insights con motori di database non Aurora. Per informazioni sull'utilizzo di Amazon RDS Performance Insights con

Amazon Aurora, consulta Uso di Amazon RDS Performance Insights nella Guida per l'utente di Amazon Aurora.

Argomenti

- Caricamento database
- CPU massima
- Supporto di classe di istanza, regione e motore di database Amazon RDS per Performance Insights
- Prezzi e conservazione dei dati per Performance Insights

Caricamento database

Il carico del database (carico DB) misura il livello di attività della sessione nel database. `DBLoad`è la metrica chiave di Performance Insights e Performance Insights raccoglie il carico del DB ogni secondo.

Argomenti

- Sessioni attive
- Media delle sessioni attive
- Media delle esecuzioni attive
- Dimensioni

Sessioni attive

Una sessione database rappresenta il dialogo di un'applicazione con un database relazionale. Una sessione attiva è una connessione che ha inviato lavoro a un motore del database ed è in attesa di una risposta dal motore del database.

Una sessione è attiva quando è in esecuzione sulla CPU o in attesa che una risorsa diventi disponibile in modo che possa proseguire. Ad esempio, una sessione attiva potrebbe attendere la lettura di una pagina (o blocco) in memoria e quindi consumare la CPU mentre legge i dati dalla pagina.

Media delle sessioni attive

La media delle sessioni attive (AAS) è l'unità per il parametro `DBLoad` in Performance Insights. Misura quante sessioni sono attive contemporaneamente nel database.

Ogni secondo, Performance Insights esegue il campionamento del numero di sessioni che eseguono contemporaneamente una query. Per ogni sessione attiva, Performance Insights raccoglie i seguenti dati:

- Istruzione SQL
- Stato della sessione (in esecuzione sulla CPU o in attesa)
- Host
- Utente che esegue SQL

Performance Insights calcola il valore delle sessioni attive medie (AAS) dividendo il numero totale di sessioni per il numero totale di campioni per un periodo di tempo specifico. Ad esempio, nella tabella seguente vengono riportati 5 campioni consecutivi di una query in esecuzione, dove ogni campione viene acquisito a intervalli di 1 secondo.

Project N.E.M.O.	Numero di sessioni che eseguono query	AAS	Calcolo
1	2	2	2 sessioni totali / 1 campione
2	0	1	2 sessioni totali / 2 campioni
3	4	2	6 sessioni totali / 3 campioni
4	0	1.5	6 sessioni totali / 4 campioni
5	4	2	10 sessioni totali / 5 campioni

Nell'esempio precedente, il carico DB per l'intervallo di tempo è 2 AAS. Questa misurazione significa che, in media, sono state attive 2 sessioni alla volta durante il periodo in cui sono stati acquisiti i 5 campioni.

Media delle esecuzioni attive

La media delle esecuzioni attive (AAE) al secondo è correlata all'AAS. Per calcolare l'AAE, Performance Insights divide il tempo totale di esecuzione di una query per l'intervallo di tempo. Nella tabella seguente viene illustrato il calcolo AAE per la stessa query nella tabella precedente.

Tempo trascorso (sec)	Tempo di esecuzione totale (sec)	AAE	Calcolo
60	120	2	120 secondi di esecuzione/60 secondi trascorsi
120	120	1	120 secondi di esecuzione/120 secondi trascorsi
180	380	2.11	380 secondi di esecuzione/180 secondi trascorsi
240	380	1.58	380 secondi di esecuzione/240 secondi trascorsi
300	600	2	600 secondi di esecuzione/300 secondi trascorsi

Nella maggior parte dei casi, l'AAS e AAE per una query sono quasi uguali. Tuttavia, poiché gli input per i calcoli sono origini dati diverse, i calcoli spesso variano leggermente.

Dimensioni

Il parametro `db.load` è diverso dagli altri parametri di serie temporali in quanto può essere suddiviso in sottocomponenti detti dimensioni. Le dimensioni possono essere considerate come categorie "slice by" (dividi per) per le diverse caratteristiche del parametro `DBLoad`.

Quando si diagnosticano problemi di prestazioni, le dimensioni seguenti sono spesso le più utili:

Argomenti

- Eventi di attesa
- Prime istruzioni SQL
- Piani

Per un elenco completo delle dimensioni per i motori Amazon RDS, consulta Carico del database suddiviso per dimensioni.

Eventi di attesa

Un evento di attesa fa sì che un'istruzione SQL attenda che si verifichi un evento specifico prima che possa continuare l'esecuzione. Gli eventi di attesa sono una dimensione o una categoria importante per il caricamento del database perché indicano dove il lavoro è impedito.

Ogni sessione attiva è in esecuzione sulla CPU o in attesa. Ad esempio, le sessioni consumano la CPU quando cercano in memoria un buffer, eseguono un calcolo o eseguono codice procedurale. Quando le sessioni non consumano la CPU, potrebbero essere in attesa che un buffer di memoria diventi libero, un file di dati da leggere o un registro in cui scrivere. Maggiore è il tempo in cui una sessione attende le risorse, minore è il tempo in cui viene eseguita sulla CPU.

Quando si sintonizza un database, si tenta spesso di scoprire le risorse che le sessioni sono in attesa. Ad esempio, due o tre eventi di attesa potrebbero rappresentare il 90% del carico DB. Questa misura significa che, in media, le sessioni attive trascorrono la maggior parte del tempo in attesa di un numero limitato di risorse. Se riesci a scoprire la causa di queste attese, puoi provare a fornire una soluzione.

Gli eventi di attesa variano in base al motore database:

- Per informazioni su tutti gli eventi di attesa MariaDB e MySQL, consulta Wait Event Summary Tables nella documentazione di MySQL.

- Per informazioni su tutti gli eventi di attesa PostgreSQL, consulta la pagina relativa al processo di raccolta delle statistiche e alle tabelle degli eventi di attesa nella documentazione di PostgreSQL.
- Per informazioni su tutti gli eventi di attesa Oracle, consulta l'argomento con le descrizioni degli eventi di attesa nella documentazione Oracle.
- Per informazioni su tutti gli eventi di attesa SQL Server, consulta Types of Waits nella documentazione SQL Server.

> **Note**
>
> Per Oracle, i processi in background a volte funzionano senza istruzioni SQL associate. In questi casi, Performance Insights segnala il tipo di processo in background concatenato da due punti e la classe di attesa associata al processo in background. I tipi di processo in background includono `LGWR`, `ARC0`, `PMON`, e così via.
> Ad esempio, quando lo strumento di archiviazione esegue I/O, il report di Performance Insights è simile a `ARC1:System I/O`. A volte manca anche il tipo di processo in background e Performance Insights indica solo la classe di attesa, ad esempio `:System I/O`.

Prime istruzioni SQL

Mentre gli eventi di attesa mostrano i colli di bottiglia, il primo SQL mostra quali query stanno contribuendo maggiormente al caricamento del DB. Ad esempio, molte query potrebbero essere attualmente in esecuzione nel database, ma una singola query potrebbe consumare il 99 percento del carico DB. In questo caso, il carico elevato potrebbe indicare un problema con la query.

Per impostazione predefinita, la console di Performance Insights visualizza le prime query SQL che contribuiscono al caricamento del database. La console mostra anche le statistiche pertinenti per ogni istruzione. Per diagnosticare problemi di prestazioni per un'istruzione specifica, è possibile esaminarne il piano di esecuzione.

Piani

Un piano di esecuzione, chiamato anche semplicemente piano, è una sequenza di passaggi che accedono ai dati. Ad esempio, un piano per unire le tabelle `t1` e `t2` potrebbe scorrere in loop tutte le righe in `t1` e confrontare ogni riga con una riga in `t2`. In un database relazionale, un ottimizzatore è un codice incorporato che determina il piano più efficiente per una query SQL.

Per le istanze DB, Performance Insights raccoglie automaticamente i piani di esecuzione. Per diagnosticare i problemi di prestazioni SQL, esamina i piani acquisiti per le query SQL ad alte risorse. I piani mostrano come il database ha analizzato ed eseguito le query.

Per informazioni su come analizzare il carico del DB utilizzando i piani, consulta:

- Oracle: Analisi dei piani di esecuzione di Oracle tramite il pannello di controllo di Performance Insights
- SQL Server: Analisi dei piani di esecuzione di SQL Server utilizzando il dashboard di Performance Insights

Acquisizione del piano

Ogni cinque minuti, Performance Insights identifica le query che richiedono più risorse e ne registra i piani. Pertanto, non devi raccogliere e gestire manualmente un gran numero di piani. Invece, puoi utilizzare la scheda Top SQL (Prime istruzioni SQL) per concentrarti sui piani per le query più problematiche.

> **Note**
>
> Performance Insights non acquisisce piani per le query il cui testo supera il limite massimo di testo della query raccolte. Per ulteriori informazioni, consulta Accesso a una maggiore quantità di testo SQL nel pannello di controllo di Performance Insights.

Il periodo di conservazione per i piani di esecuzione è lo stesso dei dati di Performance Insights. L'impostazione del periodo di conservazione nel livello gratuito è Default (7 days) (Impostazione predefinita [7 giorni]). Per mantenere i dati sulle prestazioni più a lungo, specifica da 1 a 24 mesi. Per altre informazioni sui periodi di conservazione, consulta Prezzi e conservazione dei dati per Performance Insights.

Query digest

La scheda Top SQL (Prime istruzioni SQL) mostra le query digest per impostazione predefinita. Una query digest di per sé non ha un piano, ma tutte le query che utilizzano valori letterali hanno piani. Ad esempio, una query digest potrebbe includere il testo `WHERE \`email\`=?`. Il digest potrebbe contenere due query, una con il testo `WHERE email=user1@example.com` e un'altra con `WHERE email=user2@example.com`. Ognuna di queste query letterali può includere più piani.

Quando si seleziona una query riassuntiva, la console mostra tutti i piani per le dichiarazioni secondarie del digest selezionato. Pertanto, non devi esaminare tutte le istruzioni figlio per trovare il piano. Potresti vedere piani che non sono inclusi nell'elenco delle prime 10 istruzioni figlio. La console mostra i piani per tutte le query figlio per le quali sono stati raccolti i piani, indipendentemente dal fatto che le query siano tra le prime 10.

CPU massima

Nel dashboard, il grafico di caricamento del database raccoglie, aggrega e visualizza le informazioni sulla sessione. Per verificare se le sessioni attive superano la CPU massima, esaminare la loro relazione con la linea vCPU massima. Performance Insights determina il valore massimo di vCPU in base al numero di core vCPU (CPU virtuale) per l'istanza DB.

Un processo può essere eseguito su una vCPU alla volta. Se il numero di processi supera il numero di vCPUs, i processi vengono messi in coda. Quando la coda aumenta, le prestazioni diminuiscono. Se il carico è spesso sopra la linea vCPU massima e lo stato di attesa primario è CPU, la CPU è sovraccarica. In questo caso, si potrebbero limitare le connessioni all'istanza, ottimizzare le eventuali query SQL con un elevato carico CPU o valutare la possibilità di una classe istanza di maggiori dimensioni. Istanze elevate e costanti di qualsiasi stato di attesa indicano che possono verificarsi colli di bottiglia o problemi di conflitto delle risorse da risolvere. Questo può valere anche se il carico database non supera il valore della riga CPU massima.

Supporto di classe di istanza, regione e motore di database Amazon RDS per Performance Insights

Nella tabella seguente vengono forniti i motori di database Amazon RDS che supportano Performance Insights.

> **Note**
>
> Per Amazon Aurora, vedere Supporto del motore Amazon Aurora database per Performance Insights in Guida per l'utente di Amazon Aurora.

Motore DB Amazon RDS	Versioni motore e regioni supportate	Restrizioni delle classi di istanza
Amazon RDS per MariaDB	Per ulteriori informazioni sulla disponibilità di versioni e regioni di Performance Insights con RDS per MariaDB, consulta Regioni e motori DB supportati per Performance Insights in Amazon RDS.	Performance Insights non è supportato nelle seguenti classi d'istanza: • db.t2.micro • db.t2.small • db.t3.micro • db.t3.small • db.t4g.micro • db.t4g.small
RDS for MySQL	Per ulteriori informazioni sulla disponibilità di versione e regioni di Performance Insights con RDS per MySQL, consulta Regioni e motori DB supportati per Performance Insights in Amazon RDS.	Performance Insights non è supportato nelle seguenti classi d'istanza: • db.t2.micro • db.t2.small • db.t3.micro • db.t3.small • db.t4g.micro • db.t4g.small

Motore DB Amazon RDS	Versioni motore e regioni supportate	Restrizioni delle classi di istanza
Amazon RDS for Microsoft SQL Server	Per ulteriori informazioni sulla disponibilità di versioni e regioni di Performance Insights con RDS per SQL Server, consulta Regioni e motori DB supportati per Performance Insights in Amazon RDS.	N/D
Amazon RDS per PostgreSQL	Per ulteriori informazioni sulla disponibilità di versioni e regioni di Performance Insights con RDS per PostgreSQL, consulta Regioni e motori DB supportati per Performance Insights in Amazon RDS.	N/D
Amazon RDS per Oracle	Per ulteriori informazioni sulla disponibilità di versioni e regioni di Performance Insights con RDS per Oracle, consulta Regioni e motori DB supportati per Performance Insights in Amazon RDS.	N/D

Supporto di classe di istanza, regione e motore di database Amazon RDS per funzionalità Performance Insights

Nella tabella seguente vengono forniti i motori di database Amazon RDS che supportano funzionalità Performance Insights.

Funzionalità	Livello di prezzi	Regioni supportate	Motori del database supportati	Classi di istanza supportate
Statistiche SQL per Performance Insights	Tutti	Tutti	Tutti	Tutti
Analisi dei piani di esecuzione di Oracle tramite il pannello di controllo di Performance Insights	Tutti	Tutti	RDS per Oracle	Tutti
Analisi delle prestazioni del database per un periodo di tempo	Solo livello a pagamento	• Stati Uniti orientali (Ohio) • Stati Uniti orientali (Virginia settentrionale) • Stati Uniti occidentali (California settentrionale) • Stati Uniti occidentali (Oregon) • Asia Pacifico (Mumbai) • Asia Pacifico (Seoul) • Asia Pacifico (Singapore)	RDS per PostgreSQL.	Tutti

Funzionalità	Livello di prezzi	Regioni supportate	Motori del database supportati	Classi di istanza supportate
		• Asia Pacifico (Sydney) • Asia Pacifico (Tokyo) • Canada (Centrale) • Europa (Francoforte) • Europa (Irlanda) • Europe (London) • Europe (Paris) • Europa (Stoccolma)		

Funzionalità	Livello di prezzi	Regioni supportate	Motori del database supportati	Classi di istanza supportate
Visualizzazione dei consigli proattivi di Performance Insights	Solo livello a pagamento	• Stati Uniti orientali (Ohio) • Stati Uniti orientali (Virginia settentrionale) • Stati Uniti occidentali (California settentrionale) • Stati Uniti occidentali (Oregon) • Asia Pacifico (Mumbai) • Asia Pacifico (Seoul) • Asia Pacifico (Singapore) • Asia Pacifico (Sydney) • Asia Pacifico (Tokyo) • Canada (Centrale) • Europa (Francoforte) • Europa (Irlanda)	Tutti	Tutti

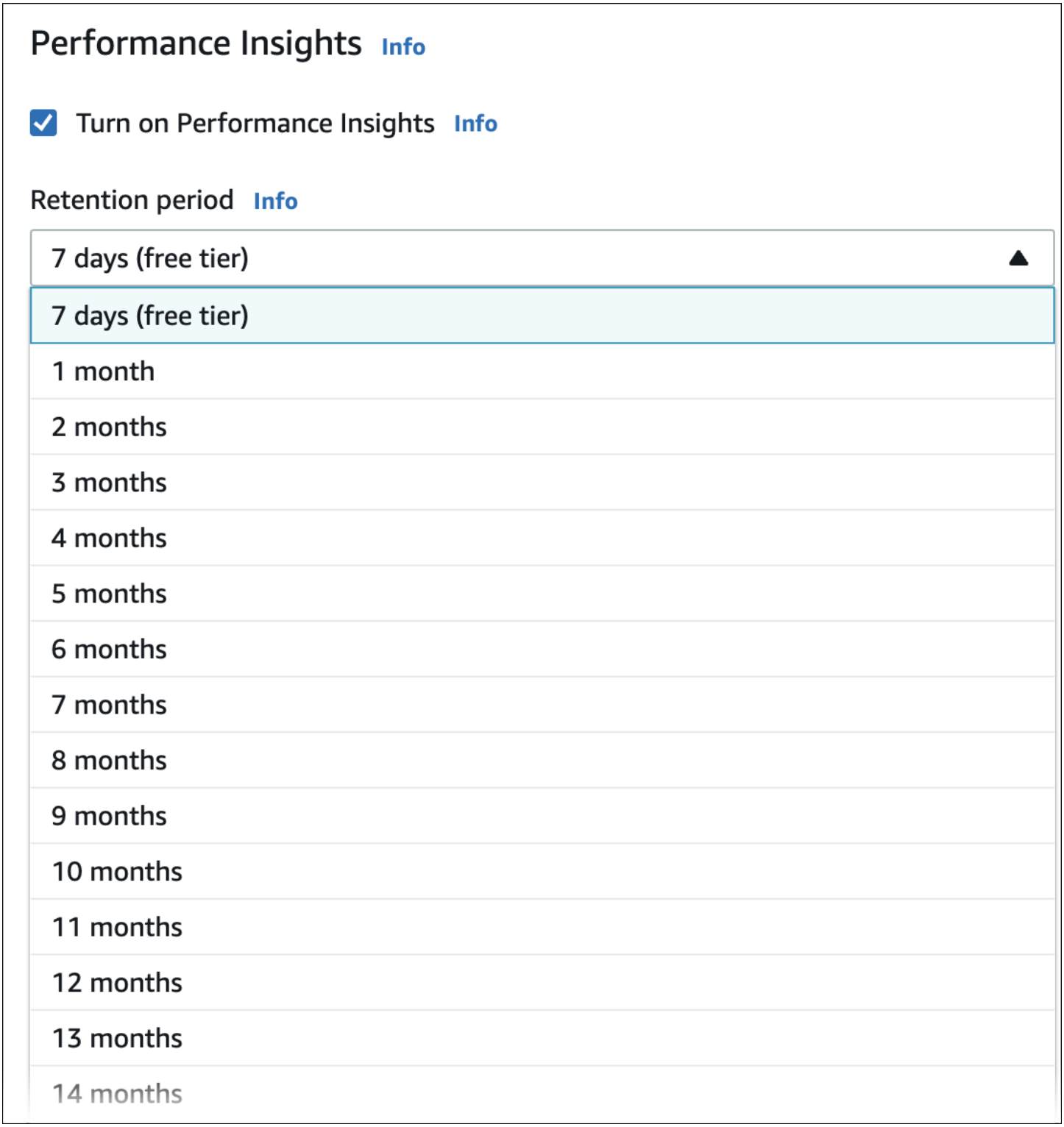
Funzionalità	Livello di prezzi	Regioni supportate	Motori del database supportati	Classi di istanza supportate
		• Europe (London) • Europe (Paris) • Europa (Stoccolma) • Sud America (San Paolo)		

Prezzi e conservazione dei dati per Performance Insights

Per impostazione predefinita, Performance Insights offre un piano gratuito che include 7 giorni di cronologia dei dati sulle prestazioni e 1 milione di richieste API al mese. Puoi anche acquistare periodi di conservazione più lunghi. Per informazioni sui prezzi, consulta Prezzi di Performance Insights.

Nella console RDS, puoi scegliere uno dei seguenti periodi di conservazione per i dati di Performance Insights:

- Default (7 giorni)
- ***n*** mesi, dove ***n*** è un numero compreso tra 1 e 24



Per informazioni su come impostare un periodo di conservazione utilizzando la AWS CLI, consulta AWS CLI.

Attivazione e disattivazione di Performance Insights

Puoi attivare Performance Insights per l'istanza database o il cluster di database Multi-AZ al momento della creazione. Se necessario, puoi disattivarlo in un secondo momento. L'attivazione e la disattivazione di Performance Insights non determina tempi di inattività, riavvio o failover.

> **Note**
>
> Performance Schema è uno strumento di prestazioni opzionale utilizzato da Amazon RDS for MariaDB o MySQL. Se si attiva o disattiva Performance Schema, è necessario riavviare il sistema. Se si attiva o disattiva Performance Insights, tuttavia, non è necessario riavviare. Per ulteriori informazioni, consulta Abilitazione di Performance Schema per Performance Insights su Amazon RDS for MariaDB o MySQL.

L'agente Performance Insights consuma CPU e memoria limitate sull'host DB. Quando il carico del DB è elevato, l'agente limita l'impatto sulle prestazioni raccogliendo i dati meno frequentemente.

Console

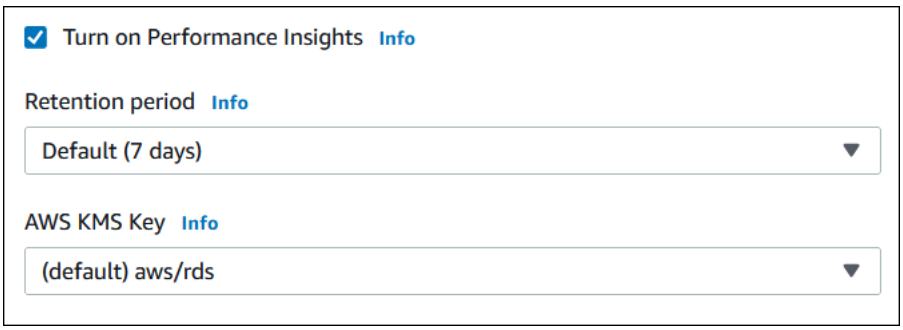
Nella console è possibile attivare o disattivare Performance Insights quando si crea o si modifica un'istanza database o un cluster di database Multi-AZ.

Attivazione o disattivazione di Performance Insights durante la creazione di istanze database o cluster di database Multi-AZ

Quando crei una nuova istanza database o nuovo un cluster di database Multi-AZ, puoi abilitare Performance Insights scegliendo Enable Performance Insights (Abilita Performance Insights) nella sezione Performance Insights. Scegliere Disable Performance Insights (Disabilita Performance Insights). Per ulteriori informazioni, consulta gli argomenti seguenti:

- Per creare un'istanza database, seguire le istruzioni per il proprio motore database in Creazione di un'istanza database Amazon RDS.
- Per creare un cluster di database Multi-AZ, seguire le istruzioni in Creazione di un cluster di database Multi-AZ.

L'immagine seguente mostra la sezione Performance Insights.



Quando selezioni Abilita Performance Insights, sono disponibili le opzioni seguenti:

- Retention (Conservazione) – Quantità di tempo per cui conservare i dati di Performance Insights. L'impostazione del periodo di conservazione nel livello gratuito è Default (7 days) (Impostazione predefinita (7 giorni)). Per mantenere i dati sulle prestazioni più a lungo, specifica da 1 a 24 mesi. Per altre informazioni sui periodi di conservazione, consulta Prezzi e conservazione dei dati per Performance Insights.
- AWS KMS key: specificare la AWS KMS key. Performance Insights crittografa tutti i dati potenzialmente sensibili con la chiave KMS. I dati vengono crittografati mentre sono in transito o inattivi. Per ulteriori informazioni, consulta Configurazione di una policy AWS KMS per Performance Insights.

Attivazione o disattivazione di Performance Insights durante la modifica di un'istanza database o nel cluster di database Multi-AZ

Nella console puoi modificare un'istanza database o nel cluster di database Multi-AZ per attivare o disattivare Performance Insights.

Per attivare o disattivare Performance Insights per un'istanza database o un cluster di database Multi-AZ usando la console

1. Accedere alla AWS Management Console e aprire la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Scegli Databases (Database).
3. Scegliere un'istanza database o un cluster di database Multi-AZ e scegliere Modify (Modifica).
4. Nella sezione Performance Insights scegliere Attiva Performance Insights o Disattiva Performance Insights.

   Quando selezioni Abilita Performance Insights, sono disponibili le opzioni seguenti:

- Retention (Conservazione) – Quantità di tempo per cui conservare i dati di Performance Insights. L'impostazione del periodo di conservazione nel livello gratuito è Default (7 days) (Impostazione predefinita (7 giorni)). Per mantenere i dati sulle prestazioni più a lungo, specifica da 1 a 24 mesi. Per altre informazioni sui periodi di conservazione, consulta Prezzi e conservazione dei dati per Performance Insights.
- AWS KMS key: specificare la chiave KMS. Performance Insights crittografa tutti i dati potenzialmente sensibili con la chiave KMS. I dati vengono crittografati mentre sono in transito o inattivi. Per ulteriori informazioni, consulta Crittografia delle risorse Amazon RDS.

5. Scegli Continue (Continua).
6. In Scheduling of Modifications (Pianificazione delle modifiche), scegli Apply immediately (Applica immediatamente). Se scegli Apply (Applica) durante la prossima finestra di manutenzione pianificata, l'istanza ignora questa impostazione e attiva immediatamente Performance Insights.
7. Scegli Modify instance (Modifica istanza).

AWS CLI

Quando usi il create-db-instanceAWS CLIcomando, attiva Performance Insights `--enable-performance-insights` specificando. Oppure disabilita Performance Insights specificando `--no-enable-performance-insights`.

Puoi anche specificare questi valori utilizzando i seguenti comandi AWS CLI:

- create-db-instance-read-replica
- modify-db-instance
- restore-db-instance-from-s3
- create-db-cluster(cluster DB Multi-AZ)
- modify-db-cluster(cluster DB Multi-AZ)

La seguente procedura descrive come attivare o disattivare Performance Insights per un'istanza database utilizzando la AWS CLI.

Per attivare o disattivare Performance Insights per un'istanza database utilizzando la AWS CLI

- Chiama il modify-db-instanceAWS CLIcomando e fornisci i seguenti valori:
  - `--db-instance-identifier` - Il nome dell'istanza database.

- `--enable-performance-insights` per attivare o `--no-enable-performance-insights` per disattivare

Il seguente esempio attiva Performance Insights per `sample-db-instance`.

Per LinuxmacOS, oUnix:

```
aws rds modify-db-instance \
    --db-instance-identifier sample-db-instance \
    --enable-performance-insights
```

Per Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier sample-db-instance ^
    --enable-performance-insights
```

Quando attivi Performance Insights nella CLI, puoi specificare, in via facoltativa, il periodo di tempo, in giorni, per cui mantenere i dati di Performance Insights con l'opzione `--performance-insights-retention-period`. Puoi specificare 7, *`mese`* * 31 (dove *`mese`* è un numero compreso tra 1 e 23), o `731`. Ad esempio, se desideri mantenere i dati sulle prestazioni per 3 mesi, specifica `93`, che è 3 * 31. L'impostazione di default è 7 giorni. Per altre informazioni sui periodi di conservazione, consulta Prezzi e conservazione dei dati per Performance Insights.

Il seguente esempio attiva Performance Insights per `sample-db-instance` e specifica che i dati Performance Insights sono mantenuti per 93 giorni (3 mesi).

Per LinuxmacOS, oUnix:

```
aws rds modify-db-instance \
    --db-instance-identifier sample-db-instance \
    --enable-performance-insights \
    --performance-insights-retention-period 93
```

Per Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier sample-db-instance ^
```

```
    --enable-performance-insights ^
    --performance-insights-retention-period 93
```

Se il periodo di conservazione specificato, ad esempio 94 giorni, non è un valore valido, RDS genera un errore.

```
An error occurred (InvalidParameterValue) when calling the CreateDBInstance operation:
Invalid Performance Insights retention period. Valid values are: [7, 31, 62, 93, 124,
 155, 186, 217,
248, 279, 310, 341, 372, 403, 434, 465, 496, 527, 558, 589, 620, 651, 682, 713, 731]
```

API RDS

Quando crei una nuova istanza database tramite l'operazione CreateDBInstance dell'API Amazon RDS, attivi Performance Insights impostando `EnablePerformanceInsights` su `True`. Per disabilitare Performance Insights, imposta `EnablePerformanceInsights` su `False`.

Puoi anche specificare il valore `EnablePerformanceInsights` utilizzando le seguenti operazioni API:

- ModifyDBInstance
- Creato B InstanceReadReplica
- Ripristina DB S3 InstanceFrom
- CreateDBCluster (Cluster di database Multi-AZ)
- ModifyDBCluster (Cluster di database Multi-AZ)

Quando si attiva Performance Insights, è possibile specificare, in via facoltativa, il periodo di tempo, in giorni, per cui conservare i dati Performance Insights con il parametro `PerformanceInsightsRetentionPeriod`. Puoi specificare 7, *`mese`* * 31 (dove *`mese`* è un numero compreso tra 1 e 23), o `731`. Ad esempio, se desideri mantenere i dati sulle prestazioni per 3 mesi, specifica 93, che è 3 * 31. L'impostazione di default è 7 giorni. Per altre informazioni sui periodi di conservazione, consulta Prezzi e conservazione dei dati per Performance Insights.

Abilitazione di Performance Schema per Performance Insights su Amazon RDS for MariaDB o MySQL

Performance Schema è una funzionalità facoltativa per il monitoraggio delle prestazioni di runtime di Amazon RDS for MariaDB o MySQL a un dettaglio di basso livello. Performance Schema è progettato

per avere un impatto minimo sulle prestazioni del database. Performance Insights è una funzionalità separata che puoi utilizzare con o senza Performance Schema.

Argomenti

Panoramica dello schema di prestazioni

Performance Schema monitora gli eventi nei database MariaDB e MySQL. Un evento è un'azione del server di database che consuma tempo ed è stata strumentata in modo che possano essere raccolte le informazioni di temporizzazione. Ecco alcuni esempi di eventi:

- Chiamate di funzione
- Attendi il sistema operativo
- Fasi dell'esecuzione SQL
- Gruppi di istruzioni SQL

Il motore di archiviazione `PERFORMANCE_SCHEMA` è un meccanismo per l'implementazione della funzionalità Performance Schema. Questo motore raccoglie i dati degli eventi utilizzando la strumentazione nel codice sorgente del database. Il motore memorizza gli eventi raccolti nelle tabelle in memoria nel database `performance_schema`. È possibile interrogare `performance_schema` proprio come puoi interrogare qualsiasi altra tabella. Per ulteriori informazioni, consulta Performance Schema di MySQL nel Manuale di riferimento di MySQL.

Performance Insights e lo schema di prestazioni

Performance Insights e Performance Schema sono funzionalità separate, ma sono connesse. Il comportamento di Performance Insights per Amazon RDS per MariaDB o MySQL varia a seconda che lo schema di prestazioni sia attivato e, in questo caso, se Performance Insights gestisce automaticamente lo schema di prestazioni. Il comportamento viene descritto nella tabella seguente.

Schema di prestazioni attivato	Modalità di gestione di Performance Insights	Comportamento di Performance Insights
Sì	Automatica	• Raccoglie informazioni dettagliate di monitoraggio a basso livello • Raccoglie le metriche di sessione attive ogni secondo • Visualizza il carico del database classificato in base a eventi di attesa dettagliati, che è possibile utilizzare per identificare i colli di bottiglia
Sì	Manuale	• Raccoglie gli eventi di attesa e le metriche per SQL • Raccoglie le metriche di sessione attive ogni cinque secondi anziché ogni secondo • Segnala gli stati utente, ad esempio l'inserimento e l'invio, che non consentono di identificare i colli di bottiglia
No	N/D	• Non raccoglie eventi di attesa, metriche per SQL o altre informazioni dettagliate di monitoraggio di basso livello • Raccoglie le metriche di sessione attive ogni cinque secondi anziché ogni secondo • Segnala gli stati utente, ad esempio l'inserimento e l'invio, che non consentono di identificare i colli di bottiglia

Gestione automatica di Performance Schema da parte di Performance Insights

Quando crei un'istanza database Amazon RDS for MariaDB o MySQL con Performance Insights abilitato, anche la funzionalità Performance Schema viene abilitata. In questo caso, Performance Insights gestisce automaticamente i parametri di Performance Schema. Questa è la configurazione consigliata.

> **Note**
>
> La gestione automatica dello schema di prestazioni non è supportata per la classe di istanza t4g.medium.

Per la gestione automatica di Performance Schema, devono verificarsi le seguenti condizioni:

- Il parametro `performance_schema` è impostato su `0`.
- Source (Origine) è impostato su `system`, che è il valore predefinito.

Se modifichi il manualmente il parametro `performance_schema` e in seguito desideri ripristinare la gestione automatica, consulta Configurazione di Performance Schema per la gestione automatica.

> **Important**
>
> Quando Performance Insights abilita Performance Schema, non modifica i valori del gruppo di parametri. Tuttavia, i valori vengono modificati sulle istanze database in esecuzione. L'unico modo per vedere i valori modificati è eseguire il comando `SHOW GLOBAL VARIABLES`.

Effetto di un riavvio su Performance Schema

Performance Insights e Performance Schema differiscono per i requisiti relativi al riavvio delle istanze DB:

Performance Schema

Per attivare o disattivare questa funzionalità, è necessario riavviare l'istanza database.

Approfondimenti sulle prestazioni

Per attivare o disattivare questa funzionalità, non è necessario riavviare l'istanza database.

Se Performance Schema non è attualmente attivato e si attiva Performance Insights senza riavviare l'istanza database, Performance Schema non verrà attivato.

Determinazione della gestione di Performance Schema da parte di Performance Insights

Per scoprire se Performance Insights gestisce Performance Schema per i principali motori versioni 5.6, 5.7 e 8.0, consulta la tabella riportata di seguito.

Impostazione del parametro performance_schema	Impostazione della colonna Source	Performance Insights sta gestendo Performance Schema?
`0`	`system`	Sì
`0` o `1`	`user`	No

Per determinare se Performance Insights sta gestendo automaticamente Performance Schema

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Scegliere Gruppi di parametri.
3. Selezionare il nome del gruppo di parametri per l'istanza database.
4. Inserire **`performance_schema`** nella barra di ricerca.
5. Controlla se Source (Fonte) è il valore di default di sistema e Values (Valori) è impostato a 0. In tal caso, Performance Insights gestisce automaticamente Performance Schema. In caso contrario, Performance Insights non sta gestendo automaticamente Performance Schema.

Configurazione di Performance Schema per la gestione automatica

Supponiamo che Performance Insights sia attivato per l'istanza database o per il cluster di database Multi-AZ ma al momento non stia gestendo Performance Schema. Se desideri consentire a Performance Insights di gestire automaticamente Performance Schema, completa la procedura seguente.

Configurazione di Performance Schema per la gestione automatica

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Scegliere Gruppi di parametri.
3. Selezionare il nome del gruppo di parametri per l'istanza database o il cluster di database Multi-AZ.
4. Inserire **`performance_schema`** nella barra di ricerca.
5. Selezionare il parametro `performance_schema`.
6. Scegli Edit parameters (Modifica parametri).
7. Selezionare il parametro `performance_schema`.
8. Nello stato Valori, scegliere 0.
9. Seleziona Salvataggio delle modifiche.
10. Riavviare l'istanza database o il cluster di database Multi-AZ.

> **Important**
>
> Ogni volta che si abilita o disabilita Performance Schema, è necessario riavviare l'istanza database o il cluster di database Multi-AZ.

Per ulteriori informazioni sulla modifica dei parametri di un'istanza, consulta Modifica di parametri in un gruppo di parametri del database. Per ulteriori informazioni sulle pagine del pannello di controllo, consulta Per analizzare il parametro utilizzando il pannello di controllo di Performance Insights. Per ulteriori informazioni su Performance Schema di MySQL, consulta il Manuale di riferimento di MySQL 8.0.

Configurazione delle policy di accesso per Performance Insights

Per accedere a Performance Insights, un principale deve disporre delle autorizzazioni appropriate di AWS Identity and Access Management (IAM). Puoi garantire l'accesso secondo le seguenti modalità:

- Collega la policy gestita `AmazonRDSPerformanceInsightsReadOnly` a un set di autorizzazioni o a un ruolo per accedere a tutte le operazioni di sola lettura dell'API di Performance Insights.
- Collega la policy gestita `AmazonRDSPerformanceInsightsFullAccess` a un set di autorizzazioni o a un ruolo per accedere a tutte le operazioni dell'API di Performance Insights.
- Crea una policy IAM personalizzata e collegala a un set di autorizzazioni o un ruolo.

Se hai specificato una chiave gestita dal cliente quando hai attivato Performance Insights, assicurati che gli utenti del tuo account dispongano delle `kms:GenerateDataKey` autorizzazioni `kms:Decrypt` e su AWS KMS key

Allegare la AmazonRDSPerformanceInsightsReadOnly policy a un preside IAM

`AmazonRDSPerformanceInsightsReadOnly`è una policy AWS gestita che garantisce l'accesso a tutte le operazioni di sola lettura dell'API Amazon RDS Performance Insights.

Se si collega `AmazonRDSPerformanceInsightsReadOnly` a un set di autorizzazioni o un ruolo, il destinatario può utilizzare Performance Insights insieme ad altre funzionalità della console.

Per ulteriori informazioni, consulta AWS politica gestita: AmazonRDS PerformanceInsightsReadOnly.

Allegare la policy a un principio IAM AmazonRDSPerformanceInsightsFullAccess

`AmazonRDSPerformanceInsightsFullAccess`è una policy AWS gestita che garantisce l'accesso a tutte le operazioni dell'API Amazon RDS Performance Insights.

Se si collega `AmazonRDSPerformanceInsightsFullAccess` a un set di autorizzazioni o un ruolo, il destinatario può utilizzare Performance Insights insieme ad altre funzionalità della console.

Per ulteriori informazioni, consulta AWS politica gestita: AmazonRDS PerformanceInsightsFullAccess.

Creazione di una policy IAM personalizzata per Performance Insights

Per gli utenti che non dispongono della `AmazonRDSPerformanceInsightsFullAccess` policy `AmazonRDSPerformanceInsightsReadOnly` o, puoi concedere l'accesso a Performance Insights

creando o modificando una policy IAM gestita dall'utente. Quando si collega la policy a un set di autorizzazioni o un ruolo, il destinatario può utilizzare Performance Insights.

Per creare una policy personalizzata

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione, selezionare Policies (Policy).
3. Scegli Create Policy (Crea policy).
4. Nella pagina Crea policy, scegli l'opzione JSON.
5. Copia e incolla il testo fornito nella sezione del documento sulla policy JSON della AWS Managed Policy Reference Guide per la AmazonRDSPerformanceInsightsReadOnlynostra policy. AmazonRDSPerformanceInsightsFullAccess
6. Scegliere Review policy (Esamina policy).
7. Specifica un nome per la policy e, facoltativamente, una descrizione e quindi scegli Create policy (Crea policy).

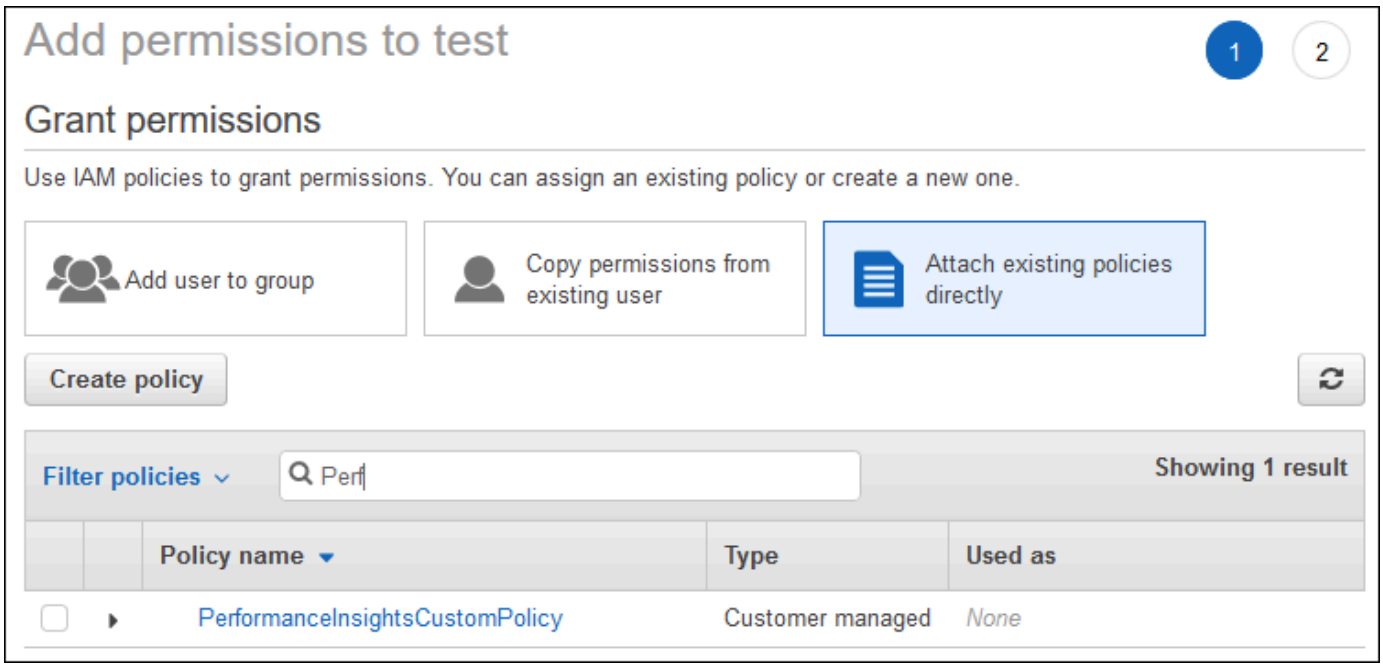
Ora è possibile collegare la policy a un set di autorizzazioni o un ruolo. La seguente procedura presuppone che si disponga già di un utente disponibile allo scopo.

Per collegare la policy a un utente

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione, seleziona Users (Utenti).
3. Seleziona un utente esistente dall'elenco.

> **Important**
>
> Per utilizzare Performance Insights, l'utente deve avere accesso a Amazon RDS nonché alla policy personalizzata. Ad esempio, la policy predefinita `AmazonRDSPerformanceInsightsReadOnly` concede l'accesso in sola lettura ad Amazon RDS. Per ulteriori informazioni, consulta Gestione dell'accesso con policy.

4. Nella pagina Summary (Riepilogo), scegli Add permissions (Aggiungi autorizzazioni).
5. Scegli Attach existing policies directly (Collega direttamente le policy esistenti). Per la ricerca, digita i primi caratteri del nome della policy, come mostrato nell'immagine seguente.



6. Scegli la policy e quindi seleziona Next: Review (Successivo: Rivedi).
7. Scegli Add Permissions (Aggiungi autorizzazioni).

Configurazione di una policy AWS KMS per Performance Insights

Performance Insights utilizza an AWS KMS key per crittografare i dati sensibili. Quando abiliti Performance Insights mediante l'API o la console, sono disponibili le seguenti opzioni:

- Scegli l'impostazione predefinita Chiave gestita da AWS.

  Amazon RDS lo utilizza Chiave gestita da AWS per la tua nuova istanza DB. Amazon RDS crea una Chiave gestita da AWS per il tuo Account AWS. Il tuo Account AWS ha un Amazon RDS diverso Chiave gestita da AWS per ognuno Regione AWS.

- Scegli una chiave gestita dal cliente.

  Se si specifica una chiave gestita dal cliente, gli utenti dell'account che chiamano l'API Performance Insights necessitano delle autorizzazioni `kms:Decrypt` e `kms:GenerateDataKey` per la chiave KMS. È possibile configurare queste autorizzazioni mediante le policy IAM. Tuttavia, è consigliabile gestire queste autorizzazioni mediante la policy della chiave KMS. Per ulteriori informazioni, consulta Policy delle chiavi in AWS KMS nella Guida per gli sviluppatori di AWS Key Management Service .

Example

Il seguente esempio mostra come aggiungere istruzioni alla policy della chiave KMS. Queste istruzioni consentono l'accesso a Performance Insights. A seconda della modalità di utilizzare la chiave KMS, potrebbe essere necessario modificare alcune restrizioni. Prima di aggiungere istruzioni alle policy, ai criteri, rimuovi tutti i commenti.

```
{
"Version" : "2012-10-17",
 "Id" : "your-policy",
 "Statement" : [ {
    //This represents a statement that currently exists in your policy.
 }
 ....,
 //Starting here, add new statement to your policy for Performance Insights.
 //We recommend that you add one new statement for every RDS instance
{
    "Sid" : "Allow viewing RDS Performance Insights",
    "Effect": "Allow",
    "Principal": {
        "AWS": [
            //One or more principals allowed to access Performance Insights
            "arn:aws:iam::444455556666:role/Role1"
        ]
    },
    "Action": [
        "kms:Decrypt",
        "kms:GenerateDataKey"
    ],
    "Resource": "*",
    "Condition" : {
        "StringEquals" : {
            //Restrict access to only RDS APIs (including Performance Insights).
            //Replace region with your AWS Region.
            //For example, specify us-west-2.
            "kms:ViaService" : "rds.region.amazonaws.com"
        },
        "ForAnyValue:StringEquals": {
            //Restrict access to only data encrypted by Performance Insights.
            "kms:EncryptionContext:aws:pi:service": "rds",
            "kms:EncryptionContext:service": "pi",

            //Restrict access to a specific RDS instance.
```

```
            //The value is a DbiResourceId.
           "kms:EncryptionContext:aws:rds:db-id": "db-AAAAABBBBBCCCCDDDDDEEEEE"
        }
    }
}
```

In che modo Performance Insights utilizza la chiave gestita dal AWS KMS cliente

La funzionalità Approfondimenti sulle prestazioni utilizza una chiave gestita dal cliente per crittografare i dati sensibili. Quando attivi la funzionalità Approfondimenti sulle prestazioni, puoi specificare una chiave AWS KMS tramite l'API. La funzionalità Approfondimenti sulle prestazioni crea autorizzazioni KMS su questa chiave. Utilizza la chiave ed esegue le operazioni necessarie per elaborare i dati sensibili. I dati sensibili includono campi come utente, database, applicazione e testo di query SQL. La funzionalità Approfondimenti sulle prestazioni garantisce che i dati rimangano crittografati mentre sono sia in transito che inattivi..

Come funziona Performance Insights con IAM AWS KMS

IAM concede autorizzazioni ad API specifiche. La funzionalità Approfondimenti sulle prestazioni dispone delle seguenti API pubbliche, che puoi limitare utilizzando le policy IAM:

- `DescribeDimensionKeys`
- `GetDimensionKeyDetails`
- `GetResourceMetadata`
- `GetResourceMetrics`
- `ListAvailableResourceDimensions`
- `ListAvailableResourceMetrics`

Puoi utilizzare le seguenti richieste API per recuperare i dati sensibili.

- `DescribeDimensionKeys`
- `GetDimensionKeyDetails`
- `GetResourceMetrics`

Quando utilizzi l'API per recuperare i dati sensibili, la funzionalità Approfondimenti sulle prestazioni usa le credenziali del chiamante. Questo controllo garantisce che l'accesso ai dati sensibili sia limitato a coloro che hanno accesso alla chiave KMS.

Quando si chiamano queste API, sono necessarie le autorizzazioni per chiamare l'API tramite la policy IAM e le autorizzazioni per richiamare l'`kms:decrypt`azione tramite la policy chiave. AWS KMS

L'API `GetResourceMetrics` può restituire dati sensibili e non sensibili. I parametri della richiesta determinano se la risposta deve includere dati sensibili. L'API restituisce dati sensibili quando la richiesta include una dimensione sensibile nei parametri del filtro o nei parametri di raggruppamento.

Per ulteriori informazioni sulle dimensioni che puoi utilizzare con l'API, consulta`GetResourceMetrics`. DimensionGroup

Example Esempi

L'esempio seguente richiede i dati sensibili per il gruppo `db.user`:

```
POST / HTTP/1.1
Host: <Hostname>
Accept-Encoding: identity
X-Amz-Target: PerformanceInsightsv20180227.GetResourceMetrics
Content-Type: application/x-amz-json-1.1
User-Agent: <UserAgentString>
X-Amz-Date: <Date>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=<Headers>,
 Signature=<Signature>
Content-Length: <PayloadSizeBytes>
{
  "ServiceType": "RDS",
  "Identifier": "db-ABC1DEFGHIJKL2MNOPQRSTUV3W",
  "MetricQueries": [
    {
      "Metric": "db.load.avg",
      "GroupBy": {
        "Group": "db.user",
        "Limit": 2
      }
    }
  ],
  "StartTime": 1693872000,
  "EndTime": 1694044800,
  "PeriodInSeconds": 86400
}
```

Example

L'esempio seguente richiede i dati non sensibili per la metrica `db.load.avg`:

```
POST / HTTP/1.1
Host: <Hostname>
Accept-Encoding: identity
X-Amz-Target: PerformanceInsightsv20180227.GetResourceMetrics
Content-Type: application/x-amz-json-1.1
User-Agent: <UserAgentString>
X-Amz-Date: <Date>
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=<Headers>,
 Signature=<Signature>
Content-Length: <PayloadSizeBytes>
{
    "ServiceType": "RDS",
    "Identifier": "db-ABC1DEFGHIJKL2MNOPQRSTUV3W",
    "MetricQueries": [
        {
            "Metric": "db.load.avg"
        }
    ],
    "StartTime": 1693872000,
    "EndTime": 1694044800,
    "PeriodInSeconds": 86400
}
```

Concessione di un accesso granulare a Performance Insights

Il controllo granulare degli accessi offre metodi aggiuntivi per controllare l'accesso a Performance Insights. Questo controllo di accesso può consentire o negare l'accesso alle singole dimensioni e alle azioni di `GetResourceMetrics` `GetDimensionKeyDetails` Performance Insights. `DescribeDimensionKeys` Per utilizzare un accesso granulare, specifica le dimensioni nella policy IAM utilizzando le chiavi di condizione. La valutazione dell'accesso segue la logica di valutazione delle policy IAM. Per ulteriori informazioni, consulta Logica di valutazione delle politiche nella Guida per l'utente IAM. Se l'istruzione politica IAM non specifica alcuna dimensione, l'istruzione controlla l'accesso a tutte le dimensioni per l'azione specificata. Per l'elenco delle dimensioni disponibili, consulta DimensionGroup.

Per scoprire a quali dimensioni le tue credenziali sono autorizzate ad accedere, utilizza il `AuthorizedActions` parametro in `ListAvailableResourceDimensions` e specifica l'azione. I valori consentiti per `AuthorizedActions` sono i seguenti:

- `GetResourceMetrics`
- `DescribeDimensionKeys`
- `GetDimensionKeyDetails`

Ad esempio, se si specifica `GetResourceMetrics` il `AuthorizedActions` parametro, `ListAvailableResourceDimensions` restituisce l'elenco delle dimensioni a cui l'`GetResourceMetrics`azione è autorizzata ad accedere. Se specificate più azioni nel `AuthorizedActions` parametro, `ListAvailableResourceDimensions` restituisce un'intersezione di dimensioni a cui tali azioni sono autorizzate ad accedere.

Example

L'esempio seguente fornisce l'accesso alle dimensioni `GetResourceMetrics` e alle `DescribeDimensionKeys` azioni specificate.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowToDiscoverDimensions",
            "Effect": "Allow",
            "Action": [
                "pi:ListAvailableResourceDimensions"
            ],
            "Resource": [
                "arn:aws:pi:us-east-1:123456789012:metrics/rds/db-
ABC1DEFGHIJKL2MNOPQRSTUV3W"
            ]
        },
        {
            "Sid": "SingleAllow",
            "Effect": "Allow",
            "Action": [
                "pi:GetResourceMetrics",
                "pi:DescribeDimensionKeys"
            ],
            "Resource": [
```

```
                "arn:aws:pi:us-east-1:123456789012:metrics/rds/db-
ABC1DEFGHIJKL2MNOPQRSTUV3W"
            ],
            "Condition": {
                "ForAllValues:StringEquals": {
                    // only these dimensions are allowed. Dimensions not included in
                    // a policy with "Allow" effect will be denied
                    "pi:Dimensions": [
                        "db.sql_tokenized.id",
                        "db.sql_tokenized.statement"
                    ]
                }
            }
        }


    ]
}
```

Di seguito è riportata la risposta per la dimensione richiesta:

```
 // ListAvailableResourceDimensions API
// Request
{
    "ServiceType": "RDS",
    "Identifier": "db-ABC1DEFGHIJKL2MNOPQRSTUV3W",
    "Metrics": [ "db.load" ],
    "AuthorizedActions": ["DescribeDimensionKeys"]
}

// Response
{
    "MetricDimensions": [ {
        "Metric": "db.load",
        "Groups": [
            {
                "Group": "db.sql_tokenized",
                "Dimensions": [
                    { "Identifier": "db.sql_tokenized.id" },
                  //  { "Identifier": "db.sql_tokenized.db_id" }, // not included
 because not allows in the IAM Policy
```

```
                    { "Identifier": "db.sql_tokenized.statement" }
                ]
            }

        ] }
    ]
}
```

L'esempio seguente specifica un accesso consentito e due negato l'accesso per le dimensioni.

```
{
    "Version": "2012-10-17",
    "Statement": [
          {
            "Sid": "AllowToDiscoverDimensions",
            "Effect": "Allow",
            "Action": [
                "pi:ListAvailableResourceDimensions"
            ],
            "Resource": [
                "arn:aws:pi:us-east-1:123456789012:metrics/rds/db-
ABC1DEFGHIJKL2MNOPQRSTUV3W"
            ]
          },

          {
            "Sid": "O01AllowAllWithoutSpecifyingDimensions",
            "Effect": "Allow",
            "Action": [
                "pi:GetResourceMetrics",
                "pi:DescribeDimensionKeys"
            ],
            "Resource": [
                "arn:aws:pi:us-east-1:123456789012:metrics/rds/db-
ABC1DEFGHIJKL2MNOPQRSTUV3W"
            ]
        },

        {
            "Sid": "O01DenyAppDimensionForAll",
            "Effect": "Deny",
            "Action": [
                "pi:GetResourceMetrics",
```

```
                "pi:DescribeDimensionKeys"
            ],
            "Resource": [
                "arn:aws:pi:us-east-1:123456789012:metrics/rds/db-
ABC1DEFGHIJKL2MNOPQRSTUV3W"
            ],
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "pi:Dimensions": [
                        "db.application.name"
                    ]
                }
            }
        },

        {
            "Sid": "O01DenySQLForGetResourceMetrics",
            "Effect": "Deny",
            "Action": [
                "pi:GetResourceMetrics"
            ],
            "Resource": [
                "arn:aws:pi:us-east-1:123456789012:metrics/rds/db-
ABC1DEFGHIJKL2MNOPQRSTUV3W"
            ],
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "pi:Dimensions": [
                        "db.sql_tokenized.statement"
                    ]
                }
            }
        }
    ]
}
```

Di seguito sono riportate le risposte per le dimensioni richieste:

```
   // ListAvailableResourceDimensions API
// Request
{
```

```
    "ServiceType": "RDS",
    "Identifier": "db-ABC1DEFGHIJKL2MNOPQRSTUV3W",
    "Metrics": [ "db.load" ],
    "AuthorizedActions": ["GetResourceMetrics"]
}

// Response
{
    "MetricDimensions": [ {
        "Metric": "db.load",
        "Groups": [
            {
                "Group": "db.application",
                "Dimensions": [

                  // removed from response because denied by the IAM Policy
                  //  { "Identifier": "db.application.name" }
                ]
            },
            {
                "Group": "db.sql_tokenized",
                "Dimensions": [
                    { "Identifier": "db.sql_tokenized.id" },
                    { "Identifier": "db.sql_tokenized.db_id" },

                  // removed from response because denied by the IAM Policy
                  //  { "Identifier": "db.sql_tokenized.statement" }
                ]
            },
            ...
        ] }
    ]
}
```

```
// ListAvailableResourceDimensions API
// Request
{
    "ServiceType": "RDS",
    "Identifier": "db-ABC1DEFGHIJKL2MNOPQRSTUV3W",
    "Metrics": [ "db.load" ],
    "AuthorizedActions": ["DescribeDimensionKeys"]
}
```

```
// Response
{
    "MetricDimensions": [ {
        "Metric": "db.load",
        "Groups": [
            {
                "Group": "db.application",
                "Dimensions": [
                  // removed from response because denied by the IAM Policy
                  //  { "Identifier": "db.application.name" }
                ]
            },
            {
                "Group": "db.sql_tokenized",
                "Dimensions": [
                    { "Identifier": "db.sql_tokenized.id" },
                    { "Identifier": "db.sql_tokenized.db_id" },

                  // allowed for DescribeDimensionKeys because our IAM Policy
                  // denies it only for GetResourceMetrics
                    { "Identifier": "db.sql_tokenized.statement" }
                ]
            },
            ...
        ] }
    ]
}
```

Per analizzare il parametro utilizzando il pannello di controllo di Performance Insights

Il pannello di controllo di Performance Insights contiene informazioni sulle performance del database, per consentire di analizzare e risolvere i problemi di performance. Nella pagina principale del pannello di controllo è possibile visualizzare le informazioni relative al carico del database. Puoi "dividere" il carico del database per dimensioni come eventi di attesa o SQL.

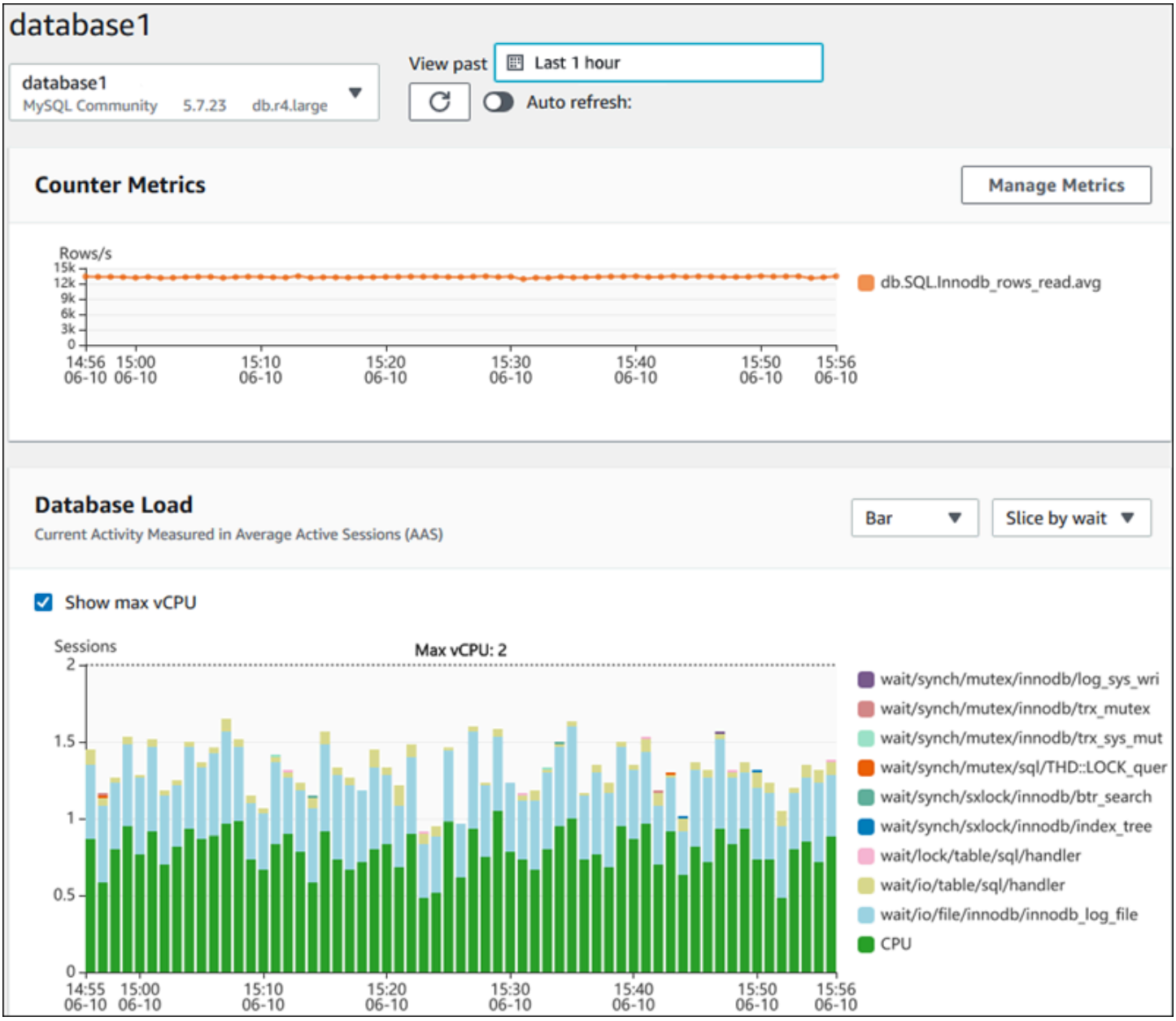
Pannello di controllo di Performance Insights

- Panoramica del pannello di controllo di Performance Insights
- Accesso al pannello di controllo di Performance Insights
- Analisi del carico del database per eventi di attesa

- Analisi delle prestazioni del database per un periodo di tempo
- Analisi delle query all'interno del pannello di controllo di Performance Insights
- Analisi del carico principale di Oracle PDB
- Analisi dei piani di esecuzione utilizzando la dashboard di Performance Insights

Panoramica del pannello di controllo di Performance Insights

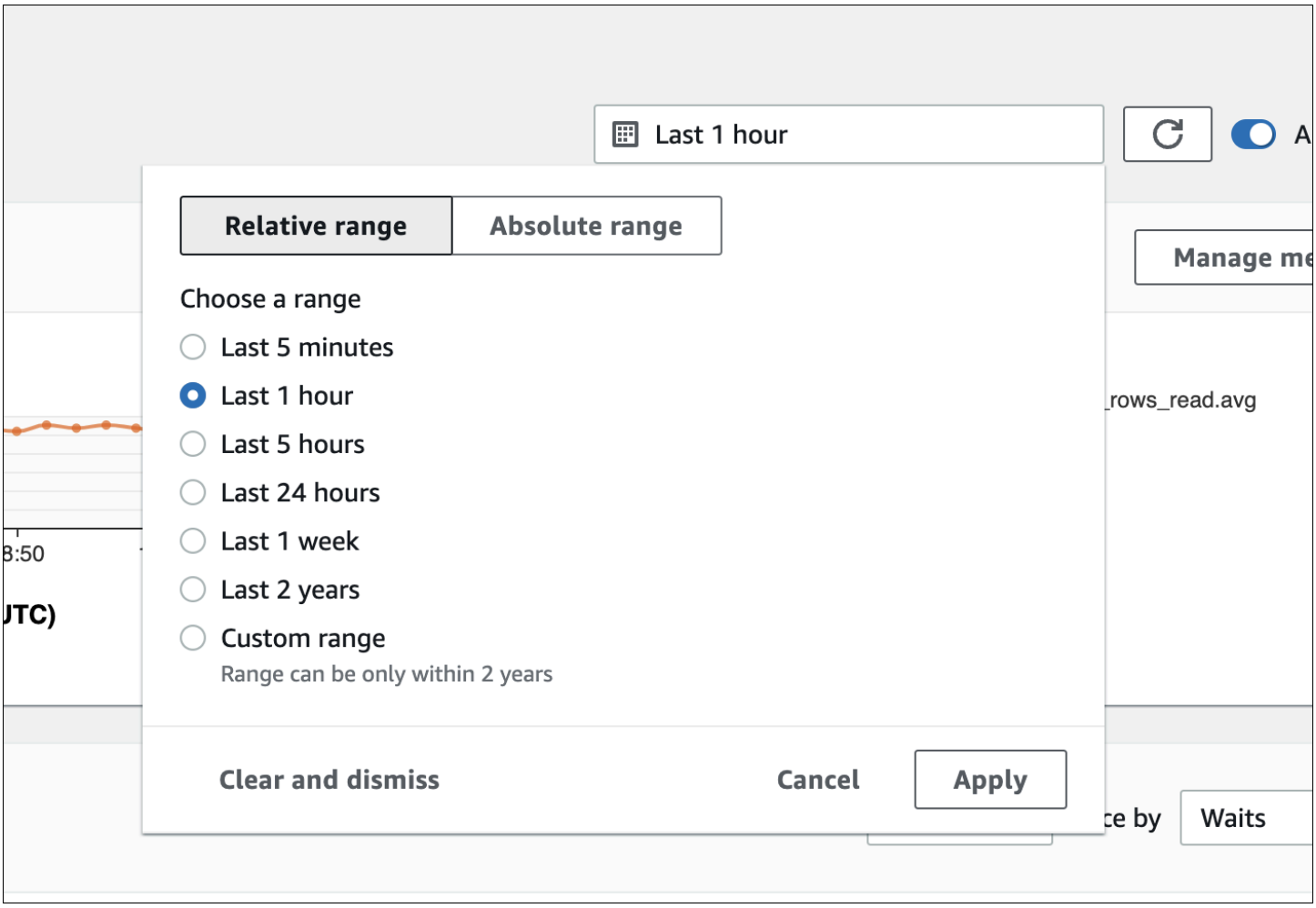
Il pannello di controllo è il modo più semplice per interagire con Performance Insights. L'esempio seguente mostra il pannello di controllo per un'istanza database MySQL.



Argomenti

Filtro intervallo temporale

Di default, il pannello di controllo di Performance Insights mostra il carico del database relativo all'ultima ora. Puoi regolare questo intervallo di tempo da 5 minuti o fino a 2 anni. Puoi inoltre selezionare un intervallo relativo personalizzato.



Puoi selezionare un intervallo assoluto con data e ora di inizio e fine. L'esempio seguente mostra l'intervallo di tempo che inizia a mezzanotte dell'11/4/22 e termina alle 23:59 del 14/4/22.

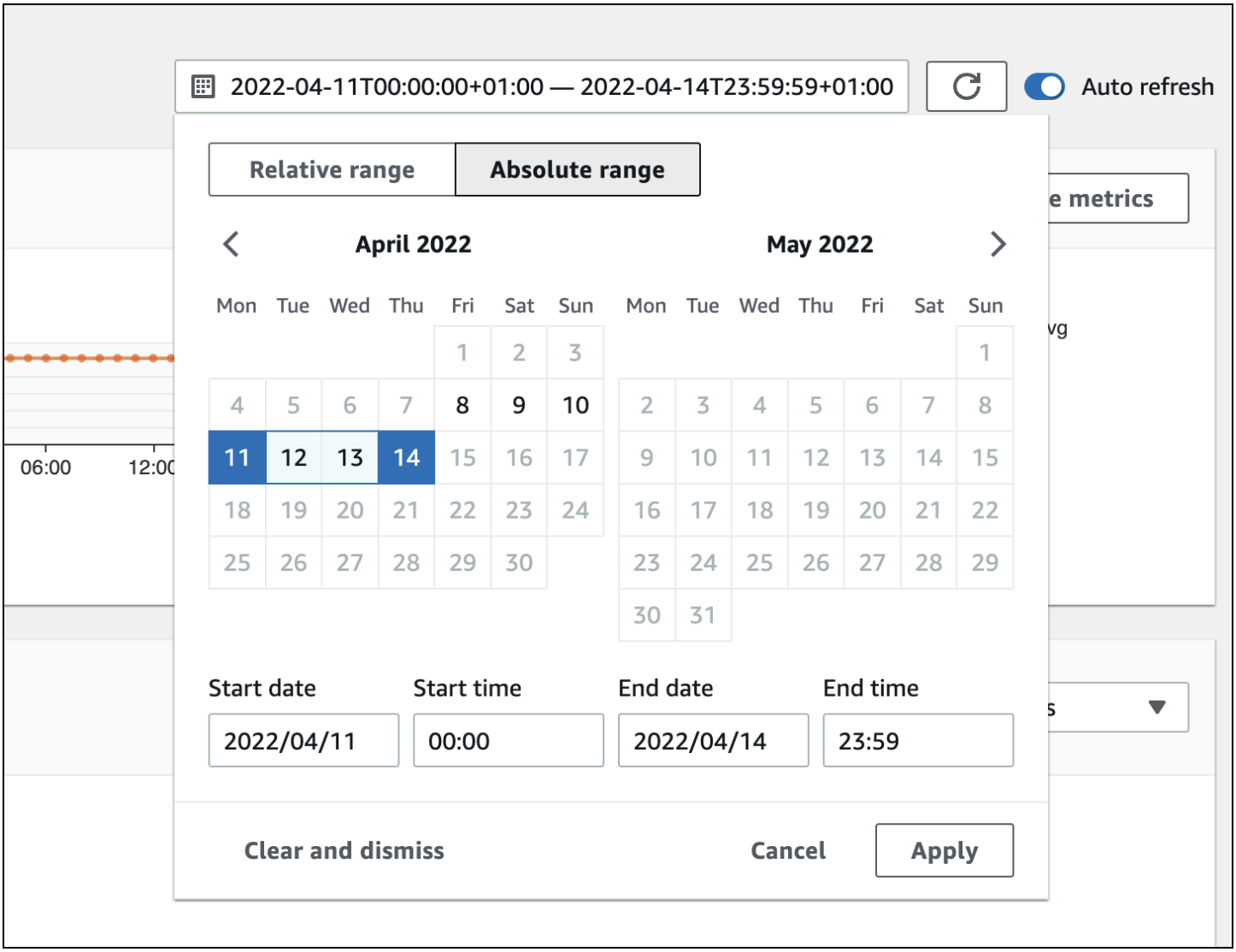


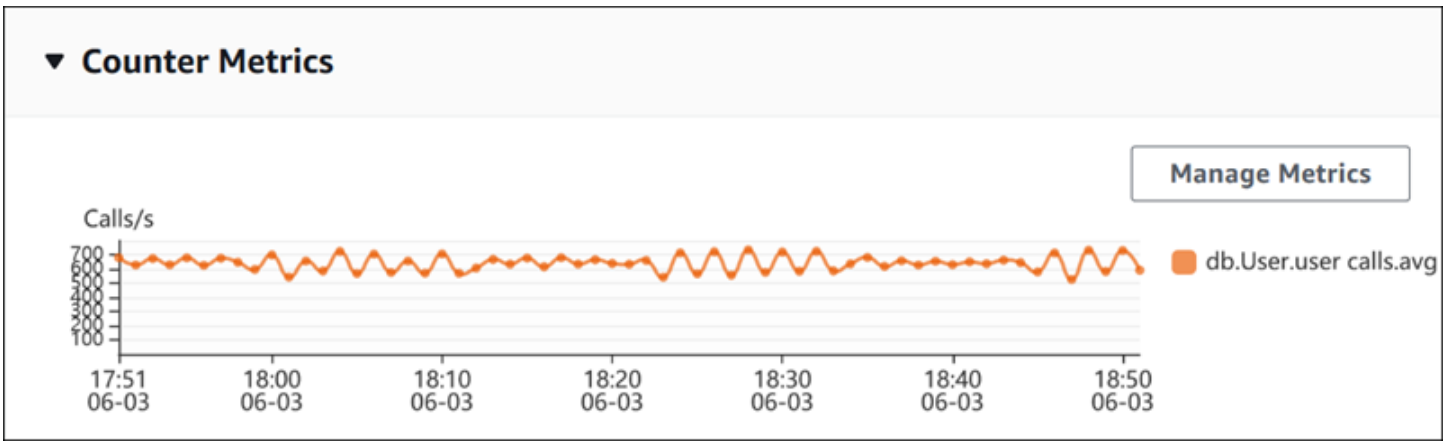
Grafico Parametri contatore

Con i parametri contatore, puoi personalizzare il pannello di controllo di Performance Insights per includere fino a 10 grafici aggiuntivi. Questi grafici mostrano una selezione di decine di parametri prestazionali di sistema operativo e database. Queste informazioni possono essere correlate ai carichi dei database per agevolare l'individuazione e l'analisi di problemi legati alle prestazioni.

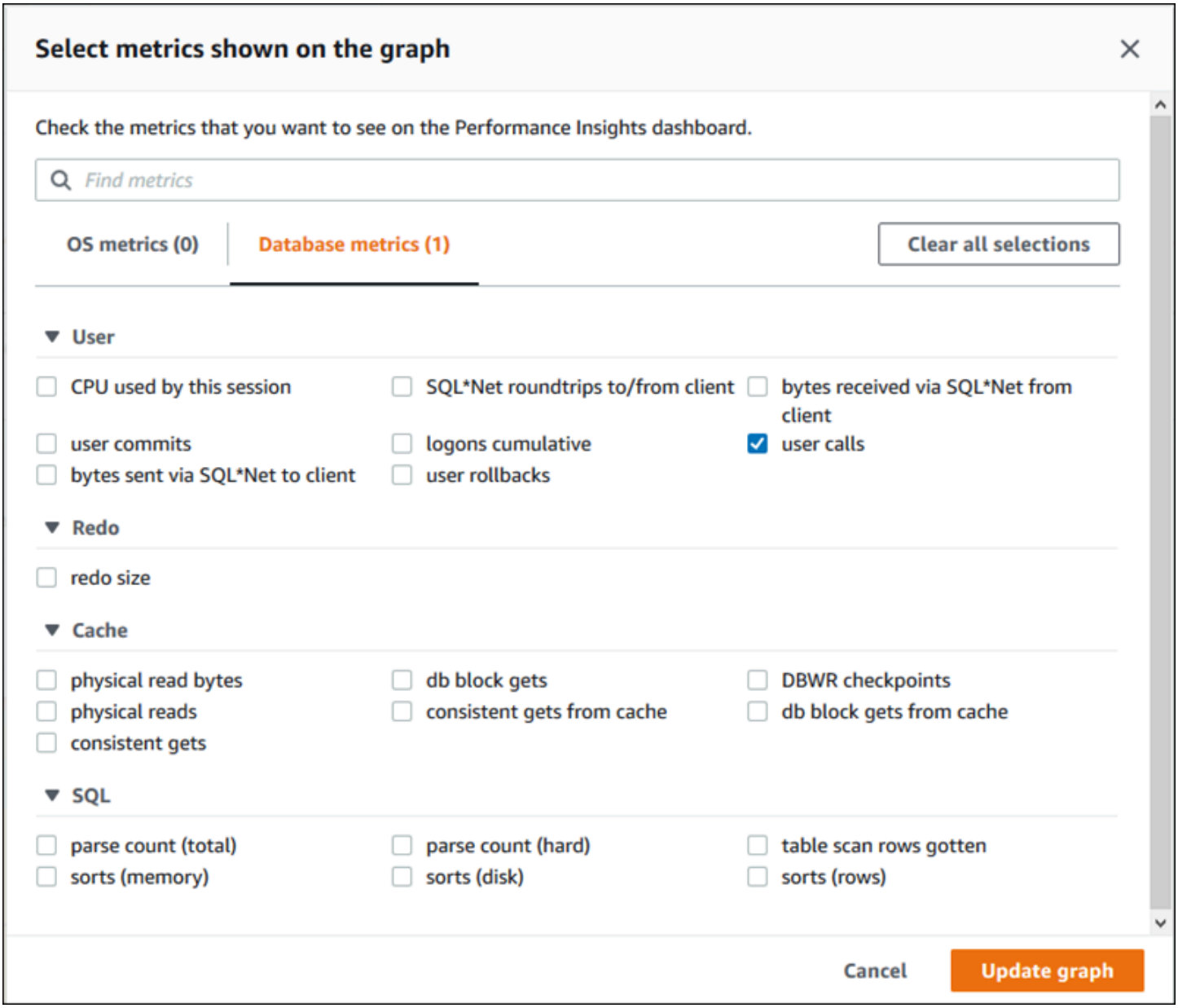
Il grafico Counter Metrics (Parametri contatore) visualizza i dati per i contatori delle prestazioni. I parametri predefiniti dipendono dal motore DB:

- MySQL e MariaDB – `db.SQL.Innodb_rows_read.avg`
- Oracle – `db.User.user calls.avg`
- Microsoft SQL Server – `db.Databases.Active Transactions(_Total).avg`

- PostgreSQL – `db.Transactions.xact_commit.avg`



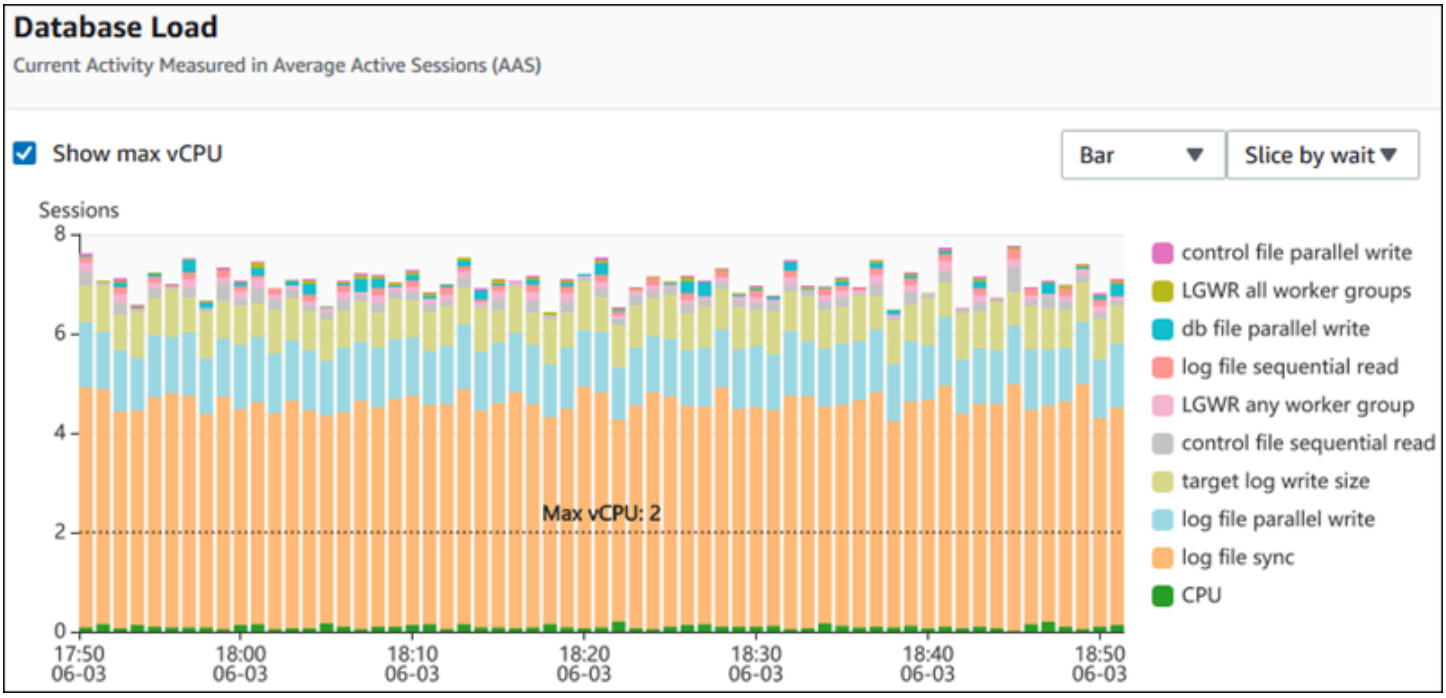
Per modificare i contatori delle prestazioni, scegli Manage Metrics (Gestisci parametri). È possibile selezionare più parametri del sistema operativo o metriche del database, come mostrato nello screenshot seguente. Per visualizzare i dettagli relativi a qualsiasi metrica, passare il mouse sul nome della metrica.



Per le descrizioni dei parametri contatore che è possibile aggiungere per ciascun motore database, consultare Parametri contatore di Performance Insights.

Grafico di carico database

Il grafico Database load (Carico database) mostra le differenze dell'attività del database in base alla capacità dell'istanza database, rappresentate dalla riga Max vCPU (vCPU massima). Per impostazione predefinita, il grafico a linee in pila rappresenta il carico DB come sessioni attive medie per unità di tempo. Il carico DB viene suddiviso (raggruppato) in base agli stati di attesa.

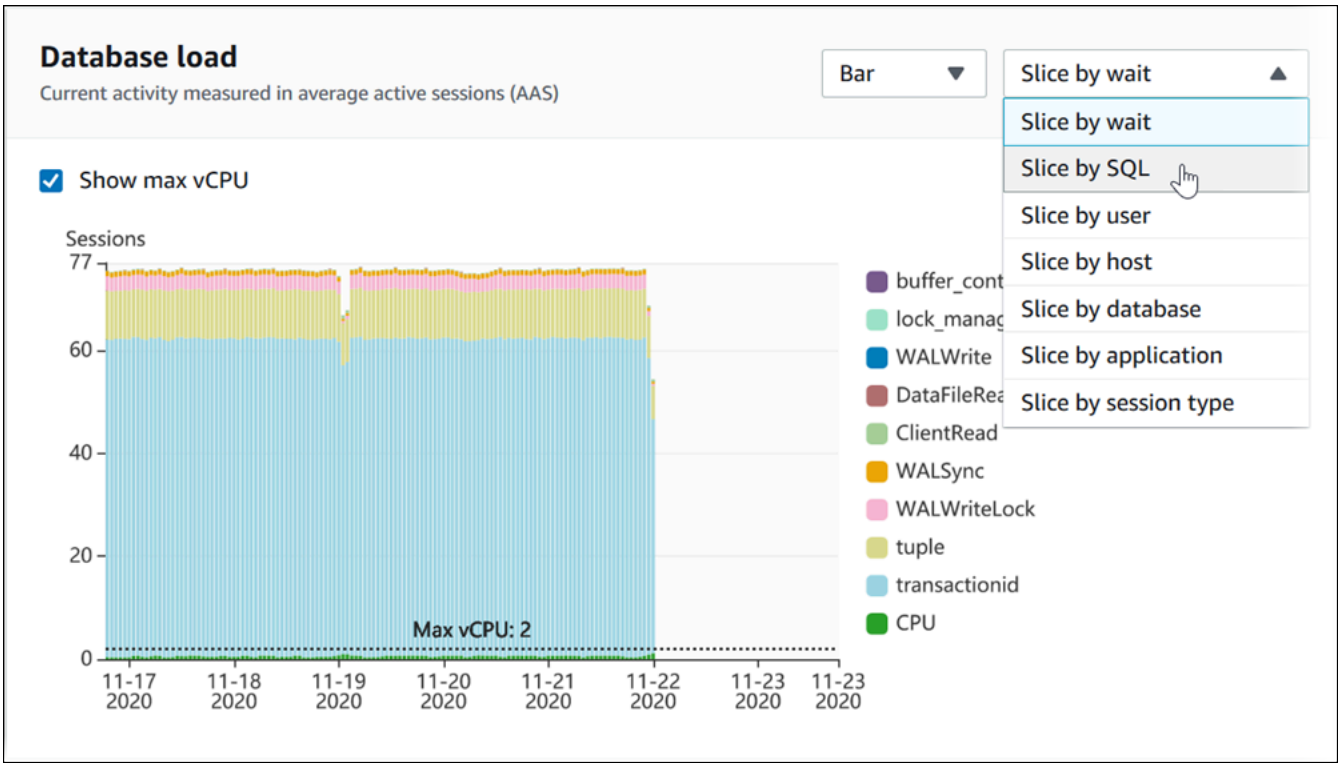


Carico del database suddiviso per dimensioni

È possibile scegliere di visualizzare il carico sotto forma di sessioni attive raggruppate in base alle dimensioni supportate. La tabella seguente mostra le dimensioni supportate per i diversi motori.

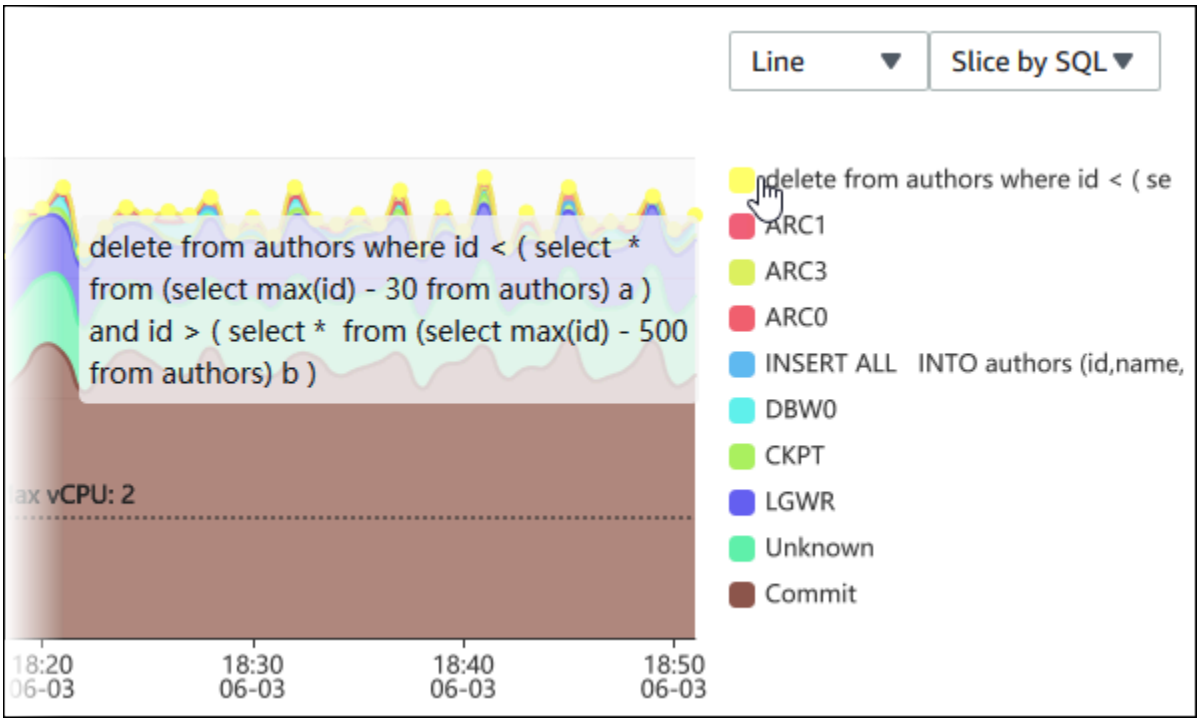
Dimensione	Oracle	SQL Server	PostgreSQL	MySQL
Host	Sì	Sì	Sì	Sì
SQL	Sì	Sì	Sì	Sì
Utente	Sì	Sì	Sì	Sì
Stati di attesa	Sì	Sì	Sì	Sì
Piani	Sì	No	No	No
Applicazione	No	No	Sì	No
Database	No	No	Sì	Sì
Tipo di sessione	No	No	Sì	No

L'immagine seguente mostra le dimensioni di un'istanza database PostgreSQL.



Dettagli del carico DB per un elemento della dimensione

Per visualizzare i dettagli su un elemento del carico del database all'interno di una dimensione, passa il mouse sul nome dell'elemento. L'immagine seguente mostra i dettagli di un'istruzione SQL.



Per visualizzare i dettagli relativi a qualsiasi elemento per il periodo di tempo selezionato nella legenda, passa il mouse su tale elemento.

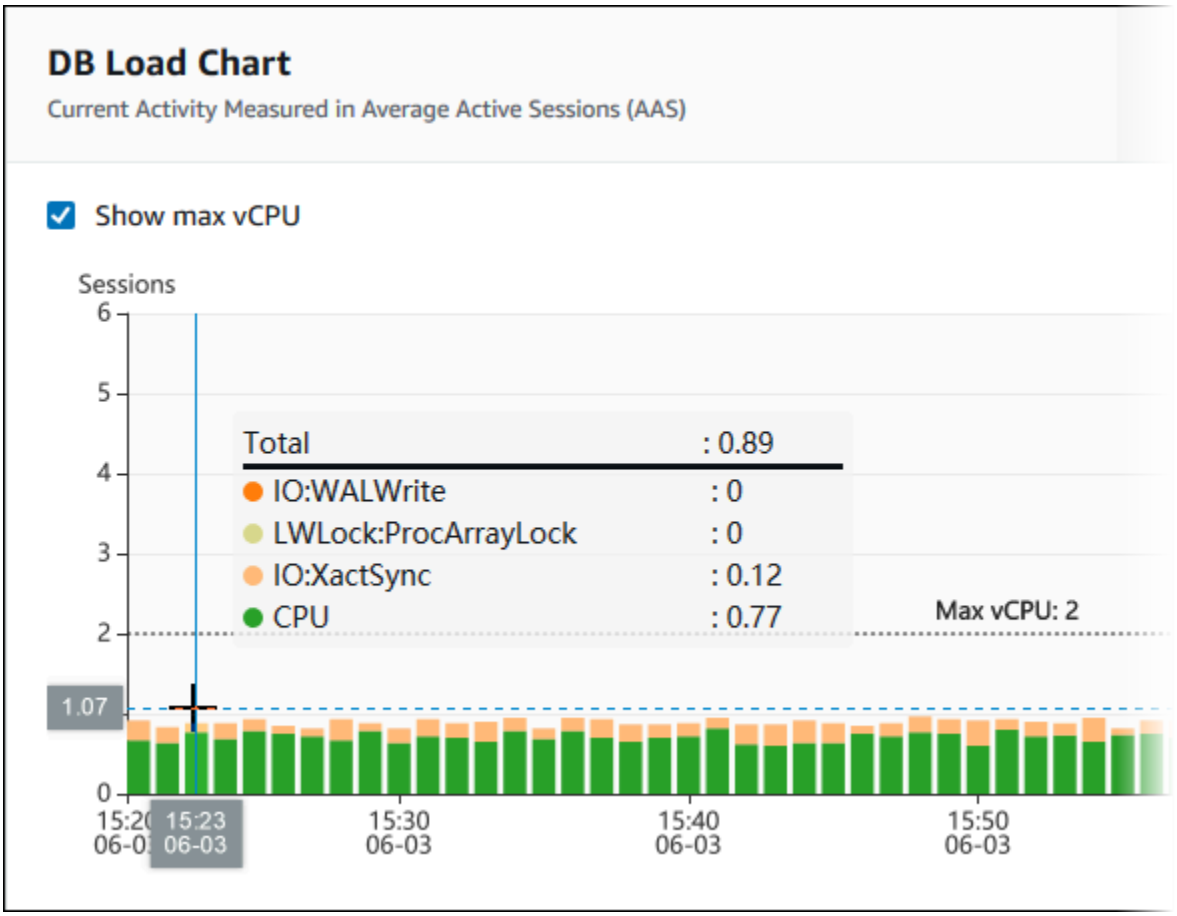
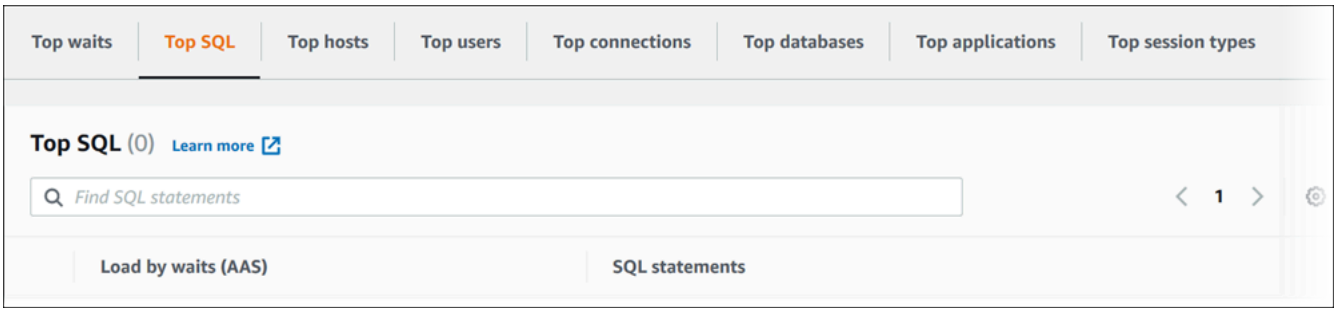


Tabella dimensioni superiori

La tabella delle dimensioni superiori seziona il carico DB in diverse dimensioni. Una dimensione è una categoria o una suddivisione per le diverse caratteristiche del carico del database. Se la dimensione è SQL, Top SQL (Prime istruzioni SQL) mostra le istruzioni SQL che contribuiscono maggiormente al carico DB.



Scegli una delle seguenti schede di dimensione.

Scheda	Descrizione	Motori supportati
Prime istruzioni SQL	Le istruzioni SQL correntem ente in esecuzione	Tutti
Principali stati d'attesa	L'evento per il quale il back-end del database è in attesa	Tutti
Host principali	Il nome host del client connesso	Tutti
Utenti principali	L'utente collegato al database	Tutti
Database principali	Nome del database a cui è connesso il client	Solo PostgreSQL, MySQL, MariaDB e SQL Server
Applicazioni principali	Il nome dell'applicazione connessa al database	
Tipi di sessione principali	Il tipo di sessione corrente	Solo PostgreSQL

Per informazioni sull'analisi delle query tramite la scheda Top SQL (Prime istruzioni SQL), vedi Panoramica della scheda Prime istruzioni SQL.

Accesso al pannello di controllo di Performance Insights

Amazon RDS fornisce una visualizzazione consolidata delle metriche di Performance Insights e CloudWatch nel pannello di controllo di Performance Insights.

Per visualizzare il pannello di controllo di Performance Insights, procedi come indicato di seguito.

Per visualizzare il pannello di controllo di Performance Insights nella console di gestione AWS

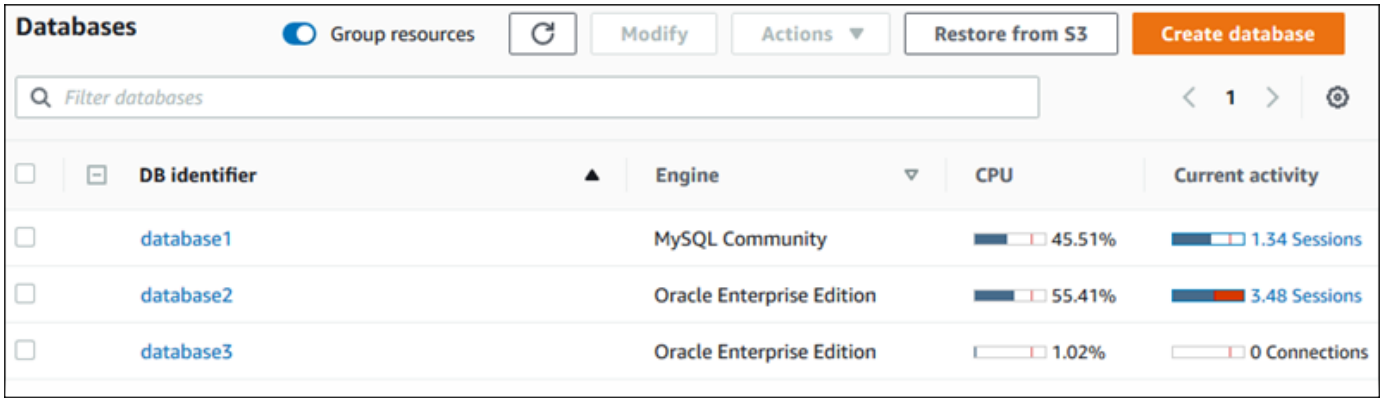
1. Apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel pannello di navigazione scegli Performance Insights.
3. Scegli un'istanza database.
4. Scegliere la visualizzazione di monitoraggio predefinita nella finestra visualizzata.

- Selezionare l'opzione Visualizzazione metriche di Performance Insights e CloudWatch (Nuova) e scegliere Continua per visualizzare le metriche di Performance Insights e CloudWatch.
- Selezionare l'opzione Visualizzazione Performance Insights e scegliere Continua per la visualizzazione di monitoraggio legacy. Continuare con questa procedura.

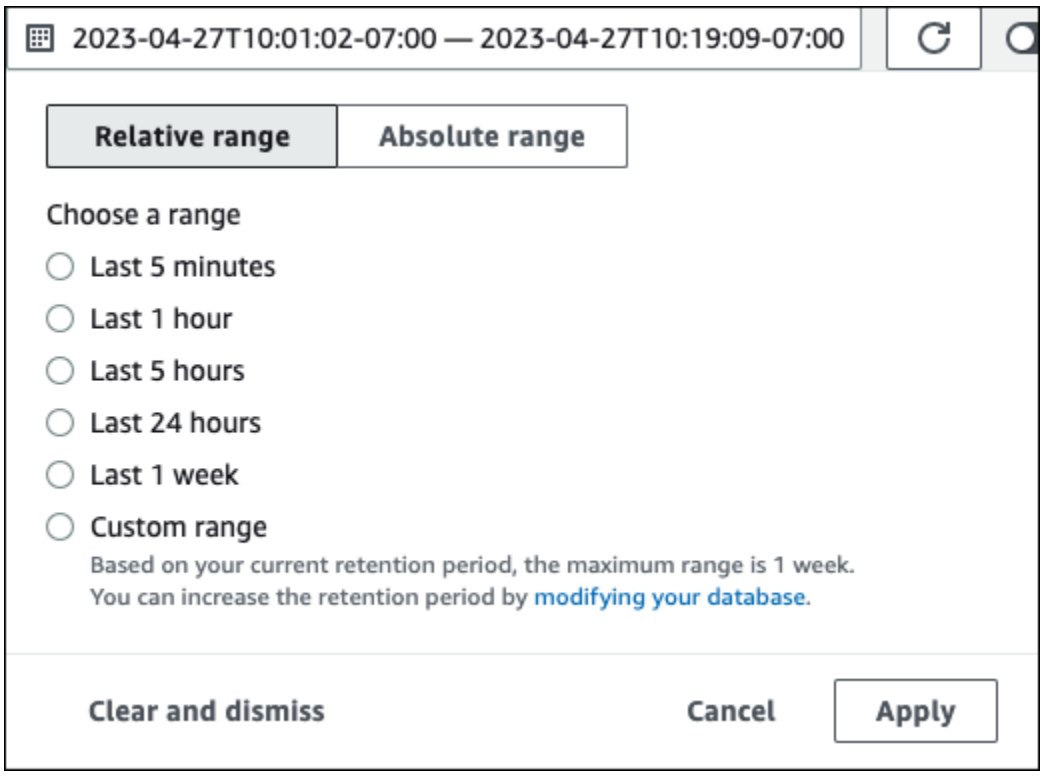
> **Note**
>
> Questa visualizzazione non sarà più disponibile a partire dal 15 dicembre 2023.

Viene visualizzato il pannello di controllo di Performance Insights per l'istanza database.

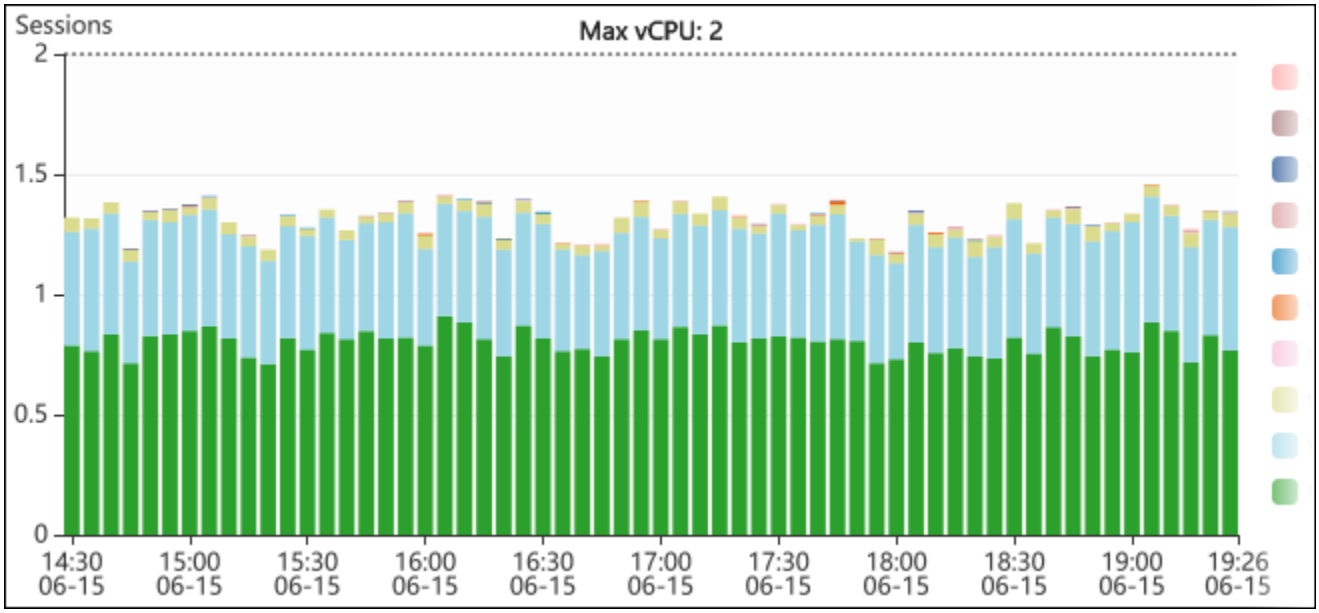
Per le istanze database con Performance Insights abilitato, è possibile accedere al pannello di controllo anche scegliendo la voce Sessioni nell'elenco delle istanze database. In Current activity (Attività corrente) la voce Sessions (Sessioni) mostra il carico del database in sessioni attive medie negli ultimi cinque minuti. Il grafico mostra graficamente il carico: Quando la barra è vuota, l'istanza database è inattiva. Con l'aumentare del carico, la barra si riempie ed è di colore blu. Quando il carico supera il numero di CPU virtuali (vCPU) nella classe di istanza database, la barra diventa rossa, a indicare un possibile collo di bottiglia.



5. (Facoltativo) Scegliere la data o l'intervallo di ore in alto a destra e specificare un intervallo di tempo relativo o assoluto diverso. È ora possibile specificare un periodo di tempo e generare un report di analisi delle prestazioni del database. Il report fornisce le informazioni e i suggerimenti identificati. Per ulteriori informazioni, consulta Creazione di un report di analisi delle prestazioni.

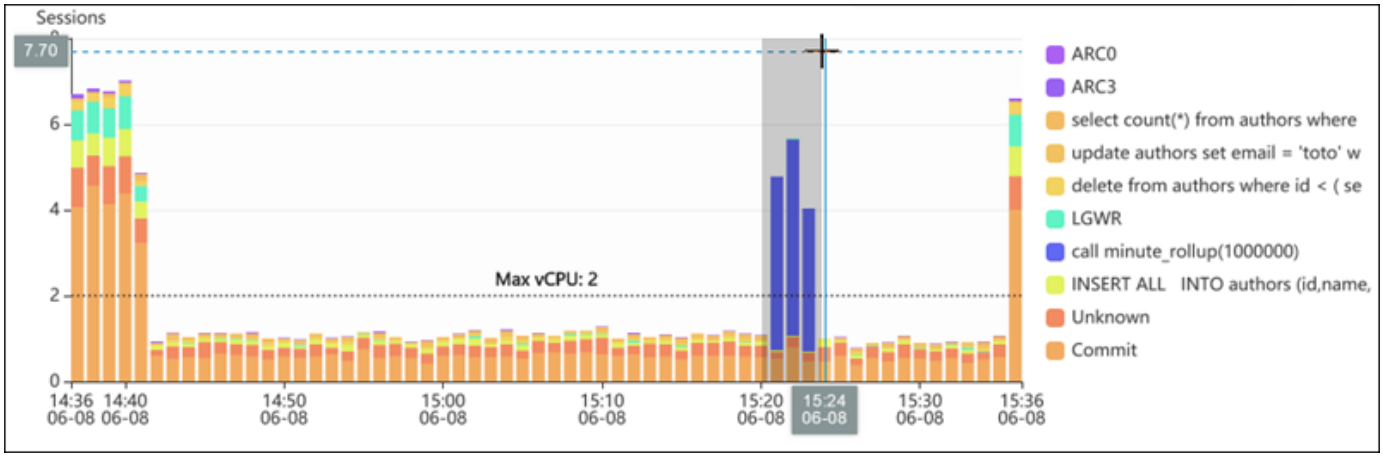


Nella schermata seguente, l'intervallo di caricamento DB è di 5 ore.

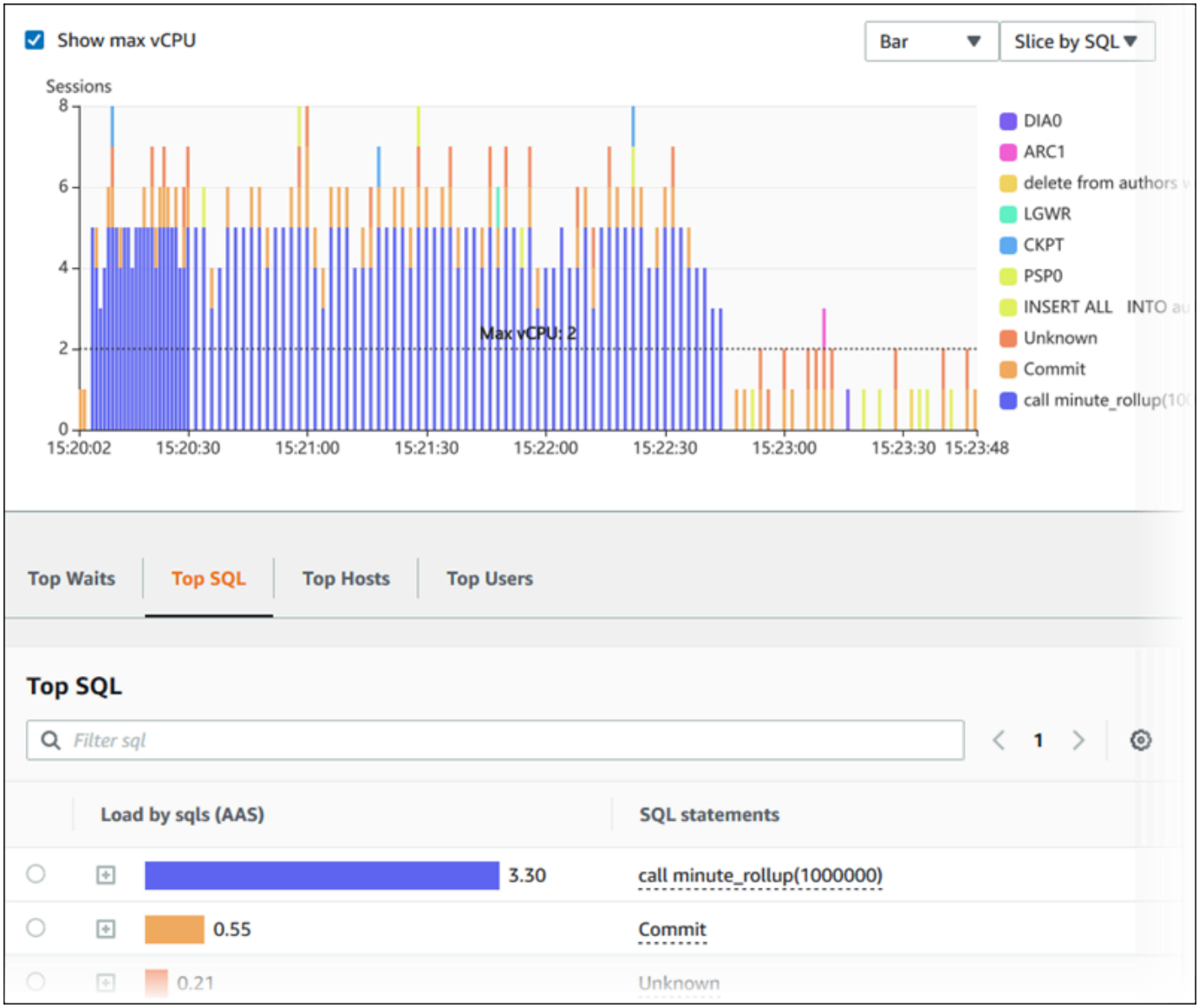


6. (Facoltativo) Per ingrandire una parte del grafico di carico del database, scegli l'ora di inizio e trascina fino alla fine del periodo di tempo che ti interessa.

   L'area selezionata viene evidenziata nel grafico del carico del database.



Quando rilasci il mouse, la parte selezionata del grafico del carico del database si ingrandisce nella regione AWS selezionata e la tabella Top dimensions (Dimensioni principali) viene ricalcolata.



7. (Facoltativo) Per aggiornare automaticamente i dati, selezionare Aggiornamento automatico.

Il pannello di controllo di Performance Insights si aggiorna automaticamente in base ai nuovi dati. La frequenza di aggiornamento dipende dalla quantità di dati visualizzati:

- Se scegli 5 minuti, l'aggiornamento avviene ogni 10 secondi.
- Se scegli 1 ora, l'aggiornamento avviene ogni 5 minuti.
- Se scegli 5 ore, l'aggiornamento avviene ogni 5 minuti.
- Se scegli 24 ore, l'aggiornamento avviene ogni 30 minuti.
- Se scegli 1 settimana, l'aggiornamento avviene ogni giorno.

- Se scegli 1 mese, l'aggiornamento avviene ogni giorno.

Analisi del carico del database per eventi di attesa

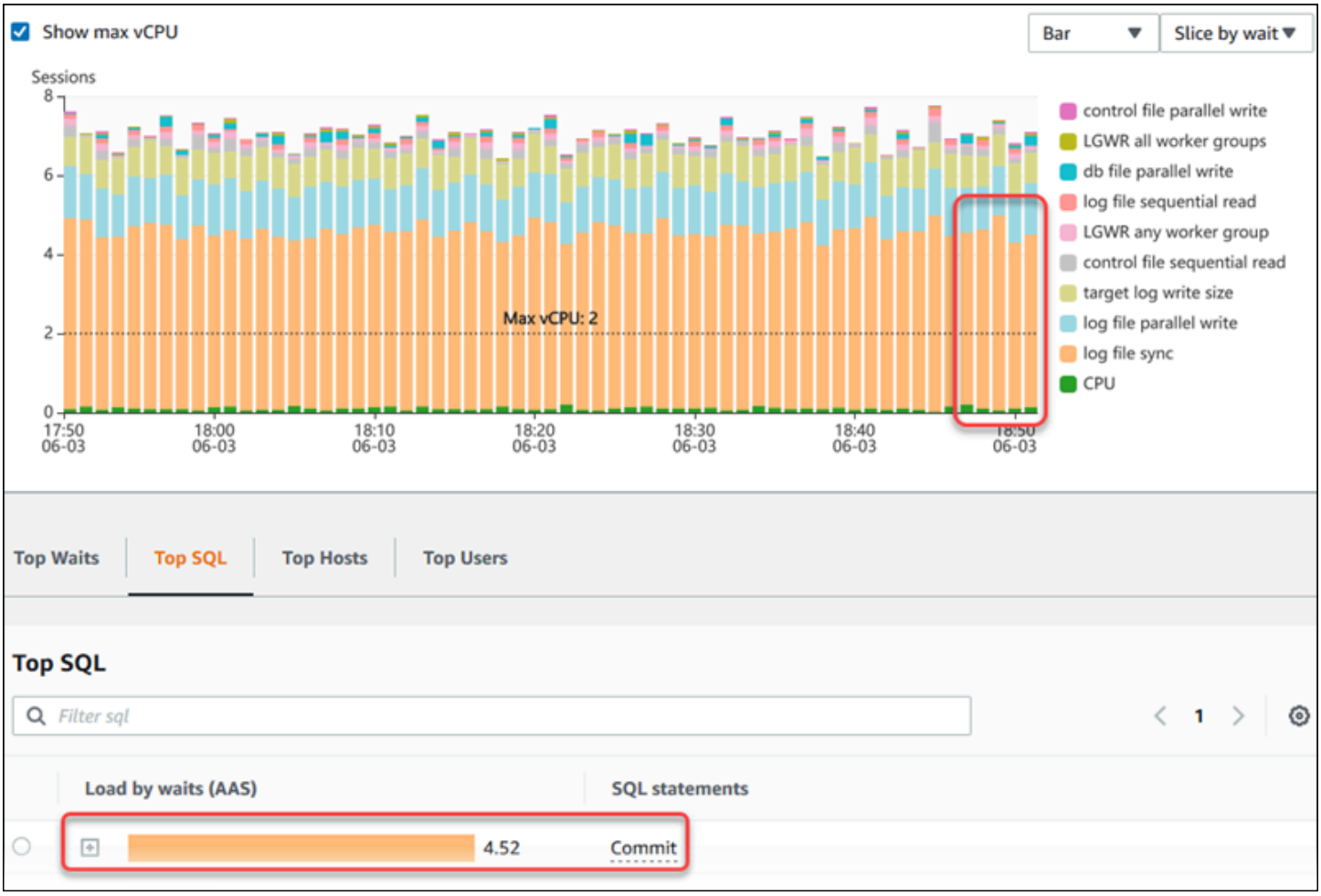
Se il grafico Database load (Caricamento database) mostra un collo di bottiglia, puoi identificare l'origine del carico. A questo scopo, osserva la tabella Top Load Items (Elementi con carico) sotto la tabella Database load (Caricamento database). Scegli uno specifico elemento, come una query SQL o un utente, ed effettua il drill-down di tale elemento per vedere i relativi dettagli.

Il carico del database raggruppato per attese e principali query SQL è la visualizzazione predefinita del pannello di controllo di Performance Insights. Questa combinazione offre di norma il maggior numero di informazioni sui problemi di prestazioni. Il carico del database raggruppato in base alle attese mostra la presenza di eventuali colli di bottiglia nel database relativamente alle risorse o alla simultaneità. In questo caso, la scheda SQL della tabella Top Load Items (Elementi con carico massimo) mostra quali query fanno aumentare il carico.

Il flusso di lavoro tipico per diagnosticare problemi di performance è il seguente:

1. Esaminare il grafico Database load (Caricamento database) per determinare se sono presenti eventi imprevisti di superamento della riga Max CPU (CPU max) da parte del carico del database.
2. Se sono presenti, osservare il grafico Database load (Caricamento database) e individuare lo stato o gli stati di attesa che sono i principali responsabili.
3. Identificare le query digest che provocano il carico individuando quali delle query della scheda SQL nella tabella Top Load Items (Elementi con carico massimo) contribuiscono maggiormente agli stati di attesa. È possibile identificarle attraverso la colonna DB Load by Waits (Carico del database in base alle attese).
4. Scegliere una delle query digest nella scheda SQL per espanderla e osservare le query figlio da cui è composta.

Ad esempio, nel dashboard seguente, la sincronizzazione file di registro attende account per la maggior parte del carico DB. Anche l'attesa di tutti i gruppi di lavoro LGWR è elevata. Il grafico Top SQL mostra ciò che causa le attese di sincronizzazione del file di registro: istruzioni `COMMIT` frequenti. In questo caso, il commit meno frequentemente ridurrà il carico del DB.



Analisi delle prestazioni del database per un periodo di tempo

Analizza le prestazioni del database con l'analisi su richiesta creando un rapporto di analisi delle prestazioni per un periodo di tempo. Visualizza i report di analisi delle prestazioni per individuare problemi relativi alle prestazioni, come problemi di risorse o modifiche a una query nell'istanza DB. Il pannello di controllo di Performance Insights consente di selezionare un periodo di tempo e creare un report di analisi delle prestazioni. Puoi anche aggiungere uno o più tag al report.

Per utilizzare questa funzionalità, devi utilizzare il periodo di conservazione del piano a pagamento. Per ulteriori informazioni, consulta Prezzi e conservazione dei dati per Performance Insights

Il report è disponibile nella scheda Report di analisi delle prestazioni - nuovi per la selezione e la visualizzazione. Il report contiene informazioni dettagliate, parametri correlati e suggerimenti per risolvere il problema relativo alle prestazioni. Il report è disponibile per la visualizzazione per tutta la durata del periodo di conservazione di Performance Insights.

Il report viene eliminato se l'ora di inizio del periodo di analisi del report è esterna al periodo di conservazione. È anche possibile eliminare il report prima della fine del periodo di conservazione.

Per individuare i problemi di prestazioni e generare il report di analisi per l'istanza database, è necessario attivare Performance Insights. Per ulteriori informazioni sull'attivazione di Performance Insights, consultare Attivazione e disattivazione di Performance Insights.

Per informazioni sull'assistenza alla regione, al motore di database e alla classe di istanza per questa funzionalità, consulta Supporto di classe di istanza, regione e motore di database Amazon RDS per funzionalità Performance Insights

Creazione di un report di analisi delle prestazioni

È possibile creare un report di analisi delle prestazioni per un periodo specifico nella pannello di controllo di Performance Insights. È possibile selezionare un periodo di tempo e aggiungere uno o più tag al report di analisi.

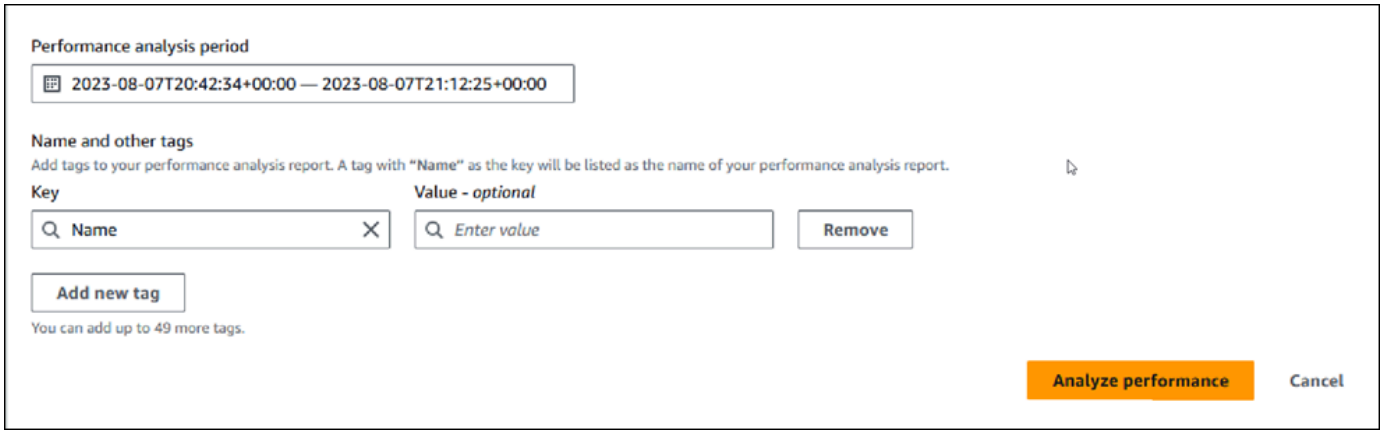
Il periodo di analisi può variare da 5 minuti a 6 giorni. Occorre disporre di almeno 24 ore di dati di prestazioni prima dell'ora di inizio dell'analisi.

Per creare un report di analisi delle prestazioni per un periodo di tempo

1. Apri la console di Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel pannello di navigazione scegli Performance Insights.
3. Scegli un'istanza database.

   Viene visualizzato il pannello di controllo di Approfondimenti sulle prestazioni per l'istanza database.
4. Scegli Analizza le prestazioni nella sezione Caricamento del database sul pannello di controllo.

   Vengono visualizzati i campi per impostare il periodo di tempo e aggiungere uno o più tag al report di analisi delle prestazioni.

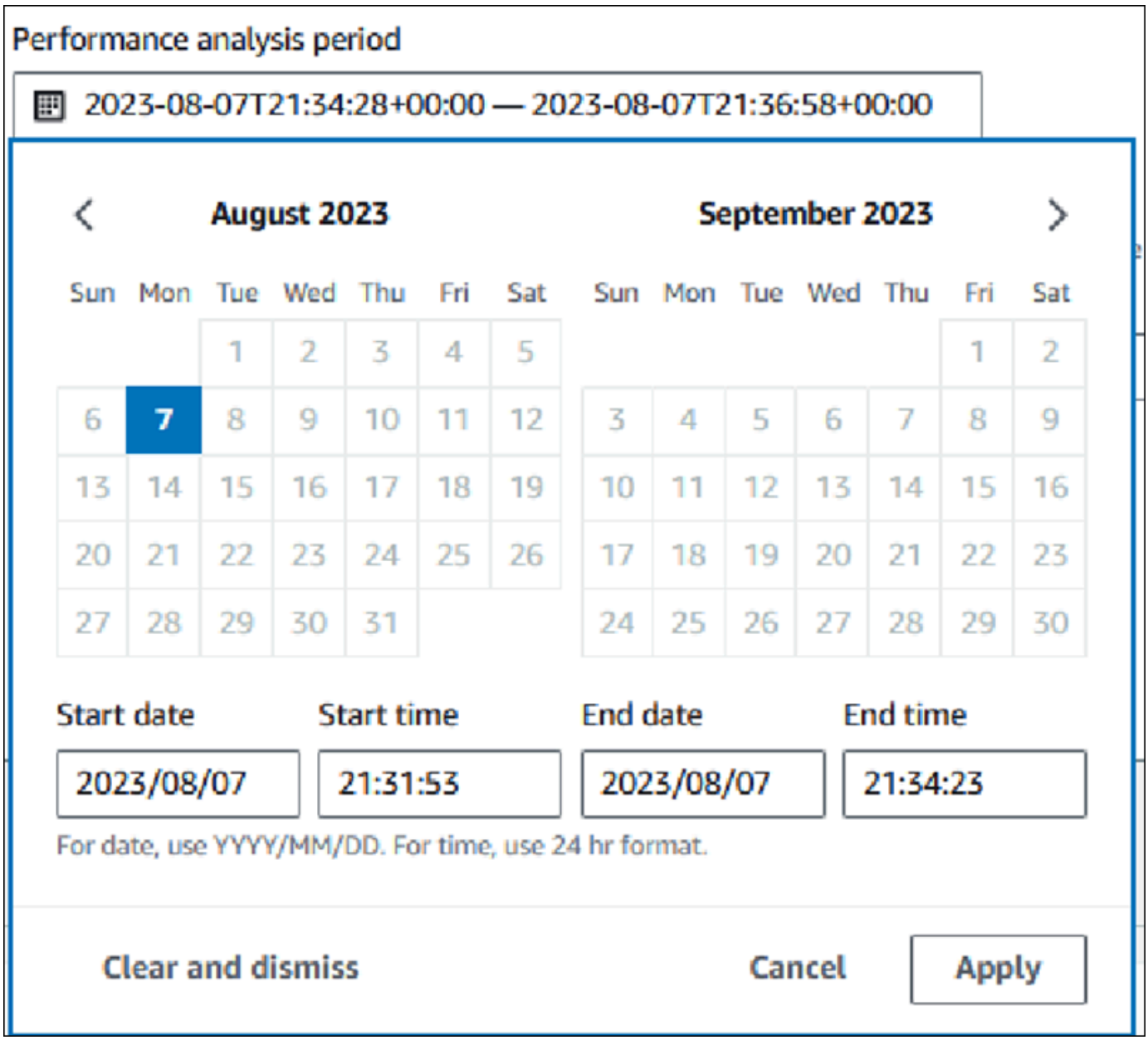


5. Scegli il periodo di tempo. Se imposti un periodo di tempo nel campo Intervallo relativo o Intervallo assoluto nell'angolo in alto a destra, puoi inserire o selezionare la data e l'ora del report di analisi solo entro questo periodo di tempo. Se selezioni il periodo di analisi al di fuori di questo periodo di tempo, viene visualizzato un messaggio di errore.

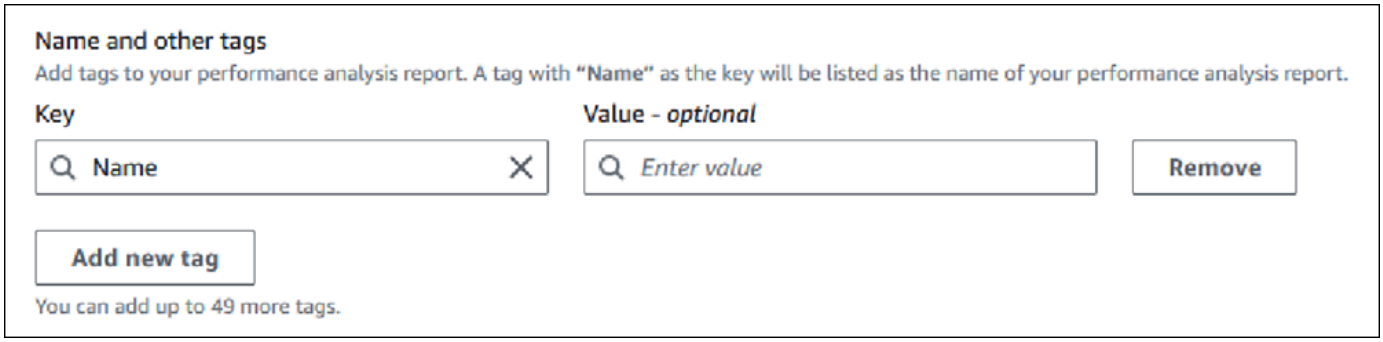
   Per impostare il periodo di tempo, puoi effettuare una delle seguenti operazioni:

   - Premi e trascina uno qualsiasi dei dispositivi di scorrimento sul grafico del carico del database.

     Nella casella Periodo di analisi delle prestazioni viene visualizzato il periodo di tempo selezionato e il grafico del carico del database evidenzia il periodo di tempo selezionato.
   - Scegli Data di inizio, Ora di inizio, Data di fine e Ora di fine nella casella Periodo di analisi delle prestazioni.



6. (Facoltativo) Inserisci Chiave e Valore-opzionale per aggiungere un tag per il report.



7. Scegli Analizza le prestazioni.

   Un banner mostra un messaggio a prescindere dall'esito della generazione del report. Il messaggio fornisce anche il collegamento per visualizzare il report.

   L'esempio seguente mostra il banner con il messaggio di creazione del report completata.

   Il report è disponibile per la visualizzazione nella scheda Report di analisi delle prestazioni - nuovi.

Puoi creare un report di analisi delle prestazioni utilizzando la AWS CLI. Per un esempio su come creare un report utilizzando AWS CLI, consulta. Creazione di un report di analisi delle prestazioni per un periodo di tempo

Visualizzazione di un report di analisi delle prestazioni

Nella scheda Report di analisi delle prestazioni - nuovi vengono elencati tutti i report creati per l'istanza database. Per ogni report viene visualizzato quanto segue:

- ID: identificatore univoco del report.
- Nome: chiave di tag aggiunta al report.
- Ora di creazione del report: ora in cui il report è stato creato.
- Ora di inizio dell'analisi: ora di inizio dell'analisi nel report.
- Ora di fine dell'analisi: ora di fine dell'analisi nel report.

Per visualizzare un report di analisi delle prestazioni

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel pannello di navigazione scegli Performance Insights.
3. Scegli un'istanza database per la quale desideri visualizzare il report di analisi.

   Viene visualizzato il pannello di controllo di Approfondimenti sulle prestazioni per l'istanza database.
4. Scorri verso il basso e scegli la scheda Report di analisi delle prestazioni - nuovi.

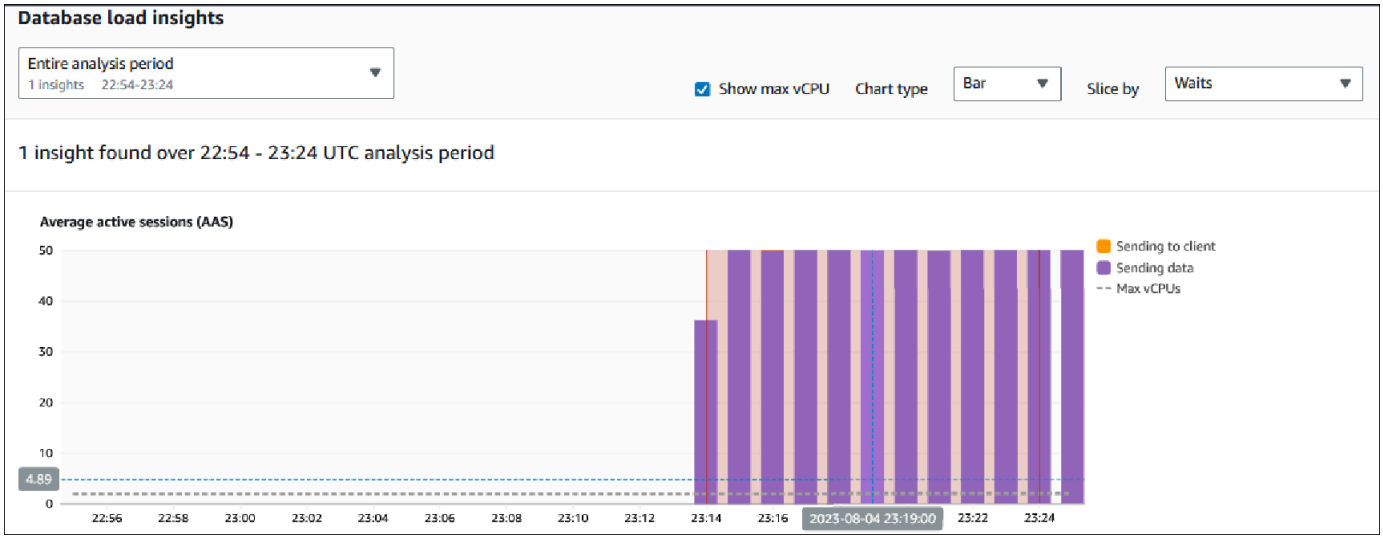
Vengono visualizzati tutti i report di analisi per i diversi periodi di tempo.

5. Scegli ID del report che desideri visualizzare.

   Il grafico del carico del database mostra l'intero periodo di analisi per impostazione predefinita se vengono identificati più approfondimenti. Se il report ha identificato un approfondimento, il grafico del carico del database visualizza l'approfondimento per impostazione predefinita.

   Nel pannello di controllo vengono elencati anche i tag per il report nella sezione Tag.

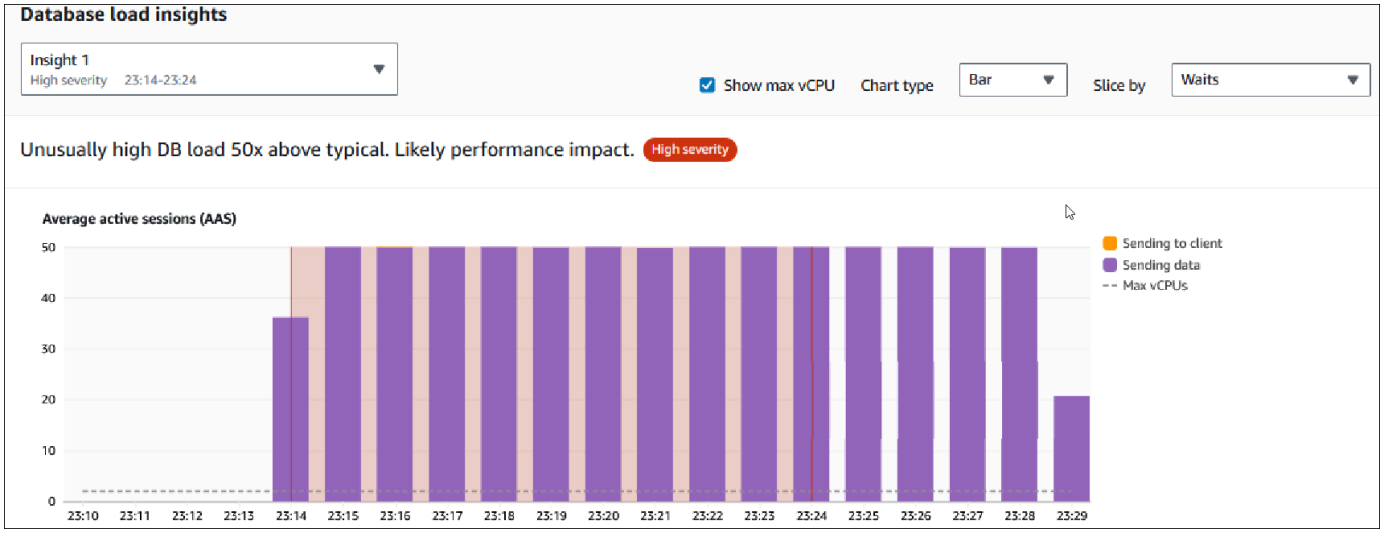
   L'esempio seguente mostra l'intero periodo di analisi per il report.

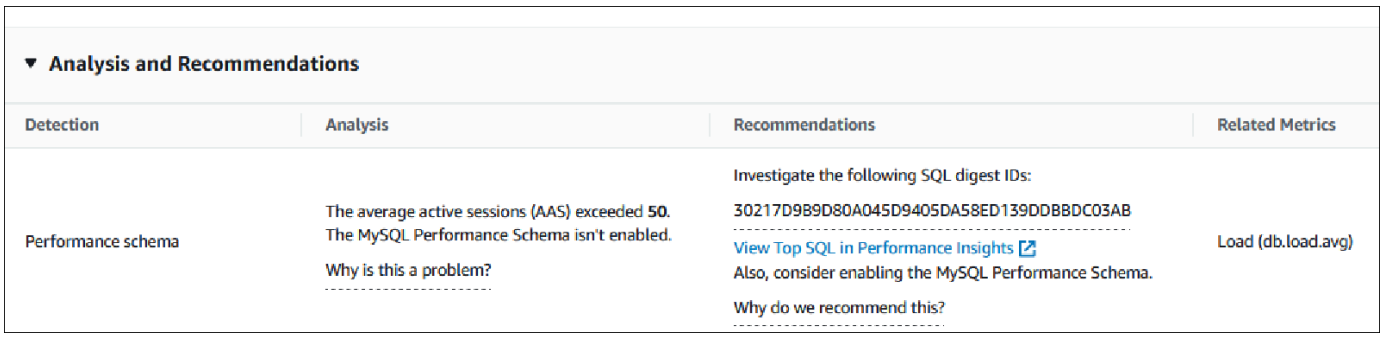


6. Scegli l'approfondimento nell'elenco Informazioni dettagliate sul carico del database che desideri visualizzare se nel report vengono identificati più approfondimenti.

   Il pannello di controllo mostra il messaggio di approfondimento, il grafico del carico del database evidenziando il periodo di tempo dell'approfondimento, l'analisi e i suggerimenti e l'elenco dei tag del report.

   Nell'esempio seguente viene mostrato l'approfondimento del carico del database nel report.





Aggiunta di tag a un report di analisi delle prestazioni

È possibile aggiungere un tag quando si crea o visualizza un report. È possibile aggiungere fino a 50 tag per un report.

Per aggiungere i tag sono richieste autorizzazioni. Per ulteriori informazioni sulle policy di accesso per Performance Insights, consulta Configurazione delle policy di accesso per Performance Insights

Per aggiungere uno o più tag durante la creazione di un report, consulta il passaggio 6 della procedura Creazione di un report di analisi delle prestazioni.

Per aggiungere uno o più tag durante la visualizzazione di un report

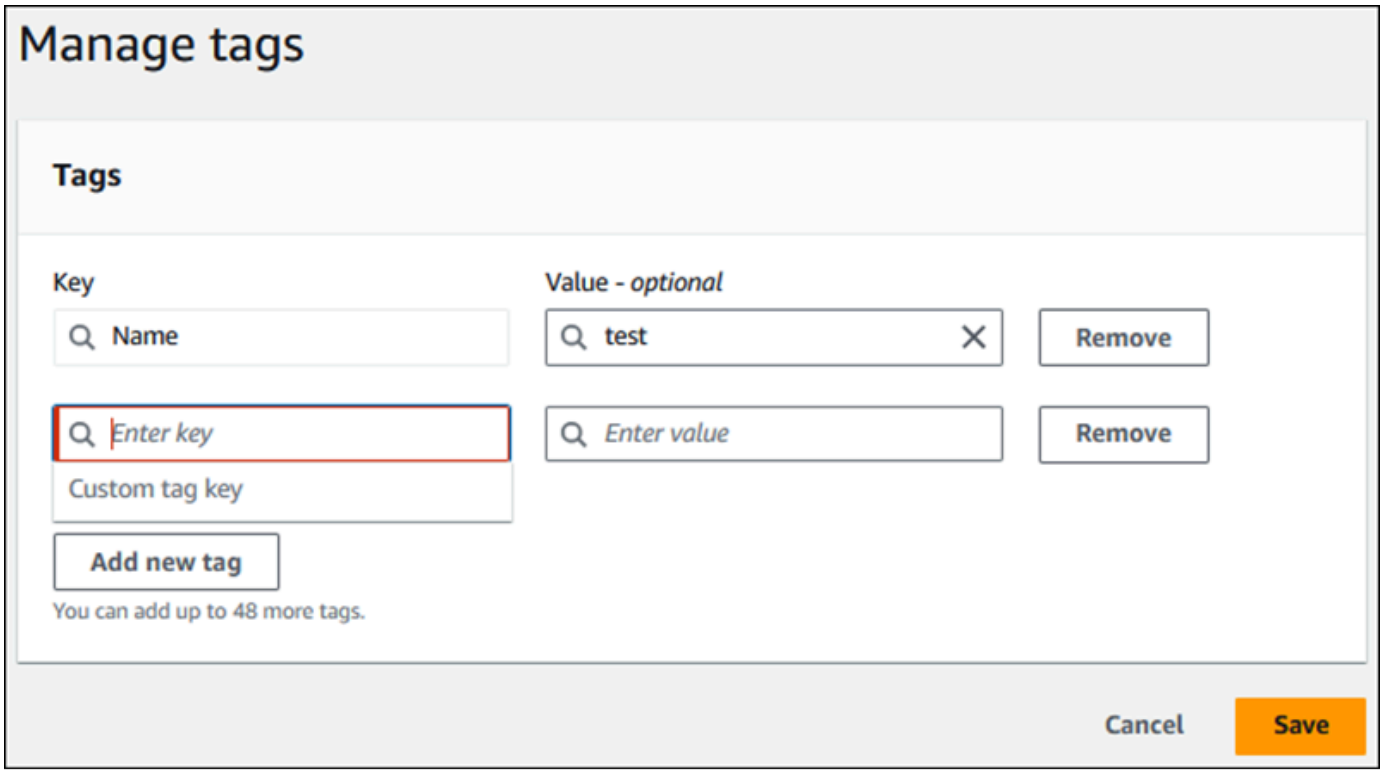
1. Apri la console di Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel pannello di navigazione scegli Performance Insights.
3. Scegli un'istanza database.

Viene visualizzato il pannello di controllo di Approfondimenti sulle prestazioni per l'istanza database.

4. Scorri verso il basso e scegli la scheda Report di analisi delle prestazioni - nuovi.
5. Scegli il report per il quale desideri aggiungere i tag.

   Il pannello di controllo visualizza il report.
6. Scorri verso il basso fino a Tag e scegli Gestisci i tag.
7. Scegli Aggiungi nuovo tag.
8. Immetti Chiave e Valore-facoltativo e scegli Aggiungi nuovo tag.

   Nell'esempio seguente viene fornita la possibilità di aggiungere un nuovo tag per il report selezionato.



   Viene creato un nuovo tag per il report.

   L'elenco dei tag per il report viene visualizzato nella sezione Tag sul pannello di controllo. Se desideri rimuovere un tag dal report, scegli Rimuovi accanto al tag.

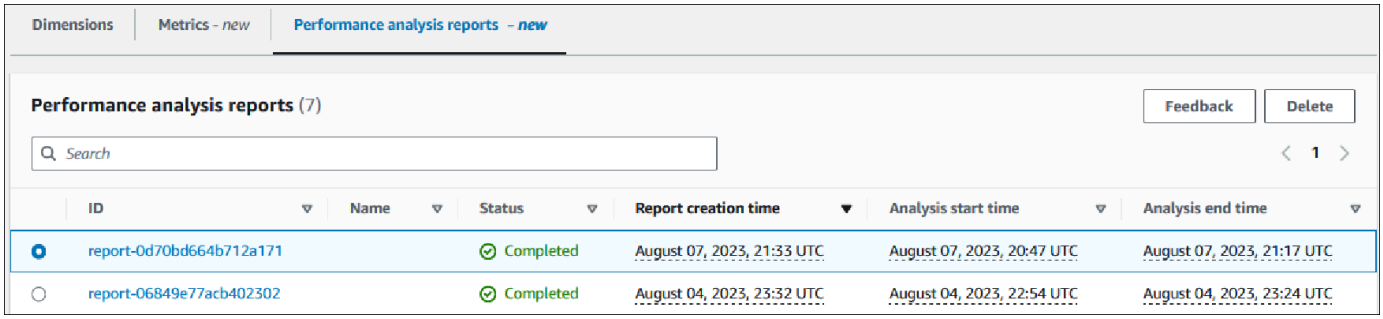
Eliminazione di un report di analisi delle prestazioni

È possibile eliminare un report dall'elenco dei report visualizzato nella scheda Report di analisi delle prestazioni o durante la visualizzazione di un report.

Per eliminare un report

1. Apri la console di Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel pannello di navigazione scegli Performance Insights.
3. Scegli un'istanza database.

   Viene visualizzato il pannello di controllo di Approfondimenti sulle prestazioni per l'istanza database.
4. Scorri verso il basso e scegli la scheda Report di analisi delle prestazioni - nuovi.
5. Seleziona il report che desideri eliminare e scegli Elimina nell'angolo in alto a destra.



   Viene visualizzata una finestra di conferma. Il report viene eliminato dopo aver scelto l'opzione di conferma.
6. (Facoltativo) Scegli ID del report che desideri eliminare.

   Nella pagina del report, scegli Elimina nell'angolo in alto a destra.

   Viene visualizzata una finestra di conferma. Il report viene eliminato dopo aver scelto l'opzione di conferma.

Analisi delle query all'interno del pannello di controllo di Performance Insights

Nel pannello di controllo di Amazon RDS Performance Insights è possibile trovare informazioni relative alle query in esecuzione e recenti nella scheda Top SQL (Prime istruzioni SQL) nella tabella Top dimensions (Dimensioni principali). Queste informazioni possono essere utilizzate per ottimizzare le query.

Argomenti

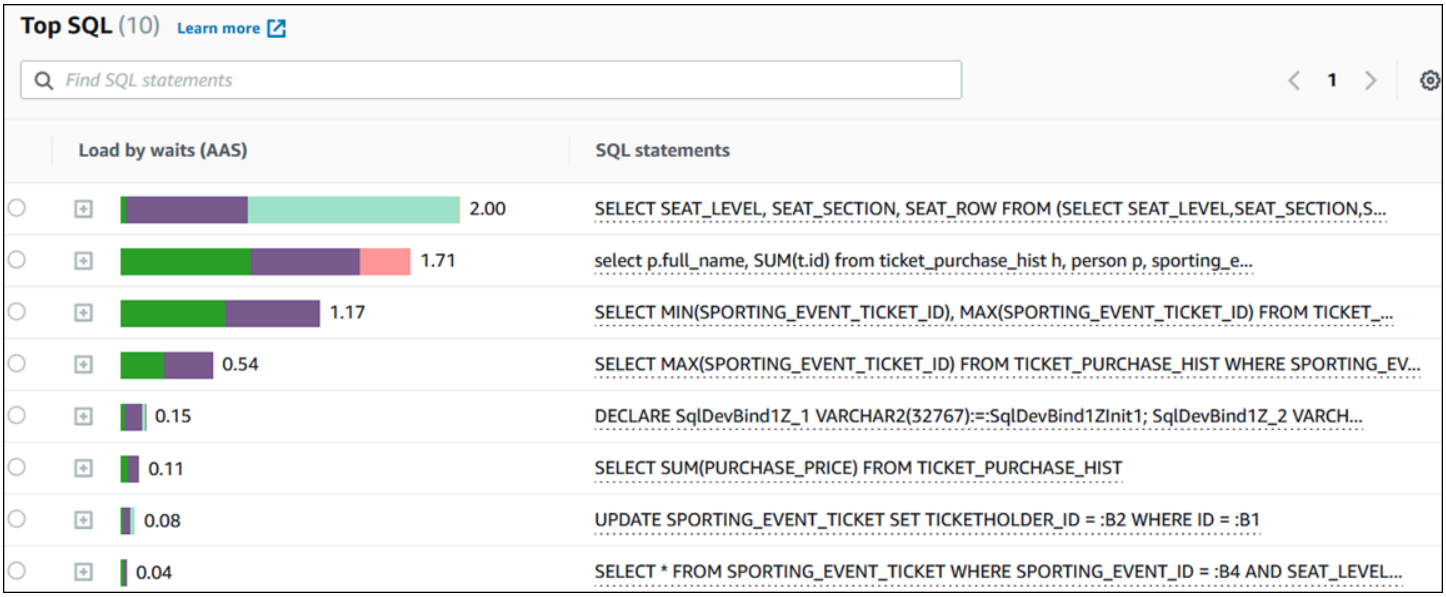
Panoramica della scheda Prime istruzioni SQL

Per impostazione predefinita, la scheda Top SQL (Prime istruzioni SQL) mostra le 25 query che contribuiscono di più al carico del database. Per ottimizzare le query puoi analizzare le informazioni, ad esempio il testo della query e le statistiche SQL. È inoltre possibile scegliere le statistiche che desideri visualizzare nella scheda Top SQL (Prime istruzioni SQL).

Argomenti

Testo SQL

Per impostazione predefinita, ciascuna riga nella tabella Top SQL (Prime istruzioni SQL) mostra 500 byte di testo per ogni istruzione.

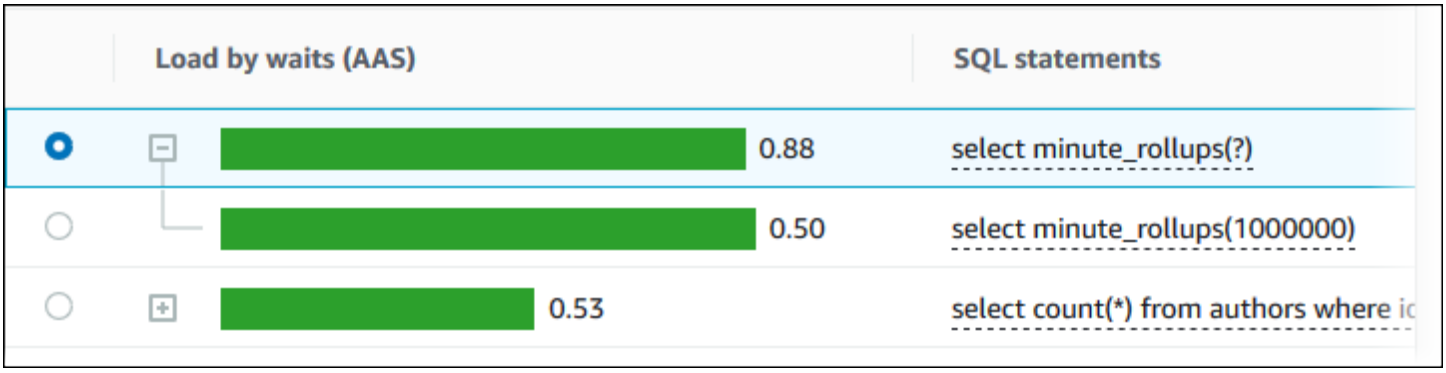


Per informazioni su come visualizzare più dei 500 byte di testo SQL di default, consulta Accesso a una maggiore quantità di testo SQL nel pannello di controllo di Performance Insights.

Un digest SQL è un composito di più query effettive strutturalmente simili ma potrebbero avere valori letterali diversi. Il digest sostituisce i valori codificati con un punto interrogativo. Ad esempio, un digest potrebbe essere `SELECT * FROM emp WHERE lname= ?`. Questo digest può includere le seguenti query figlio:

```
SELECT * FROM emp WHERE lname = 'Sanchez'
SELECT * FROM emp WHERE lname = 'Olagappan'
SELECT * FROM emp WHERE lname = 'Wu'
```

Per visualizzare le istruzioni SQL letterali in un digest, selezionare la query, quindi scegliere il simbolo più (+). Nell'esempio seguente, la query selezionata è un sunto.



> **Note**
>
> Un sunto SQL raggruppa istruzioni SQL simili, ma non oscura le informazioni riservate.

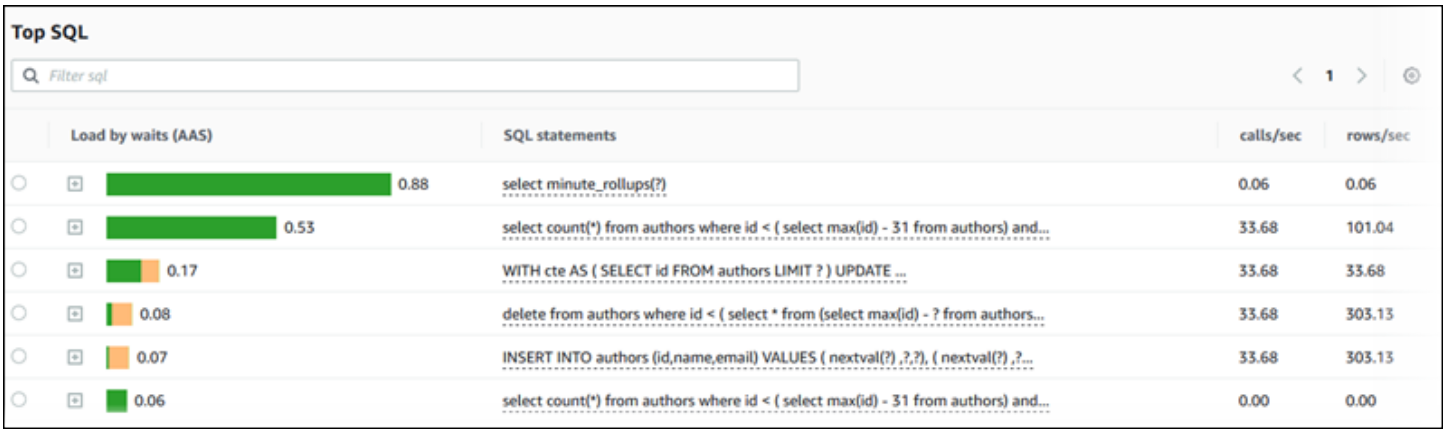
Performance Insights può mostrare il testo Oracle SQL come Unknown (Sconosciuto). Il testo ha questo stato nelle seguenti situazioni:

- Un utente di database Oracle diverso da SYS è attivo ma non esegue al momento SQL. Ad esempio, quando una query parallela viene completata, il coordinatore della query attende che i processi helper inviino le statistiche della sessione. Per tutta la durata dell'attesa, il testo della query risulta Unknown (Sconosciuto).
- Per un'istanza RDS per Oracle in Standard Edition 2, Oracle Resource Manager limita il numero di thread paralleli. Il processo in background che esegue questa attività fa sì che il testo della query venga visualizzato come Unknown (Sconosciuto).

Statistiche SQL

Statistiche SQL sono parametri relativi alle prestazioni relative alle query SQL. Ad esempio, Performance Insights potrebbe mostrare esecuzioni al secondo o righe elaborate al secondo. Performance Insights raccoglie statistiche solo per le query più comuni. In genere, queste query corrispondono alle prime query per carico mostrate nel dashboard di Performance Insights.

Tutte le righe della tabella Top SQL (Prime istruzioni SQL) mostra le statistiche rilevanti per l'istruzione SQL o il digest, come illustrato nell'esempio seguente.



Performance Insights può segnalare `0.00` e `-` (sconosciuto) per le statistiche SQL. Questa situazione si verifica nelle seguenti condizioni:

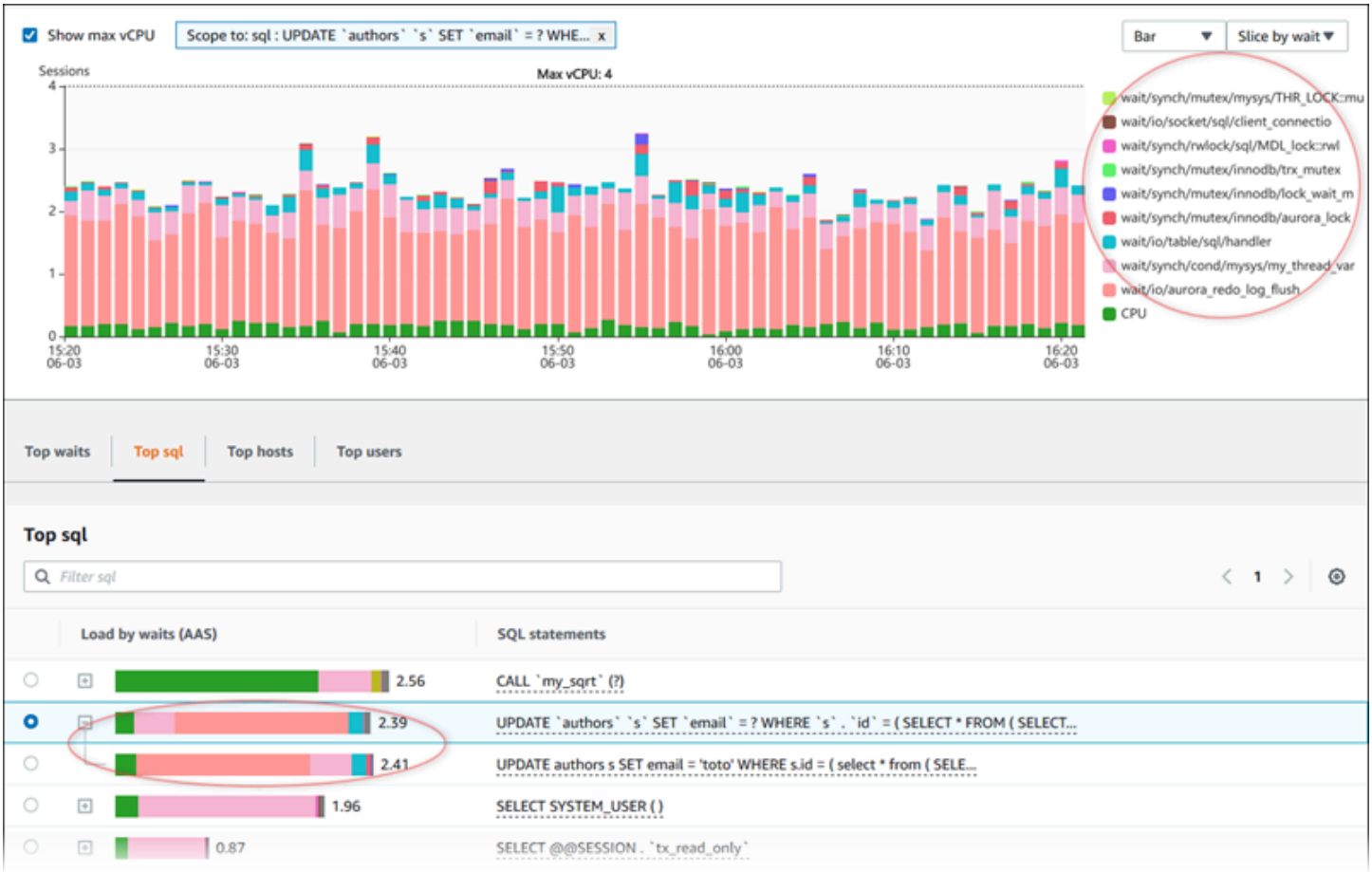
- Esiste un solo campione. Ad esempio, Performance Insights calcola i tassi di variazione per le query PostgreSQL di RDS sulla basa di molteplici campioni della vista `pg_stat_statements`. Quando un carico di lavoro viene eseguito per un breve periodo, Performance Insights potrebbe raccogliere solo un campione, il che significa che non è in grado di calcolare un tasso di varaizione. Il valore sconosciuto è rappresentato da un trattino (`-`).
- Due campioni hanno gli stessi valori. Performance Insights non è in grado di calcolare un tasso di variazione perché non si è verificata alcuna variazione, quindi riporta il tasso come `0.00`.
- Un'istruzione SQL RDS manca di un identificatore valido. PostgreSQL crea un identificatore per un'istruzione solo dopo la parsificazione e l'analisi. Pertanto, può esistere nelle strutture interne in memoria di PostgreSQL un'istruzione senza identificatore. Poiché Performance Insights esegue il campionamento delle strutture interne in memoria una volta al secondo, le query a bassa latenza potrebbero apparire solo in un singolo campione. Se l'identificatore della query non è disponibile per questo campione, Performance Insights non può associare questa istruzione alle relative statistiche. Il valore sconosciuto è rappresentato da un trattino (`-`).

Per una descrizione delle statistiche SQL per i motori Amazon RDS, consulta Statistiche SQL per Performance Insights.

Caricamento per attesa (AAS)

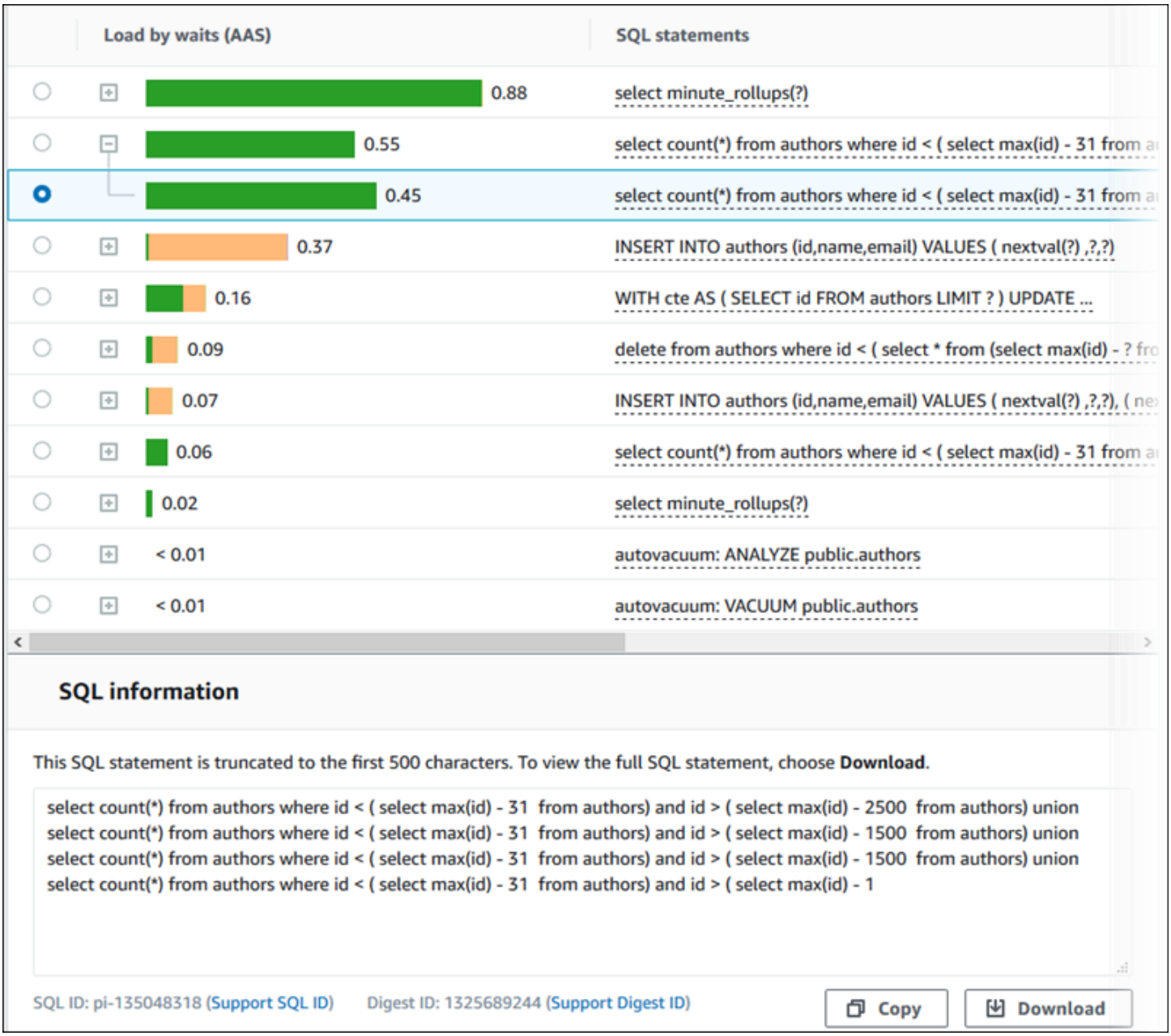
In Top SQL, la colonna Load by waits (AAS) illustra la percentuale del carico del database associato a ciascun elemento di caricamento superiore. Questa colonna indica il carico per questo elemento in base a qualunque raggruppamento attualmente selezionato nel grafico DB Load. Per ulteriori informazioni sulle sessioni attive medie (AAS), consulta Media delle sessioni attive.

Ad esempio, è possibile raggruppare il Carico DB in base agli stati di attesa. Esaminare le query SQL nella tabella degli elementi di caricamento superiore. In questo caso, la barra DB Load by Waits (Carico del database in base alle attese) è dimensionata, segmentata e rappresentata da un colore per mostrare qual è il contributo della query a un dato stato di attesa. Mostra anche quali stati di attesa stanno influenzando la query selezionata.



Informazioni SQL

Nella tabella Top SQL (Prime istruzioni SQL) è possibile aprire un'istruzione per visualizzarne le informazioni. Le informazioni vengono visualizzate nel riquadro inferiore.



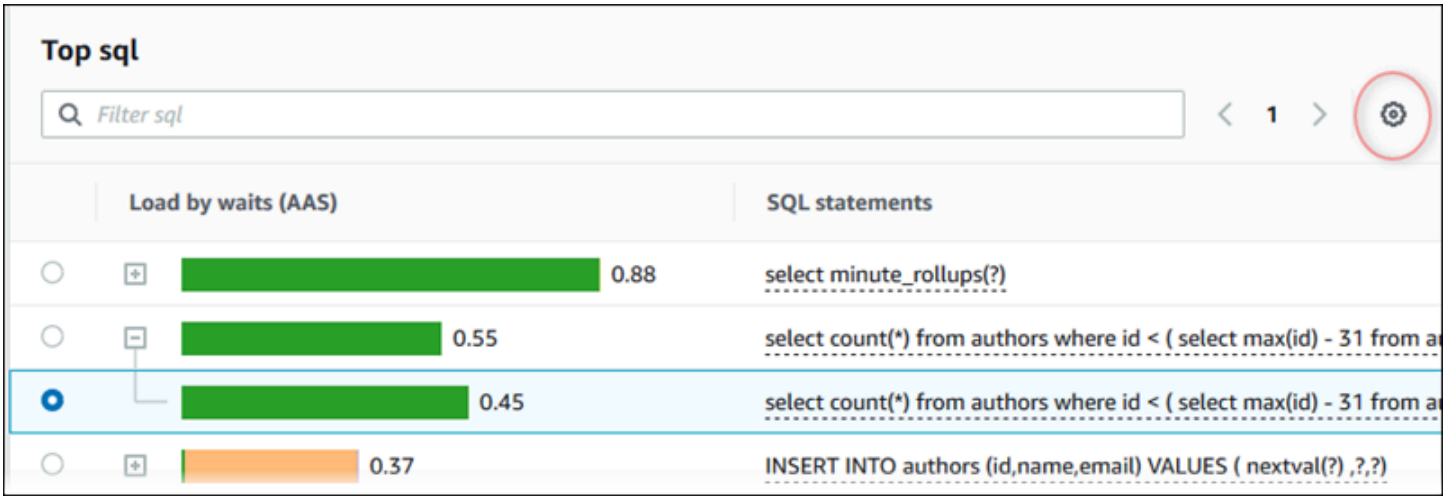
Puoi visualizzare i seguenti tipi di identificatori (ID) associati alle istruzioni SQL:

- ID SQL di supporto — Un valore hash dell'ID SQL. Questo valore serve solo per fare riferimento a un ID SQL quando si lavora con AWS Support. AWS Support non ha accesso agli ID SQL e al testo SQL effettivi.

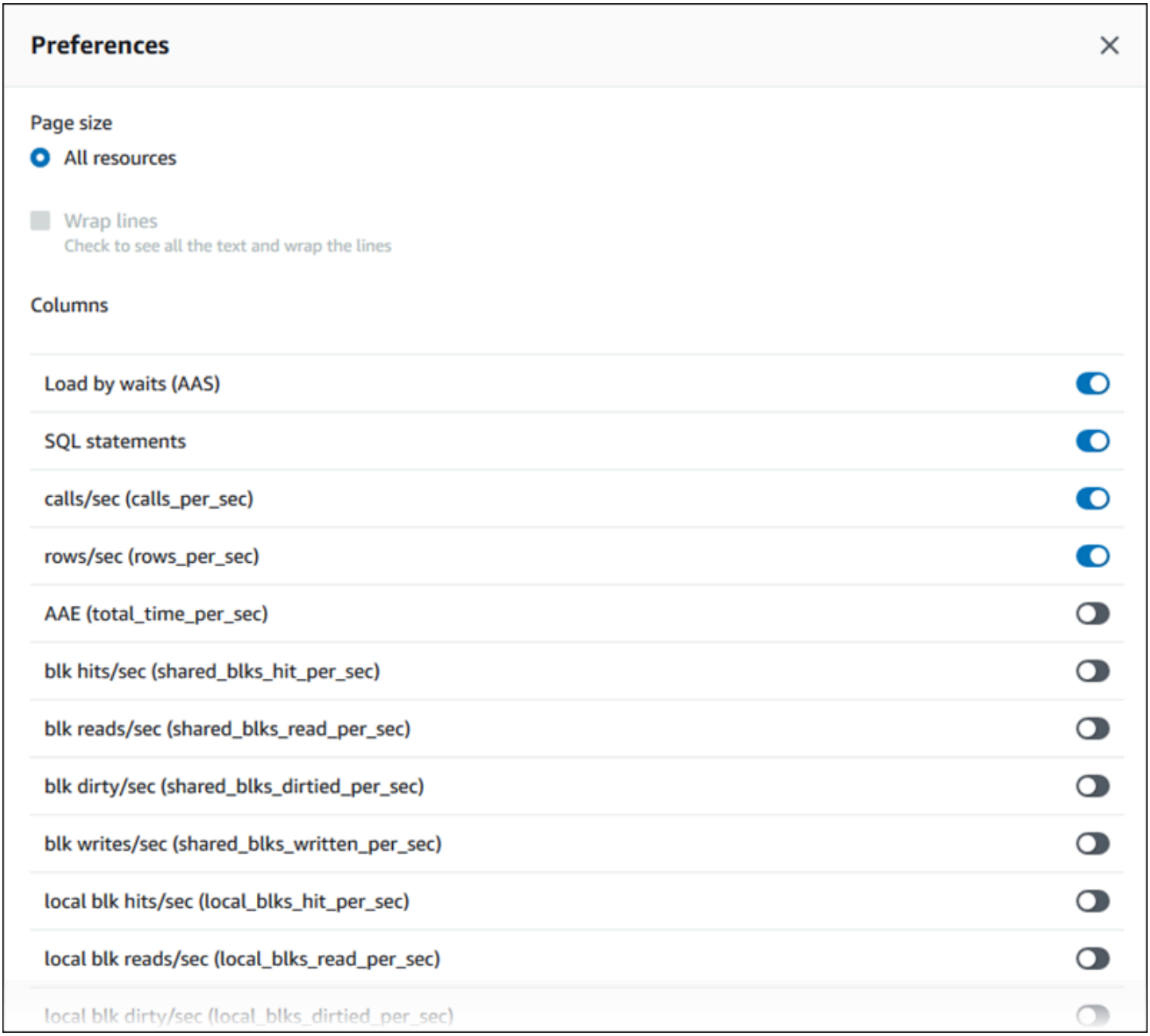
- ID Digest di supporto – Un valore hash dell'ID Digest. Questo valore serve solo per fare riferimento a un ID digest quando si lavora con Support AWS . AWS Support non ha accesso agli ID digest e al testo SQL effettivi.

Preferenze

È possibile controllare le statistiche visualizzate nella scheda Top SQL (Prime istruzioni SQL) scegliendo l'icona Preferenze.



Quando scegli l'icona Preferences (Preferenze), viene visualizzata la finestra Preferences (Preferenze). La schermata seguente è un esempio della finestra Preferences (Preferenze).



Abilitare le statistiche che si desidera visualizzare nella scheda Top SQL (Prime istruzioni SQL), utilizzare il mouse per scorrere fino alla fine della finestra, quindi scegliere Continua.

Per ulteriori informazioni sulle statistiche per secondo o per chiamata per i motori Amazon RDS, consulta la sezione delle statistiche SQL specifiche del motore in Statistiche SQL per Performance Insights

Accesso a una maggiore quantità di testo SQL nel pannello di controllo di Performance Insights

Per impostazione predefinita, ciascuna riga nella tabella Top SQL (Prime istruzioni SQL) mostra 500 byte di testo SQL per ciascuna istruzione SQL.

Quando un'istruzione SQL supera i 500 byte, puoi visualizzare più testo nella sezione SQL text (Testo SQL) sotto la tabella Top SQL (Prime istruzioni SQL). In questo caso, la lunghezza massima per il testo visualizzato in SQL text (Testo SQL) è 4 KB. Questo limite viene introdotto dalla console ed è soggetto ai limiti impostati dal motore del database. Per salvare il testo visualizzato in SQL text (Testo SQL), scegli Download (Scarica).

Argomenti

- Limiti delle dimensioni del testo per i motori Amazon RDS
- Impostazione del limite di testo SQL per le istanze database Amazon RDS for PostgreSQL
- Visualizzazione e download del testo SQL nel pannello di controllo di Performance Insights

Limiti delle dimensioni del testo per i motori Amazon RDS

Durante il download di testo SQL, il motore del database determina la sua lunghezza massima. Puoi scaricare il testo SQL fino ai seguenti limiti per motore.

Motore database	Lunghezza massima del testo scaricato
Amazon RDS per MySQL e MariaDB	1.024 byte
Amazon RDS for Microsoft SQL Server	4,096 caratteri
Amazon RDS per Oracle	1.000 byte

La sezione SQL Text (Testo SQL) della console Performance Insights visualizza fino al massimo restituito dal motore. Ad esempio, se MySQL restituisce al massimo 1 KB a Performance Insights, può raccogliere e mostrare solo 1 KB, anche se la query originale è più grande. Pertanto, quando la query viene visualizzata in SQL text (Testo SQL) o scaricata, Performance Insights restituisce lo stesso numero di byte.

Se utilizzi l'API AWS CLI o, Performance Insights non ha il limite di 4 KB imposto dalla console. `DescribeDimensionKeyse GetResourceMetrics` restituiscono al massimo 500 byte.

> **Note**
>
> `GetDimensionKeyDetails`restituisce la query completa, ma la dimensione è soggetta al limite del motore.

Impostazione del limite di testo SQL per le istanze database Amazon RDS for PostgreSQL

Amazon RDS for PostgreSQL gestisce il testo in modo diverso. È possibile impostare il limite delle dimensioni del testo con il parametro di istanza database `track_activity_query_size`. Questo parametro presenta le caratteristiche seguenti:

Dimensione di default del testo

Su Aurora Amazon RDS for PostgreSQL versione 9.6, l'impostazione di default per il parametro `track_activity_query_size` è 1.024 byte. Su Amazon RDS for PostgreSQL versione 10 o successive, l'impostazione di default per il parametro è 4.096 byte.

Dimensione massima del testo

Il limite per `track_activity_query_size` è 102.400 byte per Amazon RDS per PostgreSQL versione 12 e versioni precedenti. Il massimo è di 1 MB per la versione 13 e quelle successive.

Se il motore restituisce 1 MB a Performance Insights, la console visualizza solo i primi 4 KB. Se si scarica la query, si ottiene 1 MB per intero. In questo caso, la visualizzazione e il download restituiscono un numero diverso di byte. Per ulteriori informazioni sul parametro dell'istanza database `track_activity_query_size`, consulta Run-time Statistics nella documentazione di PostgreSQL.

Per aumentare la dimensione del testo SQL, aumenta il limite di `track_activity_query_size`. Per modificare il parametro, modifica l'impostazione del parametro nel gruppo di parametri associato all'istanza database Amazon RDS for PostgreSQL.

Modifica dell'impostazione quando l'istanza utilizza il gruppo di parametri di default

1. Crea un nuovo gruppo di parametri dell'istanza database per il motore del database e la versione del motore del database appropriati.

2. Imposta il parametro nel nuovo gruppo di parametri.
3. Associa il nuovo gruppo di parametri all'istanza database.

Per ulteriori informazioni sull'impostazione di un parametro dell'istanza database, consulta Modifica di parametri in un gruppo di parametri del database.

Visualizzazione e download del testo SQL nel pannello di controllo di Performance Insights

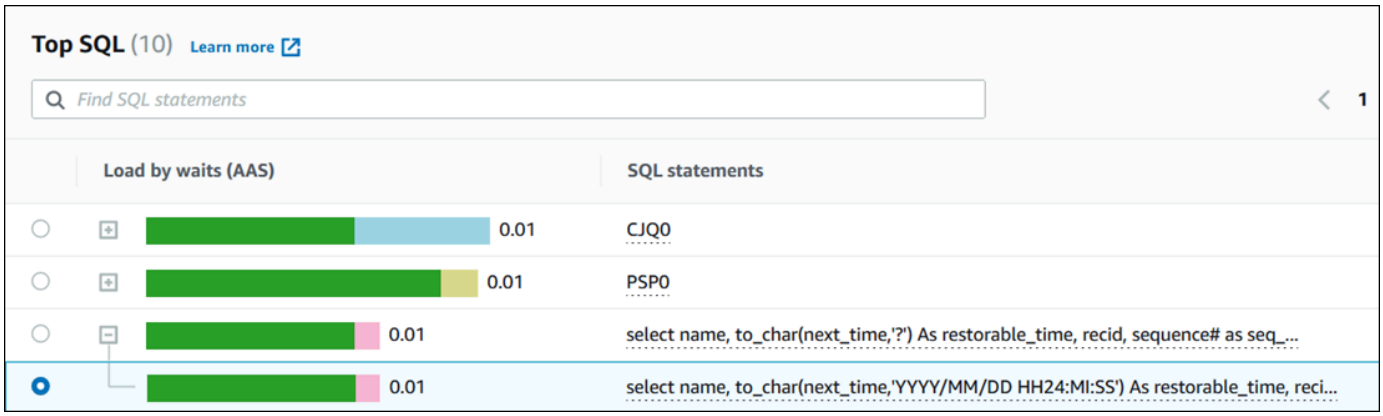
Nel pannello di controllo di Performance Insights è possibile visualizzare e scaricare il testo SQL.

Per visualizzare una maggiore quantità di testo SQL nel pannello di controllo di Performance Insights

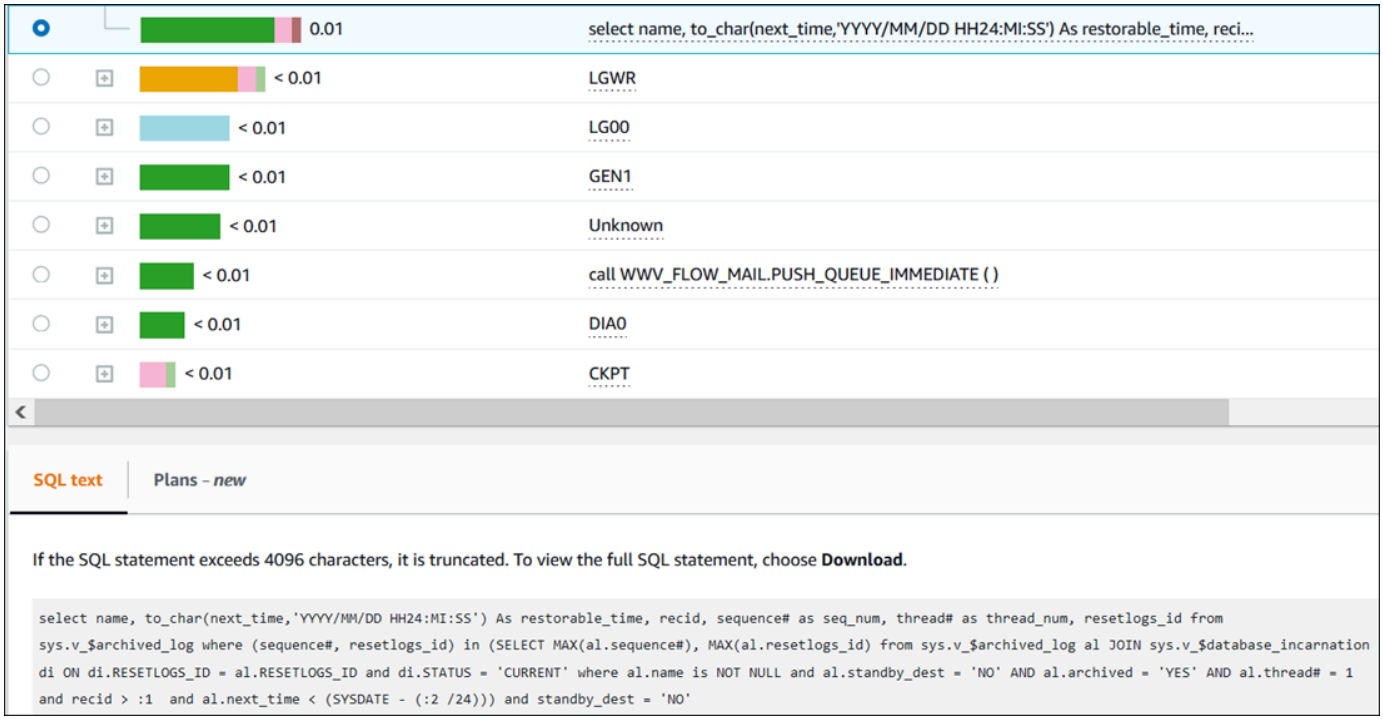
1. Apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel pannello di navigazione scegli Performance Insights.
3. Scegli un'istanza database.

   Viene visualizzato il pannello di controllo di Performance Insights per l'istanza database.
4. Scorri verso il basso fino alla scheda Top SQL (Prime istruzioni SQL).
5. Scegli il segno più per espandere un digest SQL e scegli una delle query secondarie del digest.

   Le istruzioni SQL con testo superiore a 500 byte sono simili a quelle nell'immagine seguente.



6. Scorri verso il basso fino alla scheda Testo SQL.



Il pannello di controllo di Performance Insights può visualizzare fino a 4.096 byte per ciascuna istruzione SQL.

7. (Facoltativo) Scegliere Copia per copiare l'istruzione SQL visualizzata oppure scegliere Scarica per scaricare l'istruzione SQL e visualizzare il testo SQL fino al limite del motore database.

> **Note**
>
> Per copiare o scaricare l'istruzione SQL, disattiva i sistemi di blocco popup.

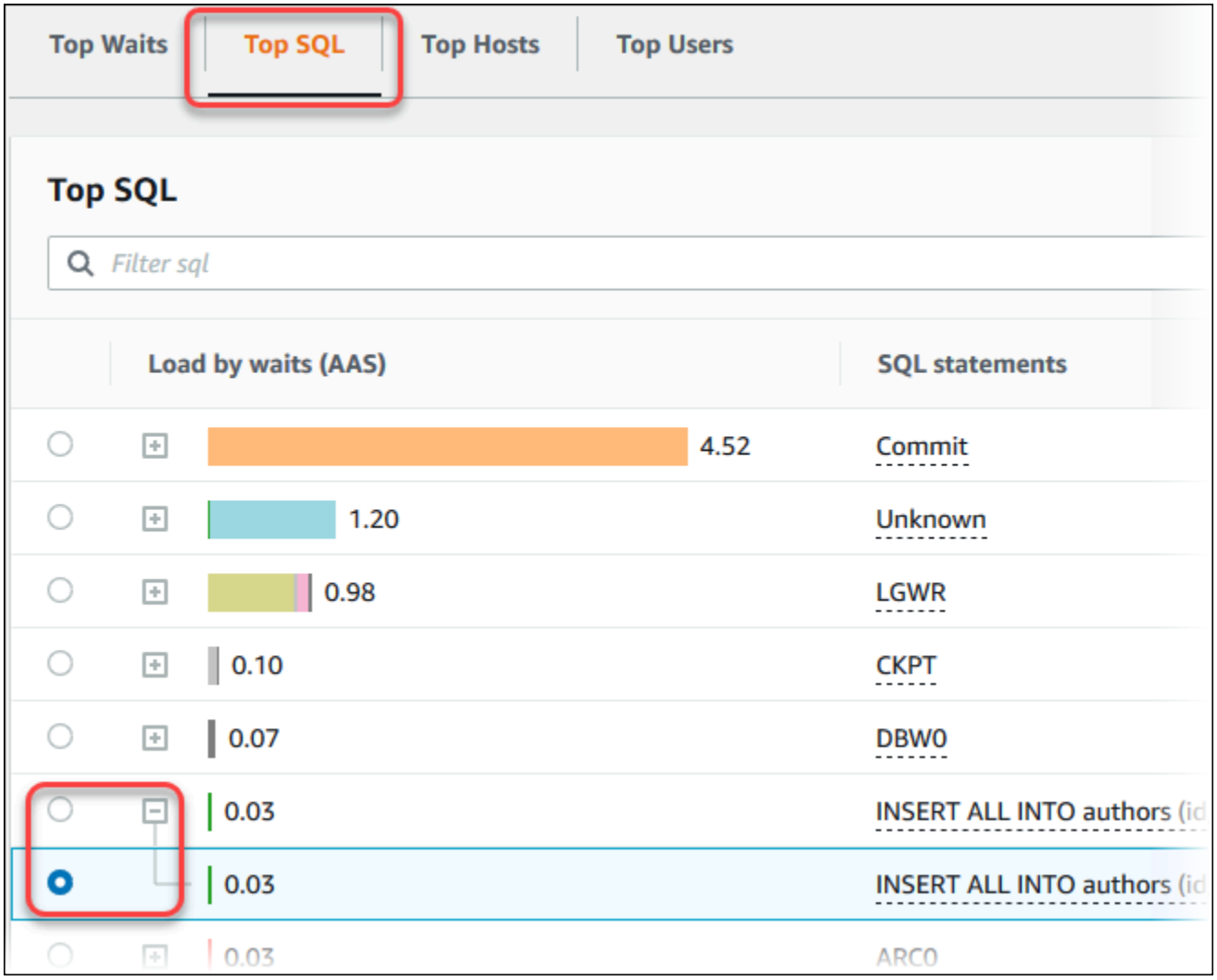
Visualizzazione delle statistiche SQL nel pannello di controllo di Performance Insights

Nel pannello di controllo di Performance Insights, le statistiche SQL sono disponibili nella scheda Top SQL (Prime istruzioni SQL) del grafico Database load (Carico database).

Per visualizzare le statistiche SQL

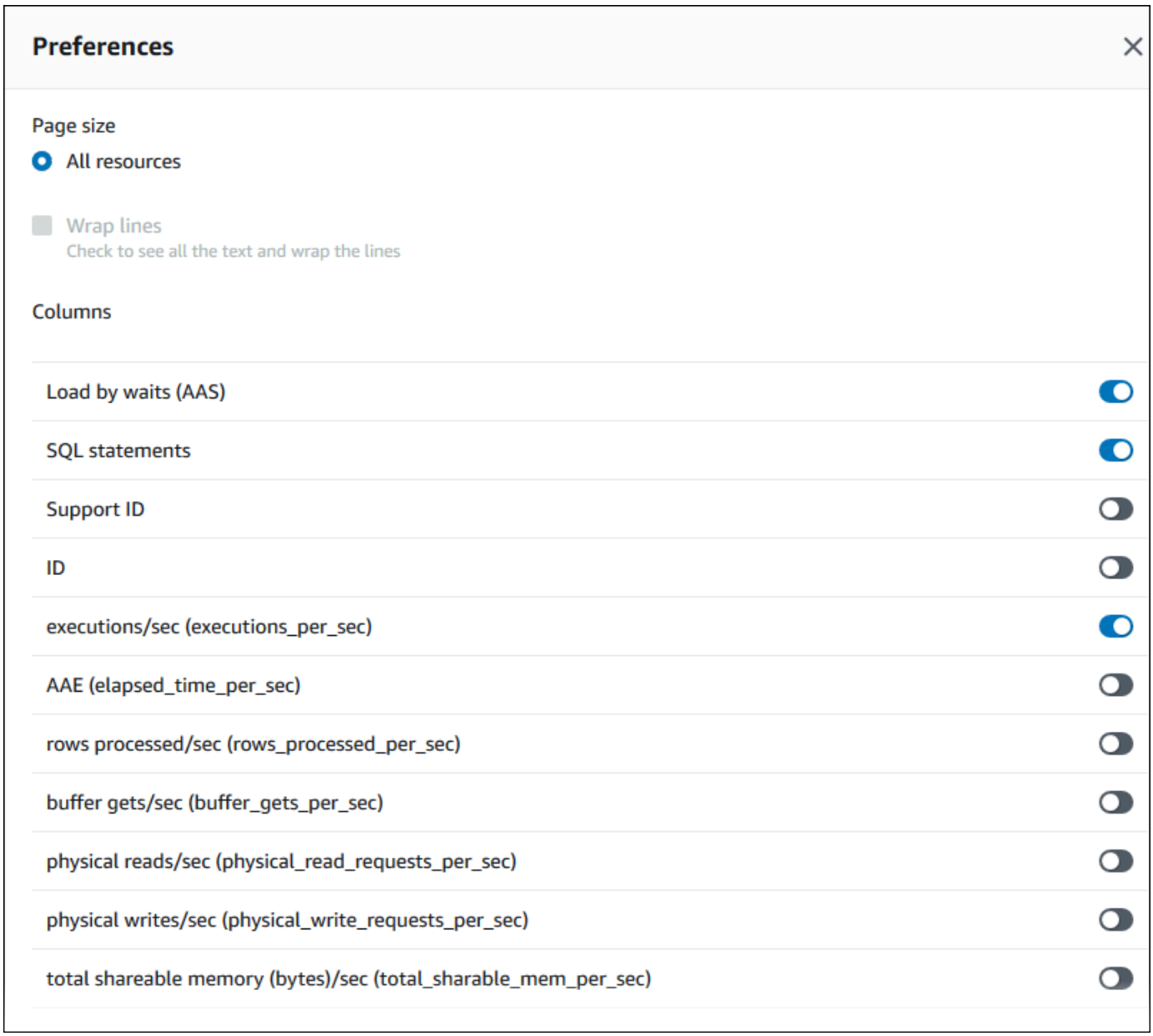
1. Apri la console di Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel pannello di navigazione scegli Performance Insights.
3. Nella parte superiore della pagina, scegli il database di cui desideri visualizzare le statistiche SQL.

4. Scorrere fino alla parte inferiore della pagina e scegli Top SQL (Prime istruzioni SQL).
5. Scegli una specifica istruzione o un sunto di una query.

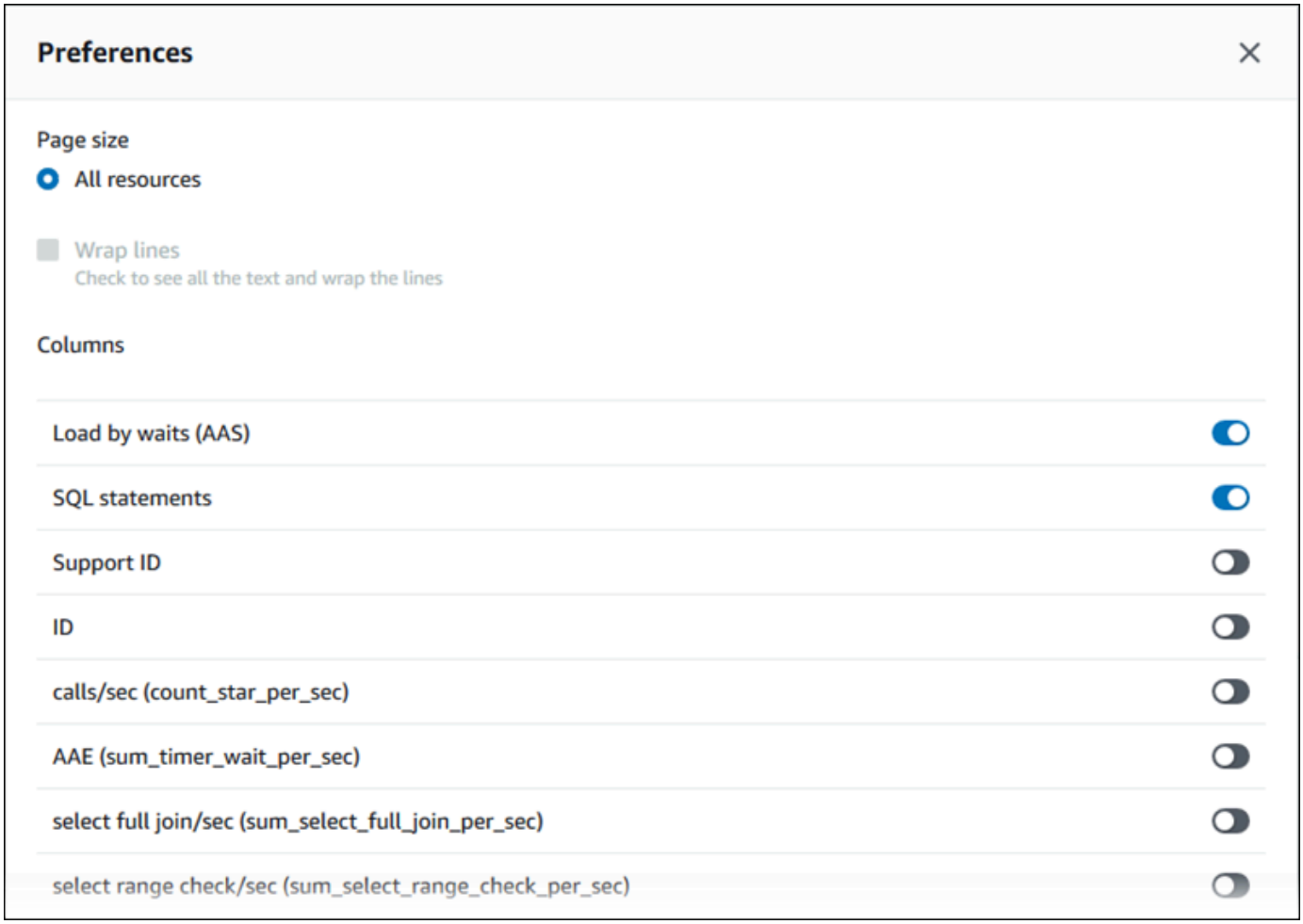


6. Scegliere le statistiche da visualizzare selezionando l'icona a forma di ingranaggio nell'angolo in alto a destra del grafico. Per le descrizioni delle statistiche SQL per i motori Amazon RDS, consulta Statistiche SQL per Performance Insights.

   L'esempio seguente mostra le preferenze per statistiche per le istanze database Oracle.



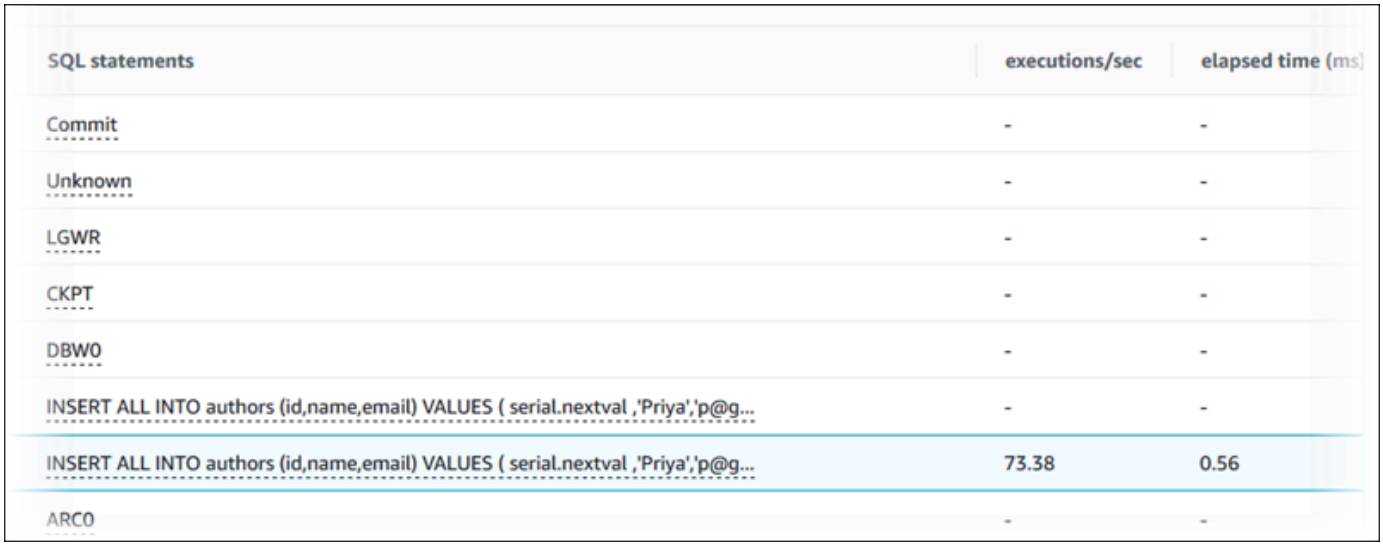
Il seguente esempio mostra le preferenze per le istanze database MariaDB e MySQL.



7. Per salvare le preferenze, scegli Save (Salva).

   La tabella Top SQL (Prime istruzioni SQL) si aggiorna.

   L'esempio seguente mostra le statistiche per una query Oracle SQL.



Analisi del carico principale di Oracle PDB

Quando analizzi il carico su un Oracle Container DB (CDB), potresti voler identificare quali database collegabili (PDB) contribuiscono maggiormente al carico del DB. Potresti anche voler confrontare le prestazioni di singoli PDB che eseguono query simili per ottimizzare le prestazioni. Per ulteriori informazioni su Oracle CDB, vedere. Architettura del database RDS per Oracle

Nel pannello di controllo di Amazon RDS Performance Insights, puoi trovare informazioni sui database collegabili (PDB) nella scheda Top PDB nella scheda Dimensioni.

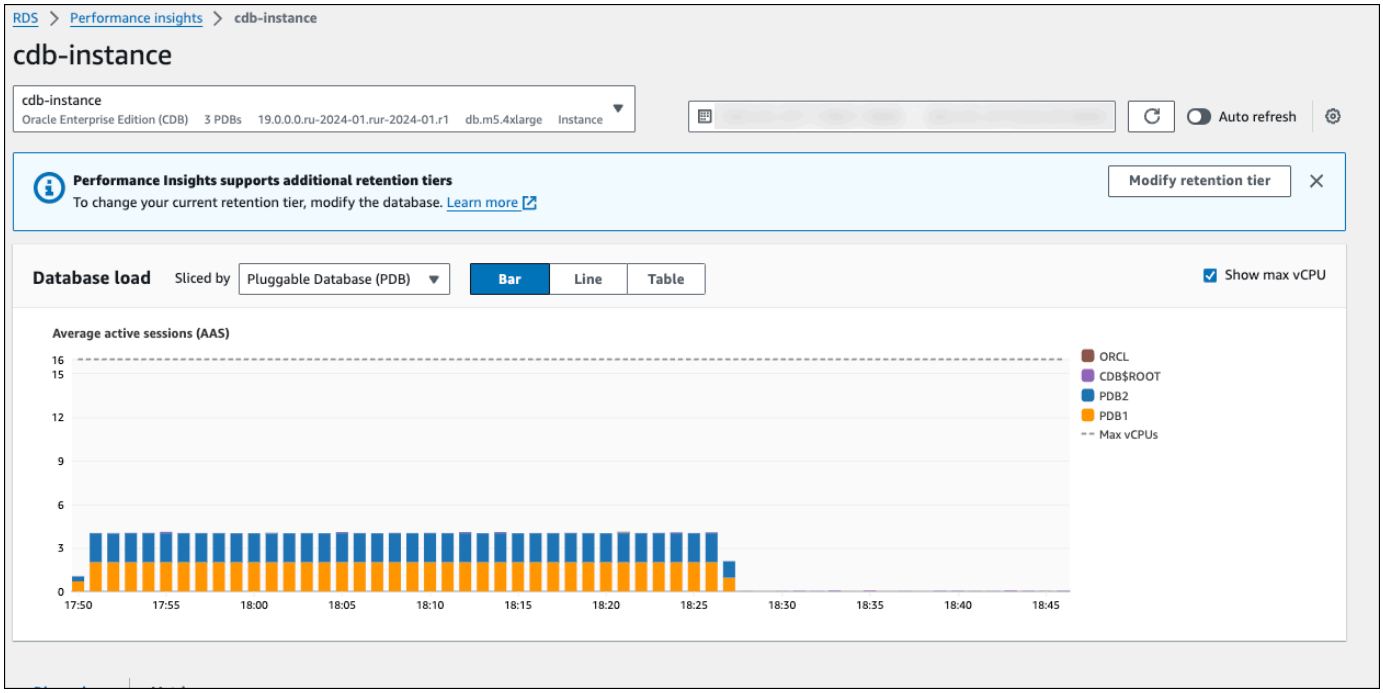
Per informazioni sulla regione, sul motore DB e sulla classe di istanze per questa funzionalità, consulta. Supporto di classe di istanza, regione e motore di database Amazon RDS per funzionalità Performance Insights

Per analizzare il carico PDB principale in un CDB Oracle

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione a sinistra, seleziona Performance Insights.
3. Scegli un'istanza Oracle CDB.

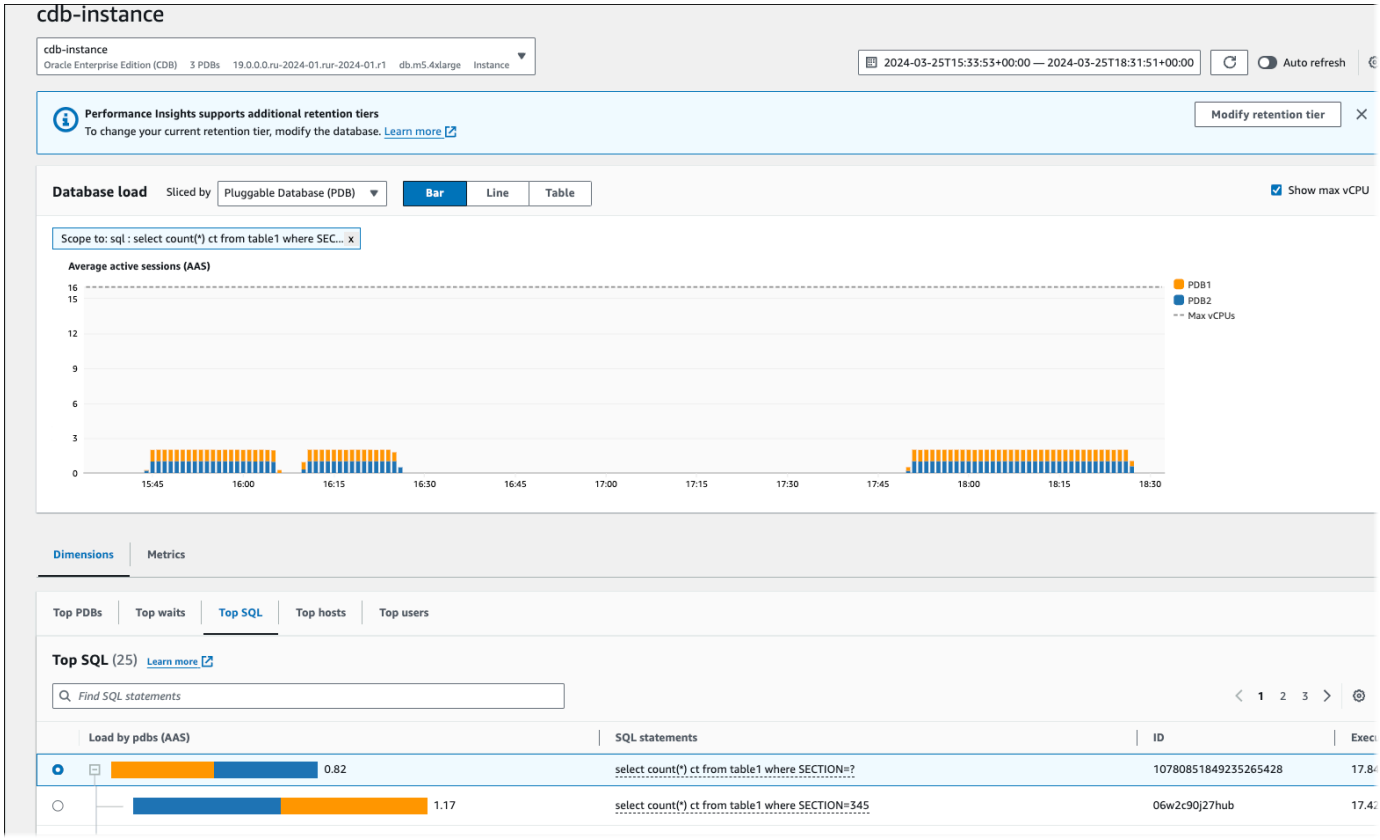
   Viene visualizzato il pannello di controllo di Approfondimenti sulle prestazioni per l'istanza database.
4. Nella sezione Caricamento del database (caricamento DB), scegli Database Pluggable (PDB) accanto a Slice by.

Il grafico delle sessioni attive medie mostra il PDB con il carico più elevato. Gli identificatori PDB vengono visualizzati a destra dei quadrati con codice colore. Ogni identificatore identifica in modo univoco un PDB.

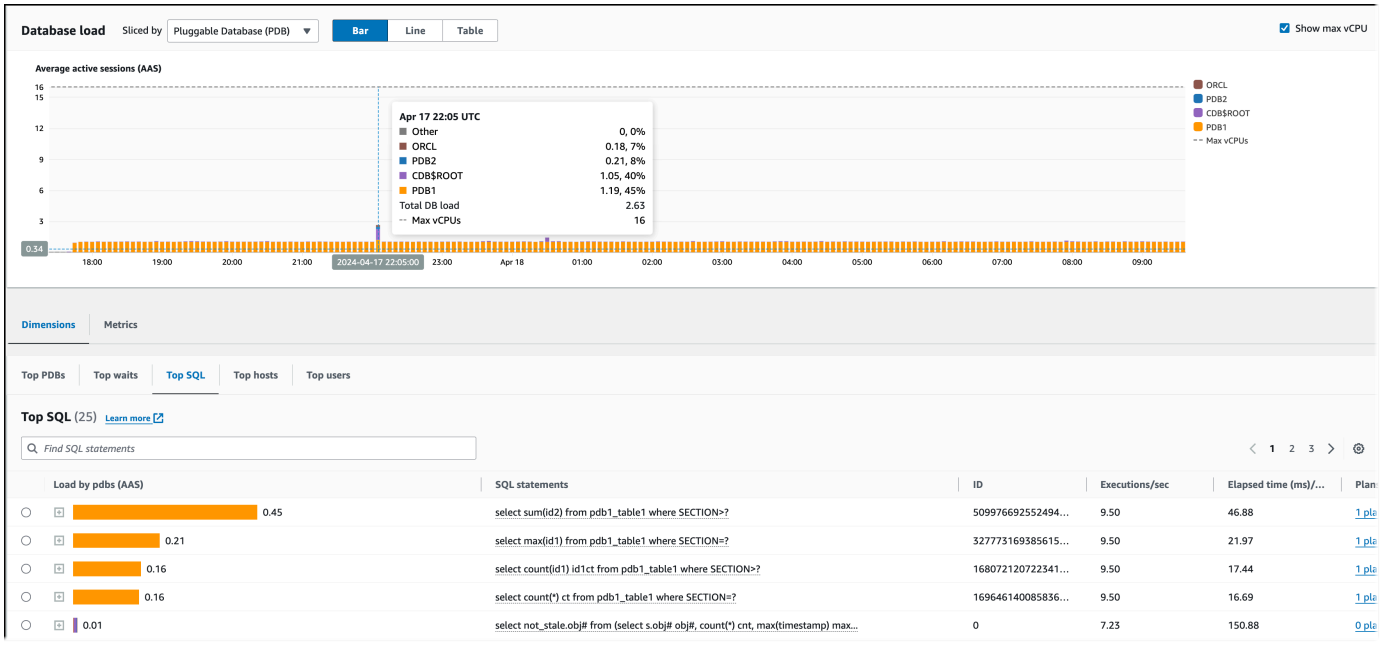


5. Scorri verso il basso fino alla scheda Top SQL (Prime istruzioni SQL).

   Nell'esempio seguente, è possibile visualizzare la stessa query SQL e il carico che genera su più PDB.



Nell'esempio seguente, un singolo PDB gestisce un carico maggiore rispetto ad altri PDB nel CDB.



Per ulteriori informazioni sugli Oracle CDB, consulta CDB e PDB.

Analisi dei piani di esecuzione utilizzando la dashboard di Performance Insights

Nella dashboard di Amazon RDS Performance Insights, puoi trovare informazioni sui piani di esecuzione per le istanze DB di Oracle e SQL Server. Puoi utilizzare queste informazioni per sapere quali piani contribuiscono maggiormente al carico del DB.

Analisi dei piani di esecuzione

- Panoramica dell'analisi dei piani di esecuzione
- Analisi dei piani di esecuzione di Oracle tramite il pannello di controllo di Performance Insights
- Analisi dei piani di esecuzione di SQL Server utilizzando il dashboard di Performance Insights

Panoramica dell'analisi dei piani di esecuzione

Puoi utilizzare la dashboard di Amazon RDS Performance Insights per sapere quali piani contribuiscono maggiormente al carico del DB per le istanze DB di Oracle e SQL Server.

Ad esempio, le istruzioni SQL principali in un determinato momento potrebbero utilizzare i piani mostrati nella tabella seguente.

Prime istruzioni SQL	Pianificazione
SELECT SUM(amount_sold) FROM sales WHERE prod_id = 10	Piano A
SELECT SUM(amount_sold) FROM sales WHERE prod_id = 521	Piano B
SELECT SUM(s_total) FROM sales WHERE region = 10	Piano A
SELECT * FROM emp WHERE emp_id = 1000	Piano C
SELECT SUM(amount_sold) FROM sales WHERE prod_id = 72	Piano A

Con la caratteristica di pianificazione di Performance Insights, è possibile effettuare le seguenti operazioni:

- Scoprire quali piani vengono utilizzati dalle principali query SQL.

Ad esempio, potresti scoprire che la maggior parte del carico del database viene generato da query che utilizzano il piano A e il piano B, con solo una piccola percentuale che utilizza il piano C.

- Confrontare piani diversi per la stessa query.

  Nell'esempio precedente, tre query sono identiche a eccezione dell'ID del prodotto. Due query utilizzano il piano A, ma una query utilizza il piano B. Per vedere la differenza tra i due piani, è possibile utilizzare Performance Insights.

- Scoprire quando una query è passata a un nuovo piano.

  È possibile che una query utilizzasse il piano A e poi è passata al piano B in un determinato momento. Si è verificato un cambiamento nel database a questo punto? Ad esempio, se una tabella è vuota, l'ottimizzatore potrebbe scegliere una scansione completa della tabella. Se la tabella viene caricata con un milione di righe, l'ottimizzatore potrebbe passare a una scansione dell'intervallo su indice.

- Esaminare le fasi specifiche di un piano con il costo più alto.

  Ad esempio, la query per una lunga durata potrebbe mostrare la mancanza di una condizione di join in un equi-join. Questa condizione mancante impone un'unione cartesiana, che unisce tutte le righe di due tabelle.

È possibile eseguire le attività precedenti utilizzando la caratteristica di acquisizione del piano di Performance Insights. Proprio come è possibile suddividere le query in base agli eventi di attesa e alle prime istruzioni SQL, è possibile suddividerle in base alla dimensione del piano.

Analisi dei piani di esecuzione di Oracle tramite il pannello di controllo di Performance Insights

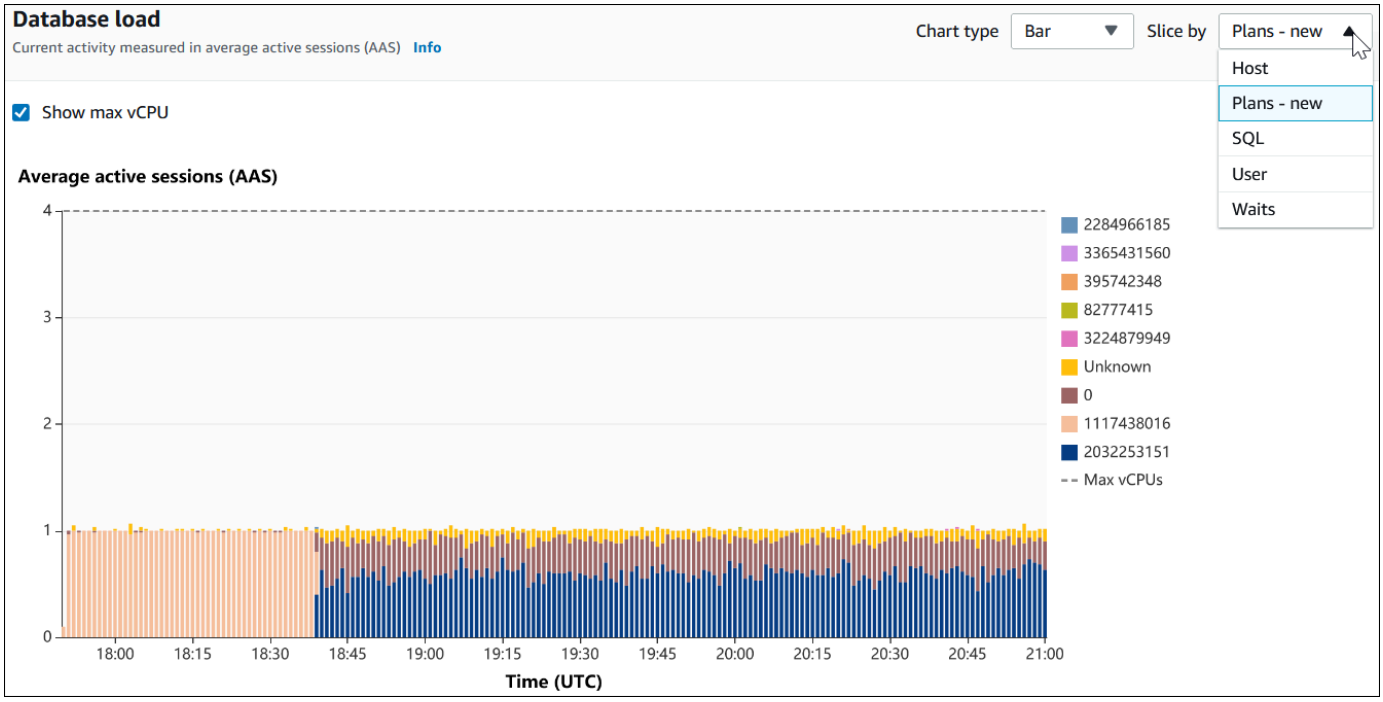
Quando si analizza il carico del database su un database Oracle, è possibile sapere quali piani contribuiscono maggiormente al carico del database. È possibile determinare quali piani contribuiscono maggiormente al carico del DB utilizzando la funzionalità di acquisizione dei piani di Performance Insights.

Per analizzare i piani di esecuzione di Oracle utilizzando la console

1. Apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel pannello di navigazione scegli Performance Insights.
3. Scegli un'istanza database di Oracle. Viene visualizzato il pannello di controllo di Performance Insights per l'istanza database.

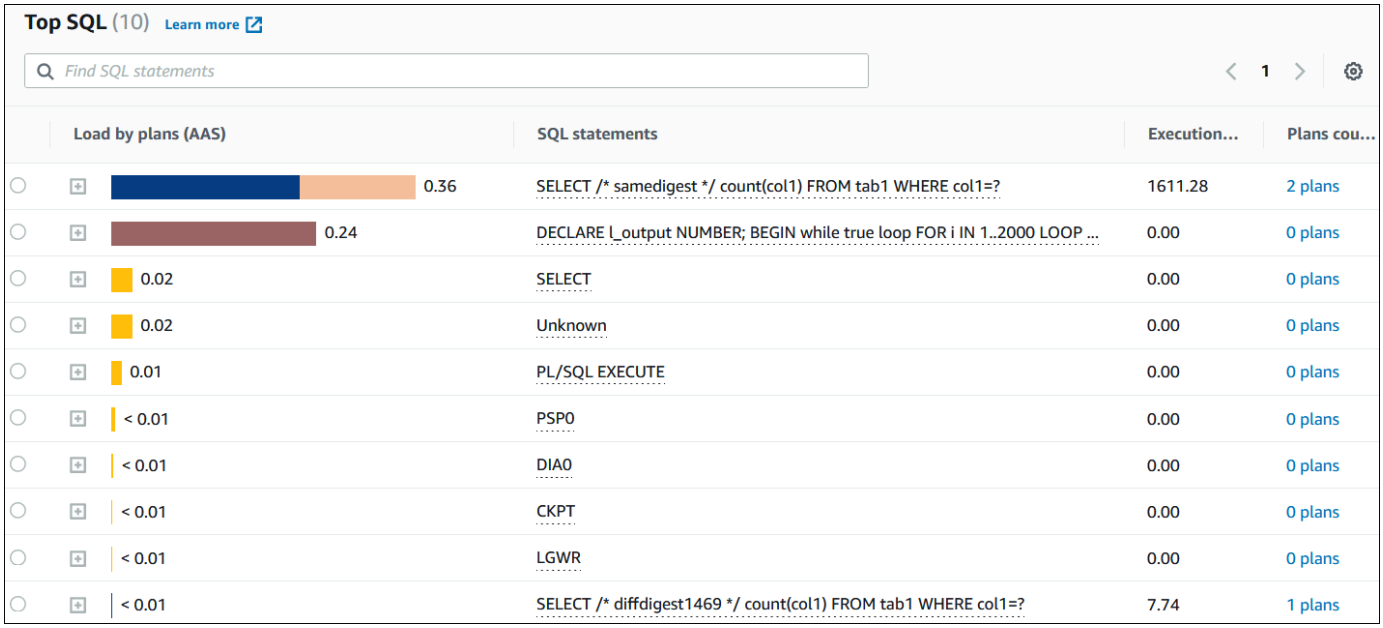
4. Nella sezione Database load (DB load) (Carico del database (Carico DB)), scegli Plans (Piani) accanto a Slice by (Dividi per).

   Il grafico Average active sessions (Media delle sessioni attive) mostra i piani utilizzati dalle istruzioni principali SQL. I valori hash del piano appaiono a destra dei quadrati con codice colore. Ogni valore hash identifica in modo univoco un piano.



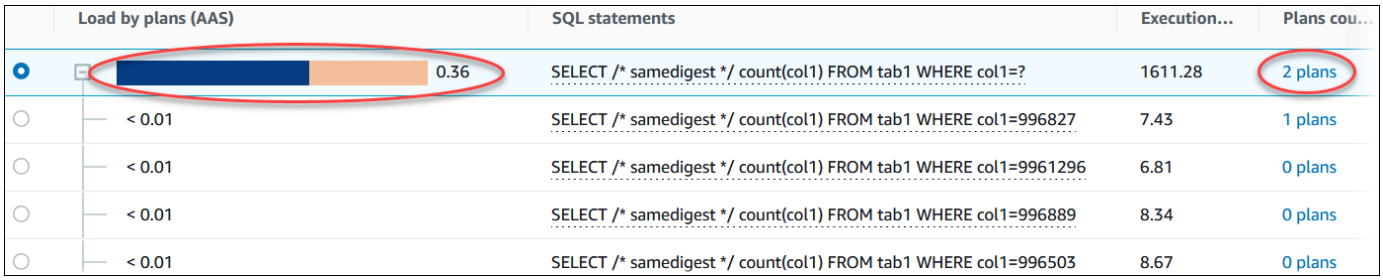
5. Scorri verso il basso fino alla scheda Top SQL (Prime istruzioni SQL).

   Nell'esempio seguente, il digest delle prime istruzioni SQL ha due piani. Si può dire che è un digest dal punto interrogativo nell'istruzione.



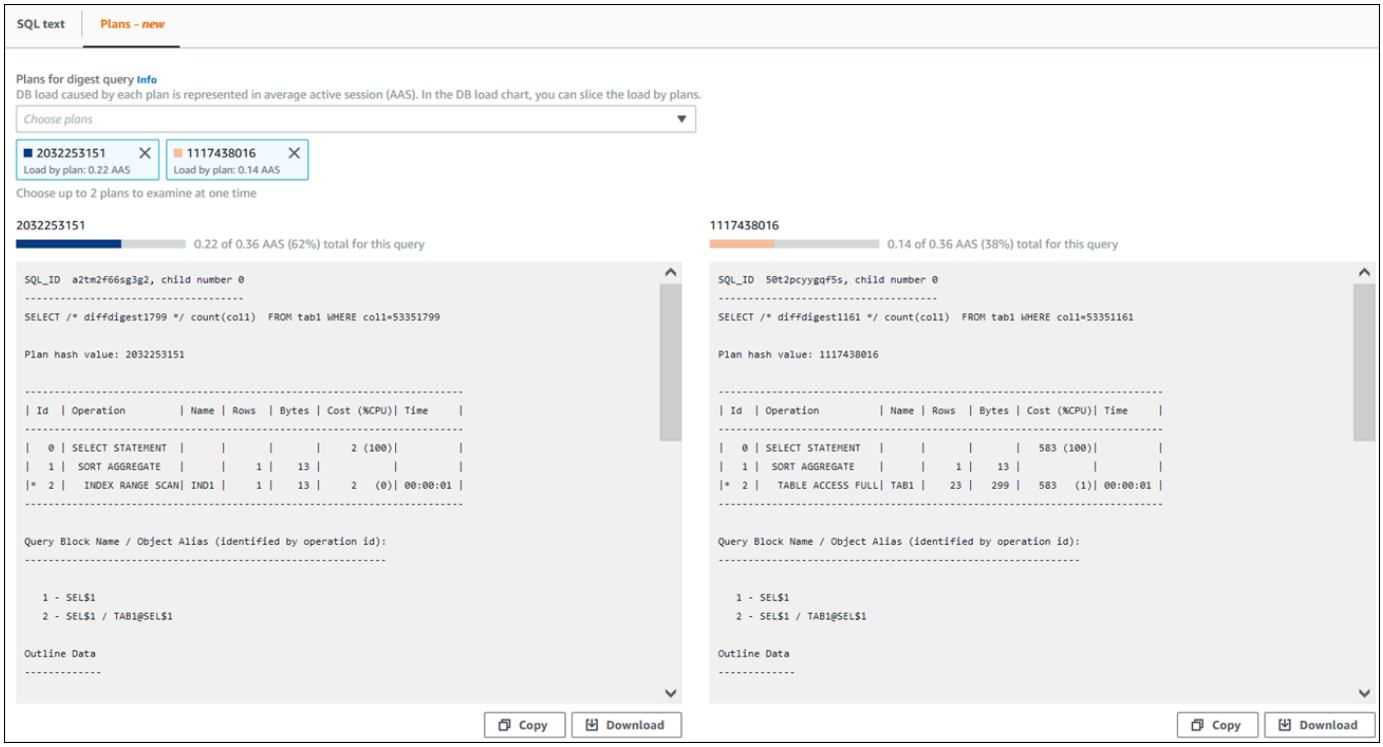
6. Scegli il digest per espanderlo nelle istruzioni componenti.

   Nell'esempio seguente, l'istruzione SELECT è una query digest. Le query dei componenti nel digest utilizzano due piani diversi. I colori dei piani corrispondono al grafico del carico del database. Il numero totale di piani nel digest è mostrato nella seconda colonna.

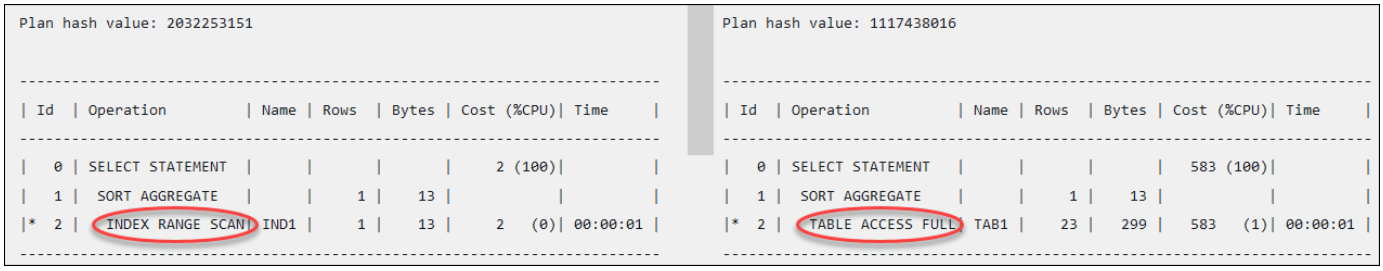


7. Scorri in basso e scegli due Plans (Piani) per il confronto dall'elenco Plans for digest query (Piani per la query digest).

   È possibile visualizzare uno o due piani per una query alla volta. Lo screenshot seguente confronta i due piani nel digest, con l'hash 2032253151 e l'hash 1117438016. Nell'esempio seguente, il 62% delle sessioni attive medie che eseguono questa query digest utilizza il piano a sinistra, mentre il 38% utilizza il piano a destra.



In questo esempio, i piani differiscono in modo importante. Il passaggio 2 del piano 2032253151 utilizza una scansione dell'indice, mentre il piano 1117438016 utilizza una scansione completa della tabella. Per una tabella con un numero elevato di righe, una query di una singola riga è quasi sempre più veloce con una scansione dell'indice.



8. (Facoltativo) Scegli Copy (Copia) per copiare il piano negli appunti, oppure Download (Scarica) per salvare il piano sul disco rigido.

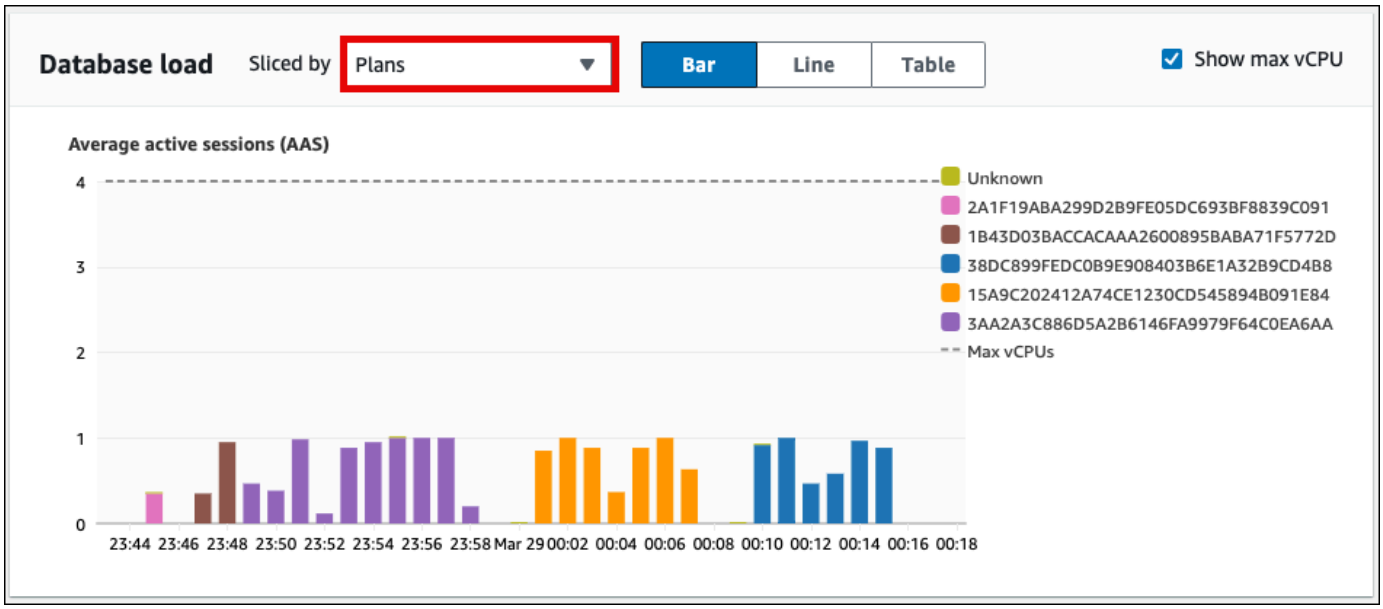
Analisi dei piani di esecuzione di SQL Server utilizzando il dashboard di Performance Insights

Quando analizzi il carico del DB su un database di SQL Server, potresti voler sapere quali piani contribuiscono maggiormente al carico del DB. È possibile determinare quali piani contribuiscono maggiormente al carico del DB utilizzando la funzionalità di acquisizione dei piani di Performance Insights.

Per analizzare i piani di esecuzione di SQL Server utilizzando la console

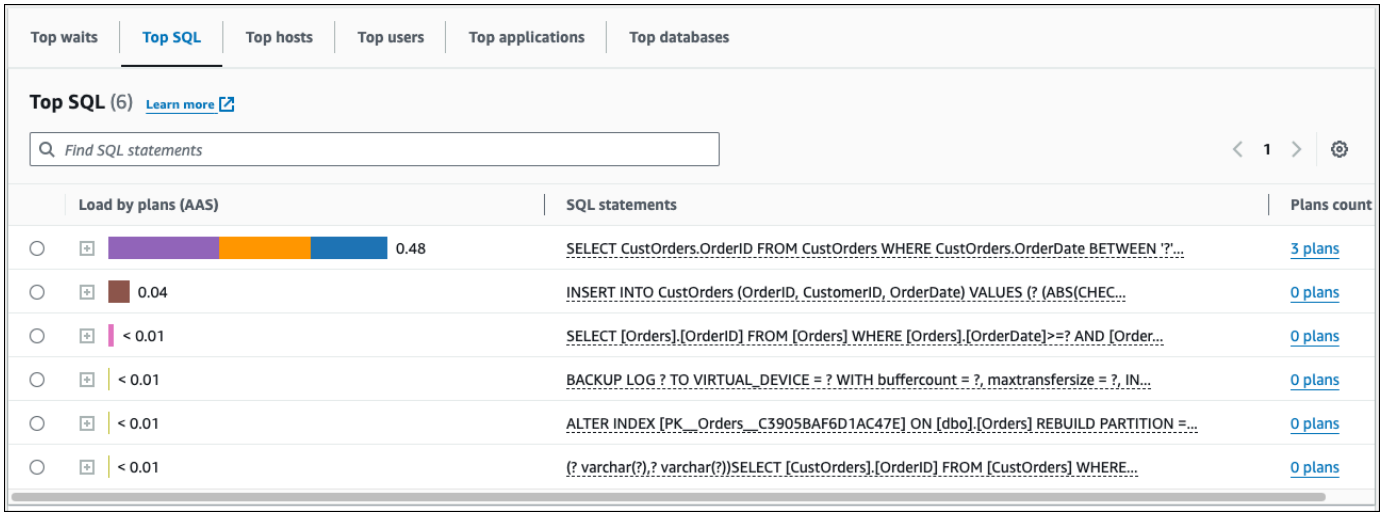
1. Apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel pannello di navigazione scegli Performance Insights.
3. Scegli un'istanza DB di SQL Server. Viene visualizzato il pannello di controllo di Performance Insights per l'istanza database.
4. Nella sezione Database load (DB load) (Carico del database (Carico DB)), scegli Plans (Piani) accanto a Slice by (Dividi per).

   Il grafico Average active sessions (Media delle sessioni attive) mostra i piani utilizzati dalle istruzioni principali SQL. I valori hash del piano appaiono a destra dei quadrati con codice colore. Ogni valore hash identifica in modo univoco un piano.



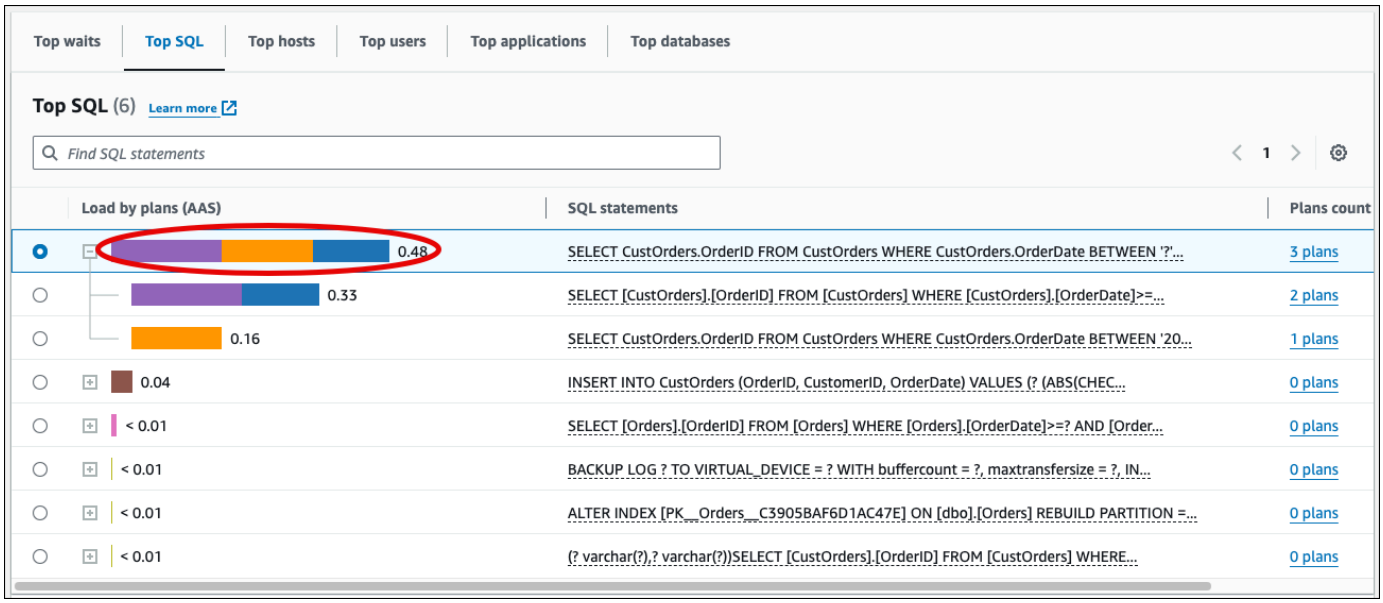
5. Scorri verso il basso fino alla scheda Top SQL (Prime istruzioni SQL).

   Nell'esempio seguente, il top SQL digest ha tre piani. La presenza di un punto interrogativo nell'istruzione SQL indica che l'istruzione è un digest. Per visualizzare l'istruzione SQL completa, scegliete un valore nella colonna Istruzioni SQL.



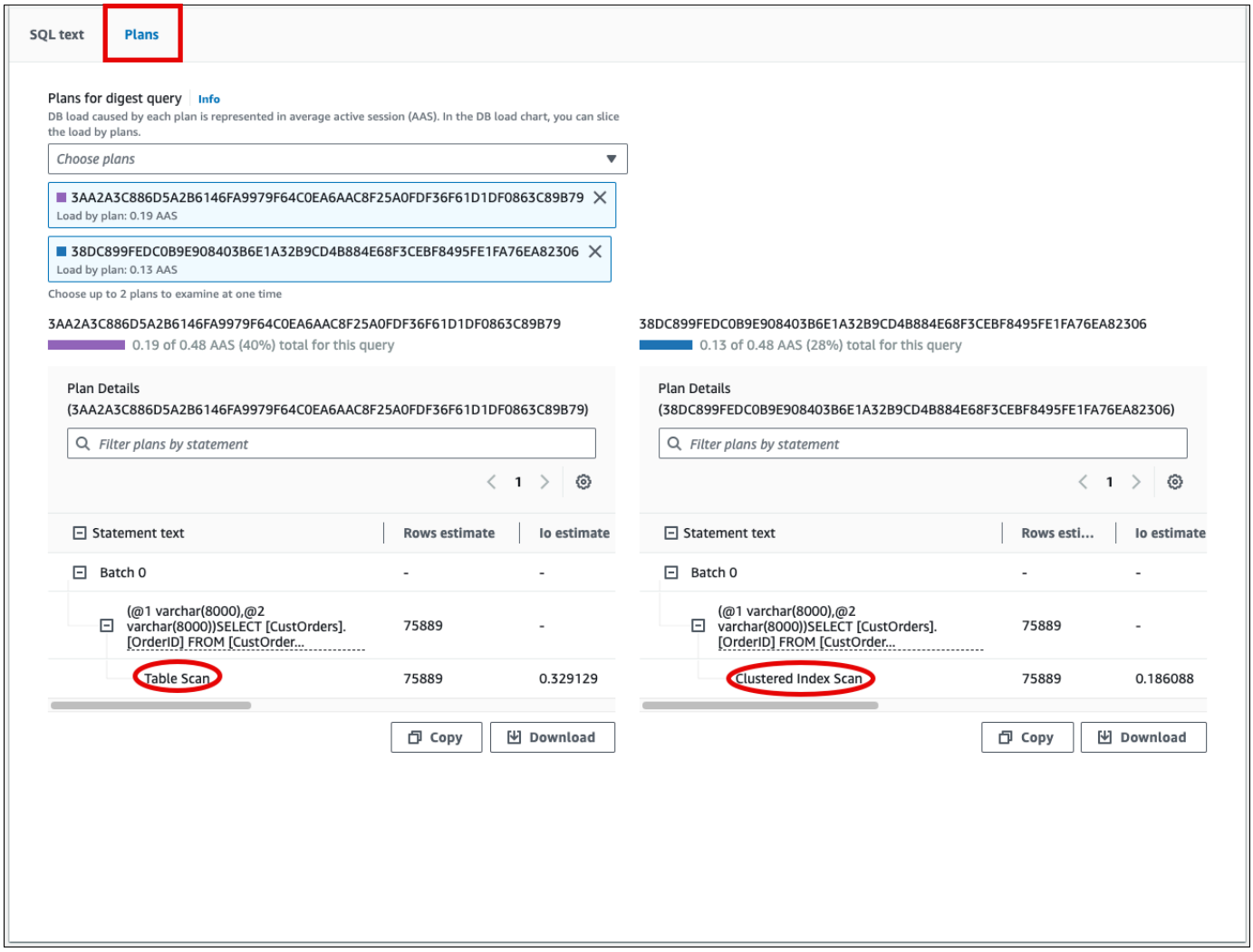
6. Scegli il digest per espanderlo nelle istruzioni componenti.

   Nell'esempio seguente, l'istruzione `SELECT` è una query digest. Le query relative ai componenti nel digest utilizzano tre diversi piani di esecuzione. I colori assegnati ai piani corrispondono al grafico di carico del database.



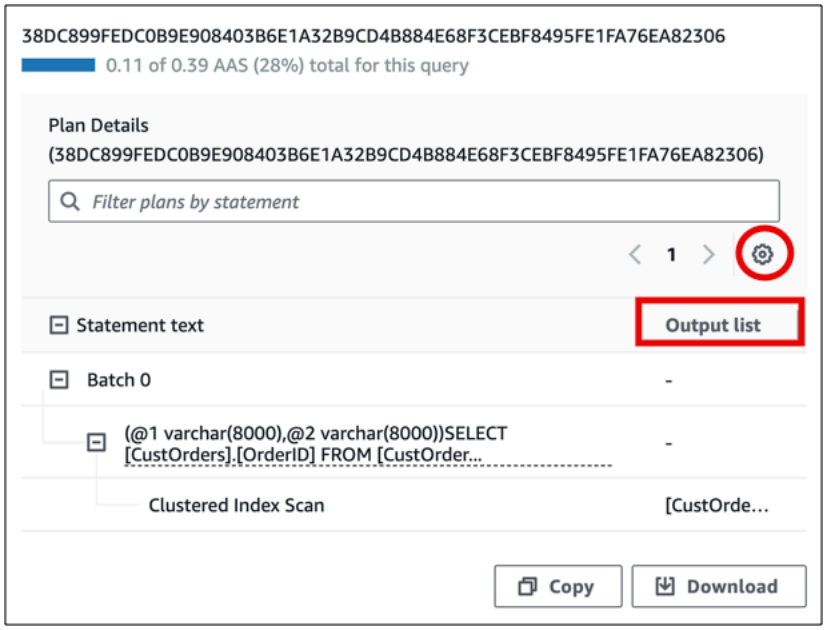
7. Scorri in basso e scegli due Plans (Piani) per il confronto dall'elenco Plans for digest query (Piani per la query digest).

   È possibile visualizzare uno o due piani per una query alla volta. La schermata seguente confronta due piani del riepilogo. Nell'esempio seguente, il 40% delle sessioni attive medie che eseguono questa query digest utilizza il piano a sinistra, mentre il 28% utilizza il piano a destra.



Nell'esempio precedente, i piani differiscono in modo importante. Il passaggio 2 del piano a sinistra utilizza una scansione della tabella, mentre il piano a destra utilizza una scansione dell'indice raggruppato. Per una tabella con un numero elevato di righe, una query che recupera una singola riga è quasi sempre più veloce con una scansione dell'indice in cluster.

8. (Facoltativo) Scegliete l'icona Impostazioni nella tabella Dettagli del piano per personalizzare la visibilità e l'ordine delle colonne. La schermata seguente mostra la tabella Dettagli del piano con la colonna Elenco output come seconda colonna.



9. (Facoltativo) Scegli Copy (Copia) per copiare il piano negli appunti, oppure Download (Scarica) per salvare il piano sul disco rigido.

> **Note**
>
> Performance Insights visualizza i piani di esecuzione stimati utilizzando una tabella ad albero gerarchica. La tabella include le informazioni di esecuzione parziali per ogni istruzione. Per ulteriori informazioni sulle colonne della tabella Plan Details, vedere SET SHOWPLAN_ALL nella documentazione di SQL Server. Per visualizzare le informazioni complete sull'esecuzione per un piano di esecuzione stimato, scegli Scarica per scaricare il piano, quindi carica il piano in SQL Server Management Studio. Per ulteriori informazioni sulla visualizzazione di un piano di esecuzione stimato utilizzando SQL Server Management Studio, vedere Visualizzazione di un piano di esecuzione stimato nella documentazione di SQL Server.

Visualizzazione dei consigli proattivi di Performance Insights

Amazon RDS Performance Insights monitora parametri specifici e crea automaticamente soglie analizzando quali livelli potrebbero essere potenzialmente problematici per una risorsa specifica. Quando i nuovi valori delle metriche superano una soglia predefinita in un determinato periodo di tempo, Performance Insights genera una raccomandazione proattiva. Questa raccomandazione aiuta

a prevenire futuri impatti sulle prestazioni del database. Per ricevere questi consigli proattivi, devi attivare Performance Insights con un periodo di conservazione a pagamento.

Per ulteriori informazioni sull'attivazione di Performance Insights, consultare Attivazione e disattivazione di Performance Insights. Per informazioni sui prezzi e sulla conservazione dei dati per Performance Insights, consultaPrezzi e conservazione dei dati per Performance Insights.

Per scoprire le regioni, i motori DB e le classi di istanze supportate per i consigli proattivi, consultaSupporto di classe di istanza, regione e motore di database Amazon RDS per funzionalità Performance Insights.

È possibile visualizzare l'analisi dettagliata e le indagini consigliate sui consigli proattivi nella pagina dei dettagli dei consigli.

Per ulteriori informazioni sui consigli, vedere. Visualizzazione e risposta ai consigli di RDS

Per visualizzare l'analisi dettagliata di una raccomandazione proattiva

1. Accedi alla AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, effettuate una delle seguenti operazioni:
   - Scegliete Consigli.

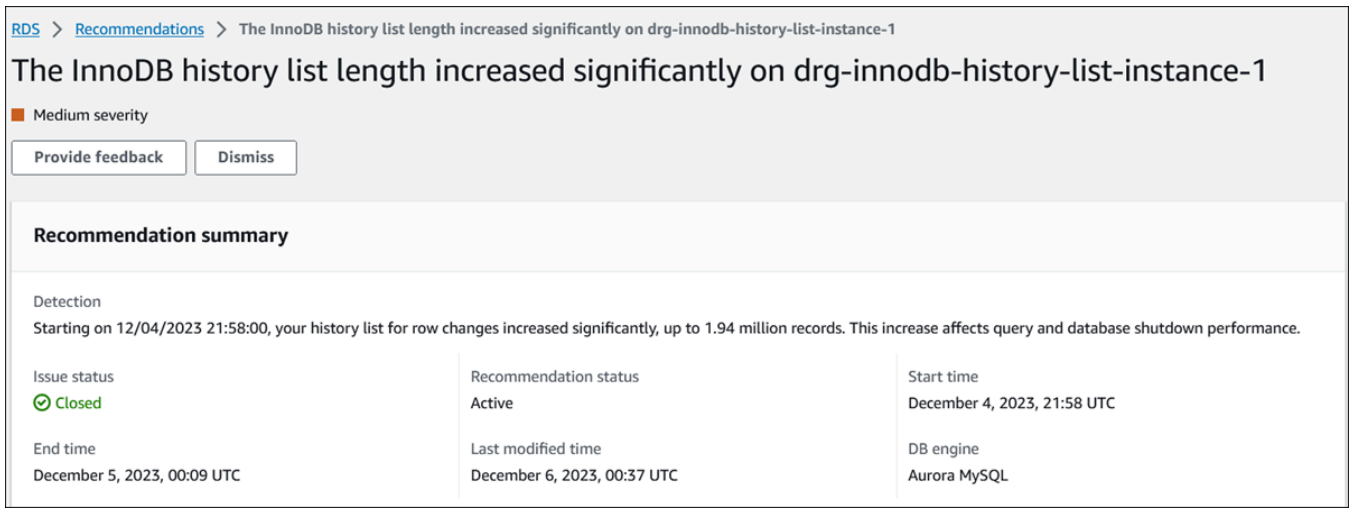
     La pagina Consigli mostra un elenco di consigli ordinati in base alla gravità per tutte le risorse del tuo account.
   - Scegli Database, quindi scegli Consigli per una risorsa nella pagina dei database.

     La scheda Consigli mostra i consigli e i relativi dettagli per la risorsa selezionata.
3. Trova un consiglio proattivo e scegli Visualizza dettagli.

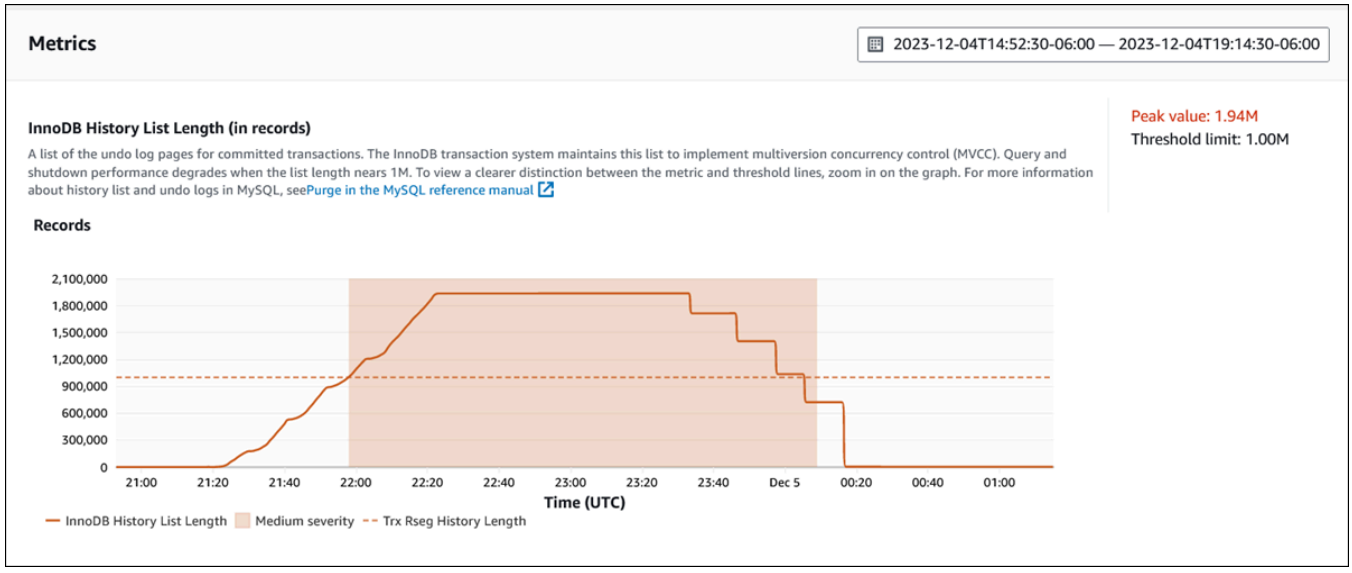
   Viene visualizzata la pagina dei dettagli del consiglio. Il titolo fornisce il nome della risorsa interessata con il problema rilevato e la gravità.

   Di seguito sono riportati i componenti della pagina dei dettagli dei consigli:

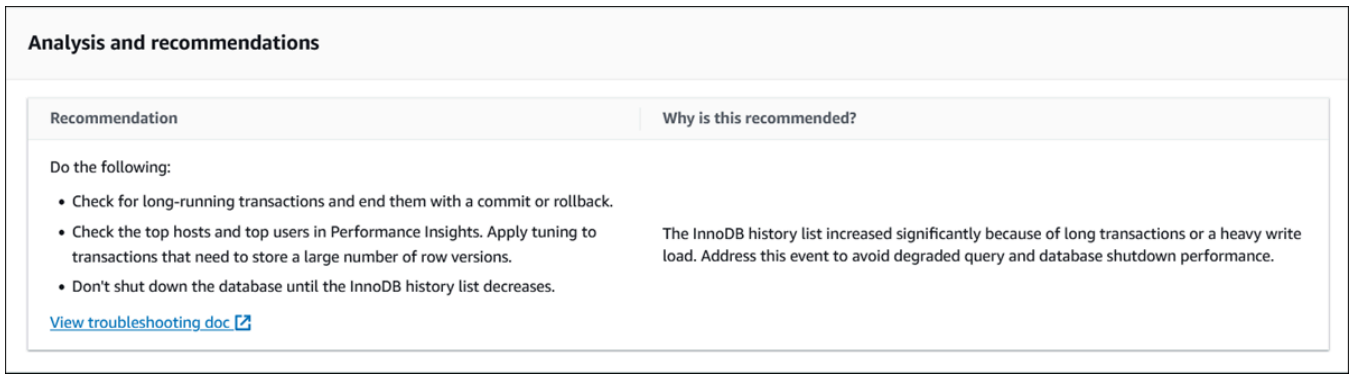
   - Riepilogo dei consigli: il problema rilevato, lo stato del suggerimento e del problema, l'ora di inizio e di fine del problema, l'ora di modifica del suggerimento e il tipo di motore.



- Metriche: i grafici del problema rilevato. Ogni grafico mostra una soglia determinata dal comportamento di base della risorsa e dai dati della metrica riportata dall'ora di inizio del problema.



- Analisi e raccomandazioni: la raccomandazione e il motivo della raccomandazione suggerita.



Puoi esaminare la causa del problema e quindi eseguire le azioni consigliate suggerite per risolvere il problema oppure scegliere Ignora in alto a destra per ignorare il consiglio.

Recupero dei parametri con l'API Performance Insights

Quando Performance Insights è attivato, l'API fornisce visibilità sulle prestazioni dell'istanza. Amazon CloudWatch Logs fornisce la fonte autorevole per i parametri di monitoraggio dei servizi forniti. AWS

Performance Insights offre una vista specifica del dominio del carico del database misurato come numero medio di sessioni attive (AAS). Questo parametro viene visualizzata dai consumer API come un set di dati temporali bidimensionali. La dimensione temporale dei dati fornisce i dati relativi al carico del database per ogni momento dell'intervallo di tempo in cui è stata eseguita la query. Ogni punto temporale scompone il carico complessivo in relazione alle dimensioni richieste, come `SQL`, `Wait-event`, `User` o `Host`, misurato in corrispondenza di quel punto temporale.

Amazon RDS Performance Insights monitora il cluster dell'istanza database Amazon RDS per consentire di analizzare e risolvere i problemi di performance del database. Un modo per visualizzare i dati di Performance Insights è disponibile nella AWS Management Console. Performance Insights fornisce inoltre un'API pubblica per eseguire query sui dati. Puoi usare l'API per effettuare quanto segue:

- Scaricamento dei dati in un database
- Aggiungi dati Performance Insights ai pannelli di controllo di monitoraggio esistenti
- Crea strumenti di monitoraggio

Per utilizzare l'API di Performance Insights, abilita Performance Insights su una delle istanze database Amazon RDS. Per informazioni sull'abilitazione di Performance Insights, consulta Attivazione e disattivazione di Performance Insights. Per ulteriori informazioni sull'API di Performance Insights, consulta la Documentazione di riferimento dell'API di Amazon RDS Performance Insights.

L'API di Performance Insights fornisce le seguenti operazioni.

Operazione di Performance Insights	AWS CLI command	Descrizione
CreatePerformanceAnalysisReport	aws pi create-performance-analysis-report	Crea un report di analisi delle prestazioni per un periodo di tempo specifico per l'istanza database. Il risultato è `AnalysisReportId` che è l'identificatore univoco del report.
DeletePerformanceAnalysisReport	aws pi delete-performance-analysis-report	Elimina un report di analisi delle prestazioni.
DescribeDimensionKeys	aws pi describe-dimension-keys	Recupera le prime N chiavi di dimensione per un parametro per un determinato periodo di tempo.
GetDimensionKeyDetails	aws pi get-dimension-key-details	Recupera gli attributi del gruppo di dimensioni specificato per un'istanza database o un'origine dati. Ad esempio, se si specifica un ID SQL e se i dettagli delle dimensioni sono disponibili, `GetDimensionKeyDetails` recupera il testo completo delle dimensioni `db.sql.statement` associate a questo ID. Questa operazione è utile perché `GetResourceMetrics` e `DescribeDimensionKeys` non supportano il recupero di testi

Operazione di Performance Insights	AWS CLI command	Descrizione
		di istruzioni SQL di grandi dimensioni.
GetPerformanceAnalysisReport	aws pi get-performance-analysis-report	Recupera il report, comprese le informazioni dettagliate relative al report. Il risultato include lo stato del report, l'ID del report, i dettagli sull'ora del report, le informazioni dettagliate e i suggerimenti.
GetResourceMetadata	aws pi get-resource-metadata	Recupera i metadati per diverse caratteristiche. Ad esempio, i metadati potrebbero indicare che una caratteristica è attivata o disattivata su un'istanza database specifica.
GetResourceMetrics	aws pi get-resource-metrics	Recupera parametri Performance Insights per un set di origini dati, su un periodo di tempo. Puoi fornire gruppi di dimensioni e dimensioni specifiche e fornire criteri di aggregazione e filtro per ogni gruppo.
ListAvailableResourceDimensions	aws pi list-available-resource-dimensions	Recupera le dimensioni su cui è possibile eseguire query per ogni tipo di parametro specificato su un'istanza specificata.

Operazione di Performance Insights	AWS CLI command	Descrizione
ListAvailableResourceMetrics	aws pi list-available-resource-metrics	Recupera tutti i parametri disponibili dei tipi di parametro specificati su cui è possibile eseguire query per un'istanza database specificata.
ListPerformanceAnalysisReports	aws pi list-performance-analysis-reports	Recupera tutti i report di analisi disponibili per l'istanza database. I report sono elencati in base all'ora di inizio di ciascun report.
ListTagsForResource	aws pi list-tags-for-resource	Elenca tutti i tag dei metadati aggiunti alla risorsa. L'elenco include il nome e il valore del tag.
TagResource	aws pi tag-resource	Aggiunge tag dei metadati alla risorsa Amazon RDS. Il tag include un nome e un valore.
UntagResource	aws pi untag-resource	Rimuove tag dei metadati dalla risorsa.

Argomenti

- AWS CLI per Performance Insights
- Recupero dei parametri di serie temporali
- AWS CLIEsempi di utilizzo di per Performance Insights

AWS CLI per Performance Insights

Puoi visualizzare i dati di Performance Insights utilizzando la AWS CLI. Puoi visualizzare la guida per i comandi AWS CLI per Performance Insights inserendo quanto segue nella riga di comando.

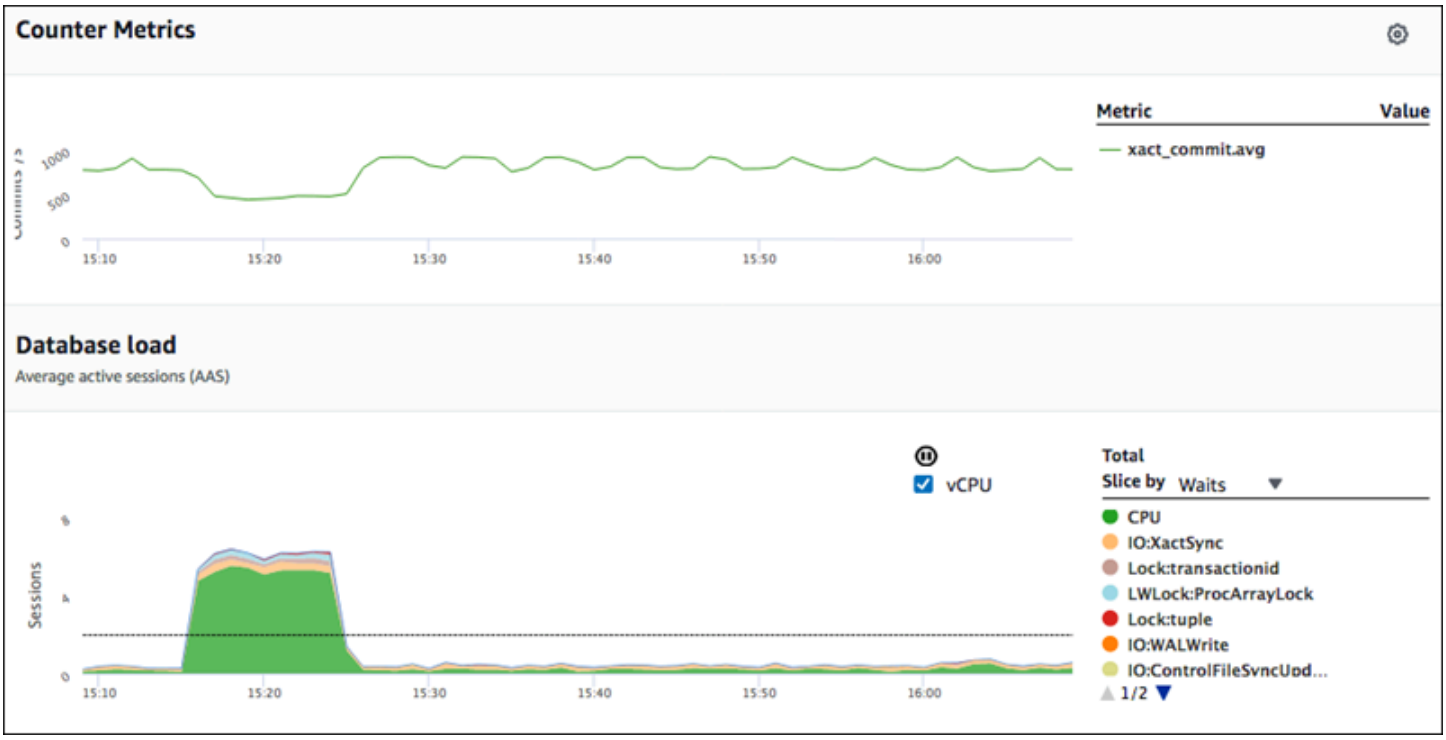
```
aws pi help
```

Se AWS CLI non è installato, consulta Installazione dell'interfaccia a riga di comando di AWS nella Guida per l'utente di AWS CLI per informazioni sull'installazione.

Recupero dei parametri di serie temporali

L'operazione `GetResourceMetrics` recupera uno o più parametri di serie temporali dai dati di Performance Insights. `GetResourceMetrics` richiede un parametro e un periodo di tempo e restituisce una risposta con un elenco di punti di dati.

Ad esempio, la AWS Management Console utilizza `GetResourceMetrics` per popolare il grafico Counter Metrics (Parametri contatore) e il grafico Database Load (Carico del database), come illustrato nell'immagine seguente.



Tutti i parametri restituiti da `GetResourceMetrics` sono parametri di serie temporali standard ad eccezione di `db.load`. Questo parametro è visualizzato nel grafico Database Load (Carico del database). Il parametro `db.load` è diverso dagli altri parametri di serie temporali in quanto può essere suddiviso in sottocomponenti detti dimensioni. Nell'immagine precedente, `db.load` è suddiviso e raggruppato in base agli stati delle attese che formano il `db.load`.

> **Note**
>
> `GetResourceMetrics` può anche restituire il parametro `db.sampleload`, ma il parametro `db.load` è appropriato nella maggior parte dei casi.

Per informazioni sui parametri contatore restituiti da `GetResourceMetrics`, consulta Parametri contatore di Performance Insights.

I seguenti calcoli sono supportati per i parametri:

- Media: il valore medio per il parametro su un periodo di tempo. Aggiungi `.avg` al nome parametro.
- Minimo: il valore minimo per il parametro su un periodo di tempo. Aggiungi `.min` al nome parametro.
- Massimo: il valore massimo per il parametro su un periodo di tempo. Aggiungi `.max` al nome parametro.
- Somma: la somma dei valori dei parametri su un periodo di tempo. Aggiungi `.sum` al nome parametro.
- Conteggio di esempio: il numero di volte che il parametro è stato raccolto su un periodo di tempo. Aggiungi `.sample_count` al nome parametro.

Ad esempio, supponiamo che un parametro venga raccolto per 300 secondi (5 minuti) e che il parametro venga raccolto una volta al minuto. I valori per ogni minuto sono 1, 2, 3, 4 e 5. In questo caso. vengono restituiti i seguenti calcoli:

- Media: 3
- Minimo: 1
- Massimo: 5
- Somma: 15
- Conteggio del campione: 5

Per ulteriori informazioni sull'utilizzo del comando `get-resource-metrics` della AWS CLI, consulta `get-resource-metrics`.

Per l'opzione `--metric-queries`, specifica una o più query per cui ottenere risultati. Ciascuna query consiste di un parametro obbligatorio `Metric` e parametri facoltativi `GroupBy` e `Filter`. Di seguito è riportato un esempio della specifica di un'opzione `--metric-queries`.

```
{
   "Metric": "string",
   "GroupBy": {
     "Group": "string",
     "Dimensions": ["string", ...],
     "Limit": integer
   },
   "Filter": {"string": "string"
     ...}
```

AWS CLIEsempi di utilizzo di per Performance Insights

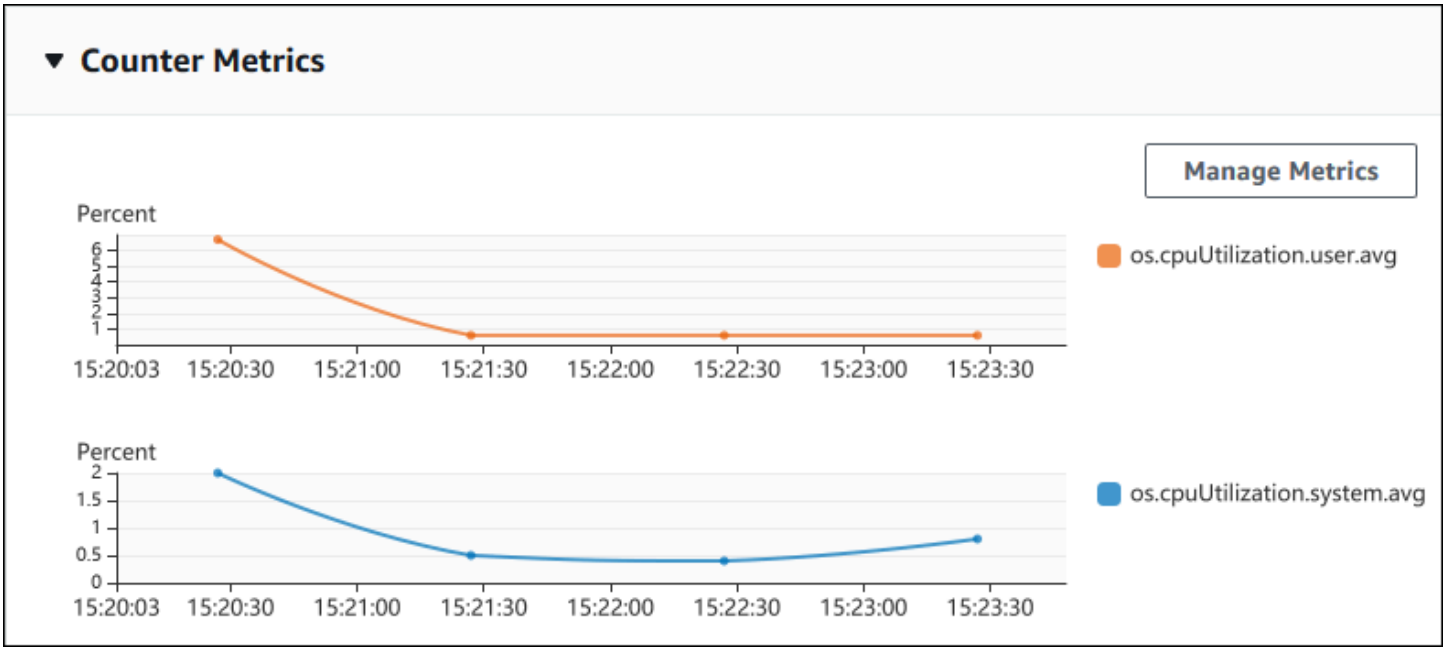
Negli esempi seguenti viene illustrato come utilizzare AWS CLI per Performance Insights.

Argomenti

- Recupero dei parametri contatore
- Recupero della media del carico del database per i principali eventi di attesa
- Recupero della media del carico del database per il principale SQL
- Recupero della media del carico del database filtrata da SQL
- Recupero del testo completo di un'istruzione SQL
- Creazione di un report di analisi delle prestazioni per un periodo di tempo
- Recupero di un report di analisi delle prestazioni
- Elenco di tutti i report di analisi delle prestazioni per l'istanza database
- Eliminazione di un report di analisi delle prestazioni
- Aggiunta di un tag a un report di analisi delle prestazioni
- Elenco di tutti i tag per un report di analisi delle prestazioni
- Eliminazione di tag da un report di analisi delle prestazioni

Recupero dei parametri contatore

Lo screenshot seguente mostra due grafici dei parametri contatore nella AWS Management Console.



L'esempio seguente mostra come raccogliere gli stessi dati che utilizza la AWS Management Console per generare i due grafici dei parametri contatore.

Per, o: Linux macOS Unix

```
aws pi get-resource-metrics \
   --service-type RDS \
   --identifier db-ID \
   --start-time 2018-10-30T00:00:00Z \
   --end-time   2018-10-30T01:00:00Z \
   --period-in-seconds 60 \
   --metric-queries '[{"Metric": "os.cpuUtilization.user.avg"  },
                      {"Metric": "os.cpuUtilization.idle.avg"}]'
```

Per Windows:

```
aws pi get-resource-metrics ^
   --service-type RDS ^
   --identifier db-ID ^
   --start-time 2018-10-30T00:00:00Z ^
   --end-time   2018-10-30T01:00:00Z ^
   --period-in-seconds 60 ^
   --metric-queries '[{"Metric": "os.cpuUtilization.user.avg"  },
                      {"Metric": "os.cpuUtilization.idle.avg"}]'
```

Puoi agevolare la lettura del comando specificando un file per l'opzione `--metrics-query`. Il seguente esempio utilizza un file denominato query.json per l'opzione. Il file presenta i seguenti contenuti.

```
[
    {
        "Metric": "os.cpuUtilization.user.avg"
    },
    {
        "Metric": "os.cpuUtilization.idle.avg"
    }
]
```

Esegui il comando seguente per utilizzare il file.

Per LinuxmacOS, oUnix:

```
aws pi get-resource-metrics \
   --service-type RDS \
   --identifier db-ID \
   --start-time 2018-10-30T00:00:00Z \
   --end-time   2018-10-30T01:00:00Z \
   --period-in-seconds 60 \
   --metric-queries file://query.json
```

Per Windows:

```
aws pi get-resource-metrics ^
   --service-type RDS ^
   --identifier db-ID ^
   --start-time 2018-10-30T00:00:00Z ^
   --end-time   2018-10-30T01:00:00Z ^
   --period-in-seconds 60 ^
   --metric-queries file://query.json
```

L'esempio precedente specifica i seguenti valori per le opzioni:

- `--service-type` – RDS for Amazon RDS
- `--identifier` – L'ID risorsa per l'istanza database
- `--start-time` e `--end-time` – I valori ISO 8601 `DateTime` per il periodo su cui eseguire le query, con supporto di più formati

Esegue query per un intervallo di tempo di un'ora:

- `--period-in-seconds` – `60` per una query al minuto
- `--metric-queries` – Una serie di due query, ognuna solo per un parametro

  Il nome del parametro utilizza punti per classificare il parametro in una categoria utile, dove l'ultimo elemento è una funzione. Nell'esempio, la funzione è `avg` per ciascuna query. Come per Amazon CloudWatch, le funzioni supportate sono `minmax`,`total`, `eavg`.

La risposta è simile a quella riportata di seguito.

```
{
    "Identifier": "db-XXX",
    "AlignedStartTime": 1540857600.0,
    "AlignedEndTime": 1540861200.0,
    "MetricList": [
        { //A list of key/datapoints
            "Key": {
                "Metric": "os.cpuUtilization.user.avg" //Metric1
            },
            "DataPoints": [
                //Each list of datapoints has the same timestamps and same number of
 items
                {
                    "Timestamp": 1540857660.0, //Minute1
                    "Value": 4.0
                },
                {
                    "Timestamp": 1540857720.0, //Minute2
                    "Value": 4.0
                },
                {
                    "Timestamp": 1540857780.0, //Minute 3
                    "Value": 10.0
                }
                //... 60 datapoints for the os.cpuUtilization.user.avg metric
            ]
        },
        {
            "Key": {
                "Metric": "os.cpuUtilization.idle.avg" //Metric2
            },
```

```
            "DataPoints": [
                {
                    "Timestamp": 1540857660.0, //Minute1
                    "Value": 12.0
                },
                {
                    "Timestamp": 1540857720.0, //Minute2
                    "Value": 13.5
                },
                //... 60 datapoints for the os.cpuUtilization.idle.avg metric
            ]
        }
    ] //end of MetricList
} //end of response
```

La risposta presenta un `Identifier`, un `AlignedStartTime` e un `AlignedEndTime`. Poiché il valore `--period-in-seconds` era `60`, l'ora di inizio e fine è stata allineata al minuto. Se `--period-in-seconds` fosse stato `3600`, l'ora di inizio e fine sarebbe stata allineata all'ora.

`MetricList` nella risposta ha una serie di voci, ciascuna con una voce `Key` e una voce `DataPoints`. Ciascun `DataPoint` ha un `Timestamp` e un `Value`. Ciascun elenco `Datapoints` ha 60 punti di dati in quanto le query sono per dati al minuto nell'arco di un'ora, con `Timestamp1/Minute1`, `Timestamp2/Minute2` e così via, fino a `Timestamp60/Minute60`.

Poiché la query è per due diversi parametri contatore, la risposta contiene due element `MetricList`.

Recupero della media del carico del database per i principali eventi di attesa

L'esempio seguente mostra la stessa query che utilizza la AWS Management Console per generare un grafico a linee ad area in pila. L'esempio recupera `db.load.avg` per l'ultima ora con carico diviso in base ai sette principali eventi di attesa. Il comando è come quello in Recupero dei parametri contatore. Tuttavia, il file query.json presenta i seguenti contenuti.

```
[
    {
        "Metric": "db.load.avg",
        "GroupBy": { "Group": "db.wait_event", "Limit": 7 }
    }
]
```

Eseguire il comando riportato qui di seguito.

Per LinuxmacOS, oUnix:

```
aws pi get-resource-metrics \
   --service-type RDS \
   --identifier db-ID \
   --start-time 2018-10-30T00:00:00Z \
   --end-time   2018-10-30T01:00:00Z \
   --period-in-seconds 60 \
   --metric-queries file://query.json
```

Per Windows:

```
aws pi get-resource-metrics ^
   --service-type RDS ^
   --identifier db-ID ^
   --start-time 2018-10-30T00:00:00Z ^
   --end-time   2018-10-30T01:00:00Z ^
   --period-in-seconds 60 ^
   --metric-queries file://query.json
```

L'esempio specifica il parametro di `db.load.avg` e un `GroupBy` dei sette principali eventi di attesa. Per i dettagli sui valori validi per questo esempio, consulta il riferimento DimensionGroupall'API Performance Insights.

La risposta è simile a quella riportata di seguito.

```
{
    "Identifier": "db-XXX",
    "AlignedStartTime": 1540857600.0,
    "AlignedEndTime": 1540861200.0,
    "MetricList": [
        { //A list of key/datapoints
            "Key": {
                //A Metric with no dimensions. This is the total db.load.avg
                "Metric": "db.load.avg"
            },
            "DataPoints": [
                //Each list of datapoints has the same timestamps and same number of
 items
                {
                    "Timestamp": 1540857660.0, //Minute1
                    "Value": 0.5166666666666667
```

```
                },
                {
                    "Timestamp": 1540857720.0, //Minute2
                    "Value": 0.38333333333333336
                },
                {
                    "Timestamp": 1540857780.0, //Minute 3
                    "Value": 0.26666666666666666
                }
                //... 60 datapoints for the total db.load.avg key
            ]
        },
        {
            "Key": {
                //Another key. This is db.load.avg broken down by CPU
                "Metric": "db.load.avg",
                "Dimensions": {
                    "db.wait_event.name": "CPU",
                    "db.wait_event.type": "CPU"
                }
            },
            "DataPoints": [
                {
                    "Timestamp": 1540857660.0, //Minute1
                    "Value": 0.35
                },
                {
                    "Timestamp": 1540857720.0, //Minute2
                    "Value": 0.15
                },
                //... 60 datapoints for the CPU key
            ]
        },
        //... In total we have 8 key/datapoints entries, 1) total, 2-8) Top Wait Events
    ] //end of MetricList
} //end of response
```

In questa risposta, ci sono otto voci in `MetricList`. C'è una voce per il `db.load.avg` totale e sette voci ciascuno per il `db.load.avg` suddivise secondo uno dei sette principali eventi di attesa. A differenza del primo esempio, poiché era presente una dimensione di raggruppamento, deve esserci una chiave per ciascun raggruppamento del parametro. Può esserci una sola chiave per ciascun parametro, come nel caso d'uso del parametro contatore di base.

Recupero della media del carico del database per il principale SQL

L'esempio seguente raggruppa `db.wait_events` in base alle 10 principali istruzioni SQL. Ci sono due diversi gruppi per le istruzioni SQL:

- `db.sql` – L'istruzione SQL completa, come `select * from customers where customer_id = 123`
- `db.sql_tokenized` – L'istruzione SQL in formato token, come `select * from customers where customer_id = ?`

Quando si analizzano le prestazioni del database, può essere utile considerare le istruzioni SQL che si differenziano solo per i loro parametri come un unico elemento logico. Pertanto, puoi utilizzare `db.sql_tokenized` durante le query. Tuttavia, soprattutto se ti interessano piani explain, a volte è più utile esaminare le istruzioni SQL complete con parametri e raggruppamento di query per `db.sql`. Vi è una relazione padre-figlio tra SQL in formato token e completo, con più SQL completi (figli) raggruppati nello stesso SQL in formato token (padre).

Il comando in questo esempio è simile a quello in Recupero della media del carico del database per i principali eventi di attesa. Tuttavia, il file query.json presenta i seguenti contenuti.

```
[
    {
        "Metric": "db.load.avg",
        "GroupBy": { "Group": "db.sql_tokenized", "Limit": 10 }
    }
]
```

Nell'esempio seguente viene utilizzato `db.sql_tokenized`.

Per LinuxmacOS, oUnix:

```
aws pi get-resource-metrics \
   --service-type RDS \
   --identifier db-ID \
   --start-time 2018-10-29T00:00:00Z \
   --end-time   2018-10-30T00:00:00Z \
   --period-in-seconds 3600 \
   --metric-queries file://query.json
```

Per Windows:

```
aws pi get-resource-metrics ^
   --service-type RDS ^
   --identifier db-ID ^
   --start-time 2018-10-29T00:00:00Z ^
   --end-time   2018-10-30T00:00:00Z  ^
   --period-in-seconds 3600 ^
   --metric-queries file://query.json
```

Questo esempio esegue una ricerca nell'arco di 24 ore, di cui un'ora period-in-seconds.

L'esempio specifica il parametro di `db.load.avg` e un `GroupBy` dei sette principali eventi di attesa. Per i dettagli sui valori validi per questo esempio, consulta il riferimento DimensionGroupall'API Performance Insights.

La risposta è simile a quella riportata di seguito.

```
{
    "AlignedStartTime": 1540771200.0,
    "AlignedEndTime": 1540857600.0,
    "Identifier": "db-XXX",

    "MetricList": [ //11 entries in the MetricList
        {
            "Key": { //First key is total
                "Metric": "db.load.avg"
            }
            "DataPoints": [ //Each DataPoints list has 24 per-hour Timestamps and a
 value
                {
                    "Value": 1.6964980544747081,
                    "Timestamp": 1540774800.0
                },
                //... 24 datapoints
            ]
        },
        {
            "Key": { //Next key is the top tokenized SQL
                "Dimensions": {
                    "db.sql_tokenized.statement": "INSERT INTO authors (id,name,email)
 VALUES\n( nextval(?)  ,?,?)",
                    "db.sql_tokenized.db_id": "pi-2372568224",
                    "db.sql_tokenized.id": "AKIAIOSFODNN7EXAMPLE"
                },
```

```
                "Metric": "db.load.avg"
            },
            "DataPoints": [ //... 24 datapoints
            ]
        },
        // In total 11 entries, 10 Keys of top tokenized SQL, 1 total key
    ] //End of MetricList
} //End of response
```

Questa risposta ha 11 voci in `MetricList` (1 SQL totale, 10 SQL principali in formato token), dove ciascuna ha 24 `DataPoints` ogni ora.

Per SQL in formato token, ci sono tre voci in ciascun elenco di dimensioni:

- `db.sql_tokenized.statement` – L'istruzione SQL in formato token.
- `db.sql_tokenized.db_id` – L'ID database nativo utilizzato per fare riferimento a SQL o un ID sintetico che Performance Insights genera nel caso in cui l'ID database nativo non sia disponibile. Questo esempio restituisce l'ID sintetico `pi-2372568224`.
- `db.sql_tokenized.id` – L'ID della query all'interno di Performance Insights.

  Nella AWS Management Console, questo ID è detto Support ID (ID supporto). Si chiama questo perché l'ID è dati che il AWS Support può esaminare per facilitare la risoluzione di un problema relativo al database. AWS prende molto seriamente la sicurezza e la privacy dei tuoi dati e quasi tutti i dati vengono archiviati crittografati con la tua chiave master cliente (CMK) AWS KMS. Pertanto, nessuno all'interno di AWS può accedere a tali dati. Nell'esempio precedente, sia `tokenized.statement` che `tokenized.db_id` vengono archiviati crittografati. Se riscontri un problema con il database, AWS Support può aiutarti facendo riferimento al Support ID (ID supporto).

Quando si eseguo query, potrebbe essere utile specificare un `Group` in `GroupBy`. Tuttavia, per un controllo più dettagliato dei dati restituiti, occorre specificare l'elenco delle dimensioni. Ad esempio, se tutto ciò di cui si necessita è `db.sql_tokenized.statement`, è possibile aggiungere l'attributo `Dimensions` al file query.json.

```
[
    {
        "Metric": "db.load.avg",
        "GroupBy": {
            "Group": "db.sql_tokenized",
```

```
            "Dimensions":["db.sql_tokenized.statement"],
            "Limit": 10
        }
    }
]
```

Recupero della media del carico del database filtrata da SQL

L'immagine precedente mostra che è stata selezionata una particolare query e che il grafico a linee ad area in pila con sessioni attive della media in alto è definito in base a tale query. Sebbene la query sia ancora per i sette principali eventi di attesa complessivi, il valore della risposta è filtrato. Il filtro fa sì che vengano prese in considerazione solo le sessioni che corrispondono al filtro specifico.

La query dell'API corrispondente in questo esempio è simile al comando in Recupero della media del carico del database per il principale SQL. Tuttavia, il file query.json presenta i seguenti contenuti.

```
[
 {
        "Metric": "db.load.avg",
        "GroupBy": { "Group": "db.wait_event", "Limit": 5  },
        "Filter": { "db.sql_tokenized.id": "AKIAIOSFODNN7EXAMPLE" }
    }
```

```
]
```

Per LinuxmacOS, oUnix:

```
aws pi get-resource-metrics \
   --service-type RDS \
   --identifier db-ID \
   --start-time 2018-10-30T00:00:00Z \
   --end-time   2018-10-30T01:00:00Z \
   --period-in-seconds 60 \
   --metric-queries file://query.json
```

Per Windows:

```
aws pi get-resource-metrics ^
   --service-type RDS ^
   --identifier db-ID ^
   --start-time 2018-10-30T00:00:00Z ^
   --end-time   2018-10-30T01:00:00Z ^
   --period-in-seconds 60 ^
   --metric-queries file://query.json
```

La risposta è simile a quella riportata di seguito.

```
{
    "Identifier": "db-XXX",
    "AlignedStartTime": 1556215200.0,
    "MetricList": [
        {
            "Key": {
                "Metric": "db.load.avg"
            },
            "DataPoints": [
                {
                    "Timestamp": 1556218800.0,
                    "Value": 1.4878117913832196
                },
                {
                    "Timestamp": 1556222400.0,
                    "Value": 1.192823803967328
                }
            ]
```

```
        },
        {
            "Key": {
                "Metric": "db.load.avg",
                "Dimensions": {
                    "db.wait_event.type": "io",
                    "db.wait_event.name": "wait/io/aurora_redo_log_flush"
                }
            },
            "DataPoints": [
                {
                    "Timestamp": 1556218800.0,
                    "Value": 1.1360544217687074
                },
                {
                    "Timestamp": 1556222400.0,
                    "Value": 1.058051341890315
                }
            ]
        },
        {
            "Key": {
                "Metric": "db.load.avg",
                "Dimensions": {
                    "db.wait_event.type": "io",
                    "db.wait_event.name": "wait/io/table/sql/handler"
                }
            },
            "DataPoints": [
                {
                    "Timestamp": 1556218800.0,
                    "Value": 0.16241496598639457
                },
                {
                    "Timestamp": 1556222400.0,
                    "Value": 0.05163360560093349
                }
            ]
        },
        {
            "Key": {
                "Metric": "db.load.avg",
                "Dimensions": {
                    "db.wait_event.type": "synch",
```

```
                    "db.wait_event.name": "wait/synch/mutex/innodb/
aurora_lock_thread_slot_futex"
                }
            },
            "DataPoints": [
                {
                    "Timestamp": 1556218800.0,
                    "Value": 0.11479591836734694
                },
                {
                    "Timestamp": 1556222400.0,
                    "Value": 0.013127187864644107
                }
            ]
        },
        {
            "Key": {
                "Metric": "db.load.avg",
                "Dimensions": {
                    "db.wait_event.type": "CPU",
                    "db.wait_event.name": "CPU"
                }
            },
            "DataPoints": [
                {
                    "Timestamp": 1556218800.0,
                    "Value": 0.05215419501133787
                },
                {
                    "Timestamp": 1556222400.0,
                    "Value": 0.05805134189031505
                }
            ]
        },
        {
            "Key": {
                "Metric": "db.load.avg",
                "Dimensions": {
                    "db.wait_event.type": "synch",
                    "db.wait_event.name": "wait/synch/mutex/innodb/lock_wait_mutex"
                }
            },
            "DataPoints": [
                {
```

```
                    "Timestamp": 1556218800.0,
                    "Value": 0.017573696145124718
                },
                {
                    "Timestamp": 1556222400.0,
                    "Value": 0.002333722287047841
                }
            ]
        }
    ],
    "AlignedEndTime": 1556222400.0
} //end of response
```

In questa risposta, tutti i valori sono filtrati in base al contributo di SQL in formato token AKIAIOSFODNN7EXAMPLE specificato nel file query.json. Le chiavi potrebbero inoltre seguire un ordine diverso rispetto a una query senza filtro, in quanto sono i cinque principali eventi di attesa che influenzano l'SQL filtrato.

Recupero del testo completo di un'istruzione SQL

L'esempio seguente recupera il testo completo di un'istruzione SQL per l'istanza database `db-10BCD2EFGHIJ3KL4M5NO6PQRS5`. `--group` è `db.sql`, e `--group-identifier` è `db.sql.id`. In questo esempio, *`my-sql-id`*rappresenta un ID SQL recuperato richiamando `pi get-resource-metrics` o. `pi describe-dimension-keys`

Esegui il comando seguente.

PerLinux, omacOS: Unix

```
aws pi get-dimension-key-details \
   --service-type RDS \
   --identifier db-10BCD2EFGHIJ3KL4M5NO6PQRS5 \
   --group db.sql \
   --group-identifier my-sql-id \
   --requested-dimensions statement
```

Per Windows:

```
aws pi get-dimension-key-details ^
   --service-type RDS ^
```

```
   --identifier db-10BCD2EFGHIJ3KL4M5NO6PQRS5 ^
   --group db.sql ^
   --group-identifier my-sql-id ^
   --requested-dimensions statement
```

In questo esempio, sono disponibili i dettagli delle dimensioni. Pertanto, Performance Insights recupera il testo completo dell'istruzione SQL, senza troncarla.

```
{
    "Dimensions":[
    {
        "Value": "SELECT e.last_name, d.department_name FROM employees e, departments d
 WHERE e.department_id=d.department_id",
        "Dimension": "db.sql.statement",
        "Status": "AVAILABLE"
    },
    ...
    ]
}
```

Creazione di un report di analisi delle prestazioni per un periodo di tempo

L'esempio seguente crea un report di analisi delle prestazioni con l'ora di inizio `1682969503` e l'ora di fine `1682979503` per il database `db-loadtest-0`.

```
      aws pi-test create-performance-analysis-report \
      --service-type RDS \
      --identifier db-loadtest-0 \
      --start-time 1682969503 \
      --end-time 1682979503 \
      --endpoint-url https://api.titan.pi.a2z.com \
      --region us-west-2
```

La risposta è l'identificatore univoco `report-0234d3ed98e28fb17` per il report.

```
     {
        "AnalysisReportId": "report-0234d3ed98e28fb17"
     }
```

Recupero di un report di analisi delle prestazioni

L'esempio seguente recupera i dettagli del report di analisi per il report `report-0d99cc91c4422ee61`.

```
        aws pi-test get-performance-analysis-report \
--service-type RDS \
--identifier db-loadtest-0 \
--analysis-report-id report-0d99cc91c4422ee61 \
--endpoint-url https://api.titan.pi.a2z.com \
--region us-west-2
```

La risposta fornisce lo stato del report, l'ID, i dettagli temporali e le informazioni dettagliate.

```
        {
    "AnalysisReport": {
        "Status": "Succeeded",
        "ServiceType": "RDS",
        "Identifier": "db-loadtest-0",
        "StartTime": 1680583486.584,
        "AnalysisReportId": "report-0d99cc91c4422ee61",
        "EndTime": 1680587086.584,
        "CreateTime": 1680587087.139,
        "Insights": [
           ... (Condensed for space)
        ]
    }
}
```

Elenco di tutti i report di analisi delle prestazioni per l'istanza database

L'esempio seguente elenca tutti i report di analisi delle prestazioni disponibili per il database `db-loadtest-0`.

```
            aws pi-test list-performance-analysis-reports \
--service-type RDS \
--identifier db-loadtest-0 \
--endpoint-url https://api.titan.pi.a2z.com \
--region us-west-2
```

La risposta elenca tutti i report con i dettagli relativi all'ID, allo stato e al periodo di tempo del report.

```
            {
    "AnalysisReports": [
        {
            "Status": "Succeeded",
            "EndTime": 1680587086.584,
            "CreationTime": 1680587087.139,
            "StartTime": 1680583486.584,
            "AnalysisReportId": "report-0d99cc91c4422ee61"
        },
        {
            "Status": "Succeeded",
            "EndTime": 1681491137.914,
            "CreationTime": 1681491145.973,
            "StartTime": 1681487537.914,
            "AnalysisReportId": "report-002633115cc002233"
        },
        {
            "Status": "Succeeded",
            "EndTime": 1681493499.849,
            "CreationTime": 1681493507.762,
            "StartTime": 1681489899.849,
            "AnalysisReportId": "report-043b1e006b47246f9"
        },
        {
            "Status": "InProgress",
            "EndTime": 1682979503.0,
            "CreationTime": 1682979618.994,
            "StartTime": 1682969503.0,
            "AnalysisReportId": "report-01ad15f9b88bcbd56"
        }
    ]
}
```

Eliminazione di un report di analisi delle prestazioni

L'esempio seguente elimina il report di analisi per il database `db-loadtest-0`.

```
       aws pi-test delete-performance-analysis-report \
--service-type RDS \
--identifier db-loadtest-0 \
--analysis-report-id report-0d99cc91c4422ee61 \
--endpoint-url https://api.titan.pi.a2z.com \
--region us-west-2
```

Aggiunta di un tag a un report di analisi delle prestazioni

L'esempio seguente aggiunge un tag con una chiave `name` e un valore `test-tag` al report `report-01ad15f9b88bcbd56`.

```
           aws pi-test tag-resource \
--service-type RDS \
--resource-arn arn:aws:pi:us-west-2:356798100956:perf-reports/RDS/db-loadtest-0/
report-01ad15f9b88bcbd56 \
--tags Key=name,Value=test-tag \
--endpoint-url https://api.titan.pi.a2z.com \
--region us-west-2
```

Elenco di tutti i tag per un report di analisi delle prestazioni

Nell'esempio seguente vengono elencati tutti i tag per il report `report-01ad15f9b88bcbd56`.

```
       aws pi-test list-tags-for-resource \
--service-type RDS \
--resource-arn arn:aws:pi:us-west-2:356798100956:perf-reports/RDS/db-loadtest-0/
report-01ad15f9b88bcbd56 \
--endpoint-url https://api.titan.pi.a2z.com \
--region us-west-2
```

La risposta elenca il valore e la chiave per tutti i tag aggiunti al report:

```
        {
    "Tags": [
        {
            "Value": "test-tag",
            "Key": "name"
        }
    ]
}
```

Eliminazione di tag da un report di analisi delle prestazioni

Nell'esempio seguente viene eliminato il tag `name` dal report `report-01ad15f9b88bcbd56`.

```
        aws pi-test untag-resource \
--service-type RDS \
--resource-arn arn:aws:pi:us-west-2:356798100956:perf-reports/RDS/db-loadtest-0/
report-01ad15f9b88bcbd56 \
--tag-keys name \
--endpoint-url https://api.titan.pi.a2z.com \
--region us-west-2
```

Dopo che il tag è stato eliminato, se si chiama l'API `list-tags-for-resource` questo tag non viene elencato.

Registrazione delle chiamate Performance Insights utilizzando AWS CloudTrail

Performance Insights viene eseguito con AWS CloudTrail, un servizio che fornisce un record delle azioni intraprese da un utente, un ruolo o un servizio AWS in Performance Insights. CloudTrail acquisisce tutte le chiamate API per Performance Insights come eventi. Questa acquisizione include chiamate dalla console Amazon RDS e dalle chiamate di codice alle operazioni API di Performance Insights.

Se viene creato un trail, è possibile abilitare la distribuzione continua di eventi CloudTrail in un bucket Simple Storage Service (Amazon S3), inclusi gli eventi per Performance Insights. Se non configuri un trail, è comunque possibile visualizzare gli eventi più recenti nella console di CloudTrail

in Event history (Cronologia eventi). Utilizzando le informazioni raccolte da CloudTrail è possibile determinare specifici dettagli. Queste informazioni includono la richiesta effettuata a Performance Insights, l'indirizzo IP da cui è stata eseguita la richiesta, l'autore della richiesta e il momento in cui è stata eseguita. Include anche dettagli aggiuntivi.

Per ulteriori informazioni su CloudTrail, consultare la AWS CloudTrailGuida per l'utente di .

Utilizzo delle informazioni di Performance Insights in CloudTrail

CloudTrail è abilitato sull'account AWS al momento della sua creazione. Quando si verifica un'attività in Performance Insights, tale attività viene registrata in un evento CloudTrail insieme ad altri eventi del servizio AWS nella console CloudTrail in Cronologia eventi. È possibile visualizzare, cercare e scaricare gli eventi recenti nell'account AWS. Per ulteriori informazioni, consulta Visualizzazione di eventi nella cronologia degli eventi di CloudTrail nella Guida per l'utente di AWS CloudTrail.

Per una registrazione di eventi nell'account AWS che includa eventi per Performance Insights, crea un trail. Un trail consente a CloudTrail di distribuire i file di log in un bucket Simple Storage Service (Amazon S3). Per impostazione predefinita, quando si crea un trail nella console, il trail sarà valido in tutte le regioni AWS. Il percorso registra gli eventi da tutte le regioni AWSnella partizione AWS e distribuisce i file di log nel bucket Simple Storage Service (Amazon S3) specificato. Inoltre, è possibile configurare altri servizi AWS per analizzare con maggiore dettaglio e usare i dati evento raccolti nei registri CloudTrail. Per ulteriori informazioni, consulta gli argomenti seguenti nella Guida per l'utente di AWS CloudTrail:

- Panoramica della creazione di un percorso
- Servizi e integrazioni CloudTrail supportati
- Configurazione delle notifiche Amazon SNS per CloudTrail
- Ricezione di file di log CloudTrail da più regioni e Ricezione di file di log CloudTrail da più account

Tutte le operazioni di Performance Insights vengono registrate da CloudTrail e documentate nella Documentazione di riferimento dell'API di Performance Insights. Ad esempio, tutte le chiamate alle operazioni `DescribeDimensionKeys` e `GetResourceMetrics` generano voci nei file di log di CloudTrail.

Ogni evento o voce di log contiene informazioni sull'utente che ha generato la richiesta. Le informazioni di identità consentono di determinare quanto segue:

- Se la richiesta è stata effettuata con le credenziali utente IAM o root.

- Se la richiesta è stata effettuata con le credenziali di sicurezza temporanee per un ruolo o un utente federato.
- Se la richiesta è stata effettuata da un altro servizio AWS.

Per ulteriori informazioni, consultare Elemento userIdentity di CloudTrail.

Voci del file di registro Performance Insights

Un percorso è una configurazione che consente la distribuzione di eventi come i file di log in un bucket Simple Storage Service (Amazon S3) specificato. I file di log di CloudTrail possono contenere una o più voci di log. Un evento rappresenta una singola richiesta da un'origine. Ogni evento include informazioni sull'operazione richiesta, la data e l'ora dell'azione, i parametri della richiesta e così via. I file di log CloudTrail non sono una traccia di stack ordinata delle chiamate API pubbliche e di conseguenza non devono apparire in base a un ordine specifico.

L'esempio seguente mostra una voce di log di CloudTrail che illustra l'operazione `GetResourceMetrics`:


```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "IAMUser",
         "principalId": "AKIAIOSFODNN7EXAMPLE",
        "arn": "arn:aws:iam::123456789012:user/johndoe",
        "accountId": "123456789012",
        "accessKeyId": "AKIAI44QH8DHBEXAMPLE",
        "userName": "johndoe"
    },
    "eventTime": "2019-12-18T19:28:46Z",
    "eventSource": "pi.amazonaws.com",
    "eventName": "GetResourceMetrics",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "72.21.198.67",
    "userAgent": "aws-cli/1.16.240 Python/3.7.4 Darwin/18.7.0 botocore/1.12.230",
    "requestParameters": {
        "identifier": "db-YTDU5J5V66X7CXSCVDFD2V3SZM",
        "metricQueries": [
            {
                "metric": "os.cpuUtilization.user.avg"
            },
            {
```

```
                "metric": "os.cpuUtilization.idle.avg"
            }
        ],
        "startTime": "Dec 18, 2019 5:28:46 PM",
        "periodInSeconds": 60,
        "endTime": "Dec 18, 2019 7:28:46 PM",
        "serviceType": "RDS"
    },
    "responseElements": null,
    "requestID": "9ffbe15c-96b5-4fe6-bed9-9fccff1a0525",
    "eventID": "08908de0-2431-4e2e-ba7b-f5424f908433",
    "eventType": "AwsApiCall",
    "recipientAccountId": "123456789012"
}
```

Analisi delle anomalie delle prestazioni con Amazon DevOps Guru per Amazon RDS

Amazon DevOps Guru è un servizio operativo completamente gestito che aiuta sviluppatori e operatori a migliorare le prestazioni e la disponibilità delle loro applicazioni. DevOpsGuru delega le attività associate all'identificazione dei problemi operativi in modo da poter implementare rapidamente i consigli per migliorare la tua applicazione. Per ulteriori informazioni, consulta Cos'è Amazon DevOps Guru? nella Guida per l'utente di Amazon DevOps Guru.

DevOpsGuru rileva, analizza e fornisce raccomandazioni per i problemi operativi esistenti per tutti i motori Amazon RDS DB. DevOpsGuru for RDS estende questa funzionalità applicando l'apprendimento automatico ai parametri di Performance Insights per i database RDS per PostgreSQL. Queste funzionalità di monitoraggio consentono a DevOps Guru for RDS di rilevare e diagnosticare i rallentamenti delle prestazioni e consigliare azioni correttive specifiche. DevOpsGuru for RDS può anche rilevare condizioni problematiche nei for PostgreSQL) prima che si verifichino.

È ora possibile visualizzare questi consigli nella console RDS. Per ulteriori informazioni, consulta Visualizzazione e risposta ai consigli di RDS.

Il video seguente è una panoramica di DevOps Guru for RDS.

Per un'analisi approfondita di questo argomento, consulta Amazon DevOps Guru for RDS under the hood.

Argomenti

- Vantaggi di DevOps Guru for RDS
- Come funziona DevOps Guru for RDS
- Configurazione di Guru per RDS DevOps

Vantaggi di DevOps Guru for RDS

Se sei responsabile di un database RDS per PostgreSQL, potresti non sapere che si sta verificando un evento o una regressione che interessa il database. Quando scopri il problema, potresti non sapere perché si sta verificando o cosa fare al riguardo. Invece di rivolgerti a un amministratore di database (DBA) per ricevere assistenza o affidarti a strumenti di terze parti, puoi seguire i consigli di Guru for RDS. DevOps

L'analisi dettagliata di Guru for RDS consente di DevOps ottenere i seguenti vantaggi:

Diagnosi rapida

DevOpsGuru for RDS monitora e analizza continuamente la telemetria del database. DevOpsGuru for RDS utilizza tecniche statistiche e di apprendimento automatico per estrarre questi dati e rilevare anomalie. Per ulteriori informazioni sui dati di telemetria, consulta Monitoraggio del carico del DB con Approfondimenti sulle prestazioni su Amazon RDS e Monitoraggio delle metriche del sistema operativo con monitoraggio avanzato nella Guida per l'utente di Amazon RDS.

Risoluzione rapida

Ogni anomalia identifica il problema delle prestazioni e suggerisce strade di indagine o azioni correttive. Ad esempio, DevOps Guru for RDS potrebbe consigliare di esaminare specifici eventi di attesa. In alternativa, è consigliabile regolare le impostazioni del pool di applicazioni per limitare il numero di connessioni al database. Sulla base di questi consigli, è possibile risolvere i problemi di prestazioni più rapidamente rispetto alla risoluzione manuale dei problemi.

Approfondimenti proattivi

DevOpsGuru for RDS utilizza le metriche delle tue risorse per rilevare comportamenti potenzialmente problematici prima che diventino un problema più grave. Ad esempio, è in grado di rilevare quando il database utilizza un numero crescente di tabelle temporanee su disco, ovvero una situazione che potrebbe pregiudicare le prestazioni. DevOpsGuru fornisce quindi consigli per aiutarvi a risolvere i problemi prima che diventino problemi più gravi.

Conoscenza approfondita dei tecnici e del machine learning di Amazon

Per rilevare problemi di prestazioni e aiutarti a risolvere i problemi, DevOps Guru for RDS si affida all'apprendimento automatico (ML) e a formule matematiche avanzate. Gli ingegneri di database di Amazon hanno contribuito allo sviluppo dei risultati di DevOps Guru for RDS, che racchiudono molti anni di gestione di centinaia di migliaia di database. Attingendo a questa conoscenza collettiva, DevOps Guru for RDS può insegnarti le migliori pratiche.

Come funziona DevOps Guru for RDS

DevOpsGuru for RDS raccoglie dati sui database RDS per PostgreSQL da Amazon RDS Performance Insights. La `DBLoad` metrica più importante è. DevOpsGuru for RDS utilizza le metriche di Performance Insights, le analizza con l'apprendimento automatico e pubblica le informazioni sulla dashboard.

Un'analisi è una raccolta di anomalie correlate rilevate da Guru. DevOps

In DevOps Guru for RDS, un'anomalia è un pattern che si discosta da quelle che vengono considerate prestazioni normali per il tuo database Amazon RDS per PostgreSQL.

Approfondimenti proattivi

Un approfondimento proattivo consente di individuare i comportamenti problematici prima che si verifichino. Contiene le anomalie accompagnate da suggerimenti e metriche correlati per aiutarti a risolvere le condizioni problematiche nei tuoi database RDS per PostgreSQL prima che diventino problemi più seri. Questi approfondimenti sono pubblicati nella dashboard Guru. DevOps

Ad esempio, DevOps Guru potrebbe rilevare che il database RDS per PostgreSQL sta creando molte tabelle temporanee su disco. Se non affrontata per tempo, questa tendenza può causare problemi di prestazioni. Ogni approfondimento proattivo include i suggerimenti per i comportamenti correttivi e i collegamenti ad argomenti pertinenti in Ottimizzazione di RDS per PostgreSQL con approfondimenti proattivi di Amazon DevOps Guru. Per ulteriori informazioni, consulta Working with Insights in DevOps Guru nella Amazon DevOps Guru User Guide.

Approfondimenti reattivi

Un approfondimento reattivo identifica un comportamento anomalo nel momento in cui si verifica. Se DevOps Guru for RDS rileva problemi di prestazioni nelle tue istanze DB RDS per PostgreSQL, pubblica una panoramica reattiva nella dashboard Guru. DevOps Per ulteriori informazioni, consulta Working with Insights in DevOps Guru nella Amazon DevOps Guru User Guide.

Anomalie causali

Un'anomalia causale è un'anomalia di livello superiore all'interno di un approfondimento reattivo. Il caricamento del database (caricamento del DB) è l'anomalia causale di Guru for DevOps RDS.

Un'anomalia misura l'impatto sulle prestazioni assegnando un livello di gravità di Elevato,Medio, oppureBasso. Per ulteriori informazioni, consulta Concetti chiave per DevOps Guru for RDS nella Amazon DevOps Guru User Guide.

Se DevOps Guru rileva un'anomalia corrente sulla tua istanza DB, verrai avvisato nella pagina Databases della console RDS. La console ti avvisa anche delle anomalie che si sono verificate nelle ultime 24 ore. Per andare alla pagina delle anomalie dalla console RDS, scegliere il link nel messaggio di avviso. La console RDS ti avvisa anche nella pagina dell'istanza database RDS per PostgreSQL.

Anomalie contestuali

Un'anomalia contestuale è un risultato del carico del database correlato a un approfondimento reattivo. Ogni anomalia contestuale descrive uno specifico problema di prestazioni di RDS per PostgreSQL che richiede un'indagine. Ad esempio, DevOps Guru for RDS potrebbe consigliare di prendere in considerazione l'aumento della capacità della CPU o di esaminare gli eventi di attesa che contribuiscono al carico del DB.

> **Important**
>
> È consigliabile testare eventuali modifiche in un'istanza di test prima di modificare un'istanza di produzione. In questo modo, capisci l'impatto del cambiamento.

Per ulteriori informazioni, consulta la sezione Analyzing anomalies in Amazon RDS nella Amazon Guru User Guide. DevOps

Configurazione di Guru per RDS DevOps

Per consentire a DevOps Guru for Amazon RDS di pubblicare approfondimenti per e RDS per PostgreSQL, completa le seguenti attività.

Argomenti

- Configurazione delle politiche di accesso IAM per Guru for RDS DevOps
- Attivazione di Approfondimenti sulle prestazioni per le istanze database RDS per PostgreSQL
- Attivare DevOps Guru e specificare la copertura delle risorse

Configurazione delle politiche di accesso IAM per Guru for RDS DevOps

Per visualizzare gli avvisi di DevOps Guru nella console RDS, il tuo utente o ruolo AWS Identity and Access Management (IAM) deve disporre di una delle seguenti politiche:

- La politica gestita AWS `AmazonDevOpsGuruConsoleFullAccess`
- La politica AWS gestita `AmazonDevOpsGuruConsoleReadOnlyAccess` e una delle seguenti politiche:
  - La politica AWS gestita `AmazonRDSFullAccess`
  - Una policy gestita dal cliente che include `pi:GetResourceMetrics` e `pi:DescribeDimensionKeys`

Per ulteriori informazioni, consulta Configurazione delle policy di accesso per Performance Insights.

Attivazione di Approfondimenti sulle prestazioni per le istanze database RDS per PostgreSQL

DevOpsGuru for RDS si affida a Performance Insights per i suoi dati. Senza Performance Insights, DevOps Guru pubblica le anomalie, ma non include analisi e raccomandazioni dettagliate.

Quando crei o modifichi un'istanza database RDS per PostgreSQL, puoi attivare Approfondimenti sulle prestazioni. Per ulteriori informazioni, consulta Attivazione e disattivazione di Performance Insights.

Attivare DevOps Guru e specificare la copertura delle risorse

Puoi attivare DevOps Guru per fargli monitorare i tuoi database Aurora RDS per PostgreSQL in uno dei seguenti modi.

Argomenti

- Attivazione di Guru nella console RDS DevOps
- Aggiungere risorse per PostgreSQL nella console Guru DevOps
- Aggiungere risorse per PostgreSQL utilizzandoAWS CloudFormation

Attivazione di Guru nella console RDS DevOps

Puoi seguire più percorsi nella console Amazon RDS per attivare DevOps Guru.

Argomenti

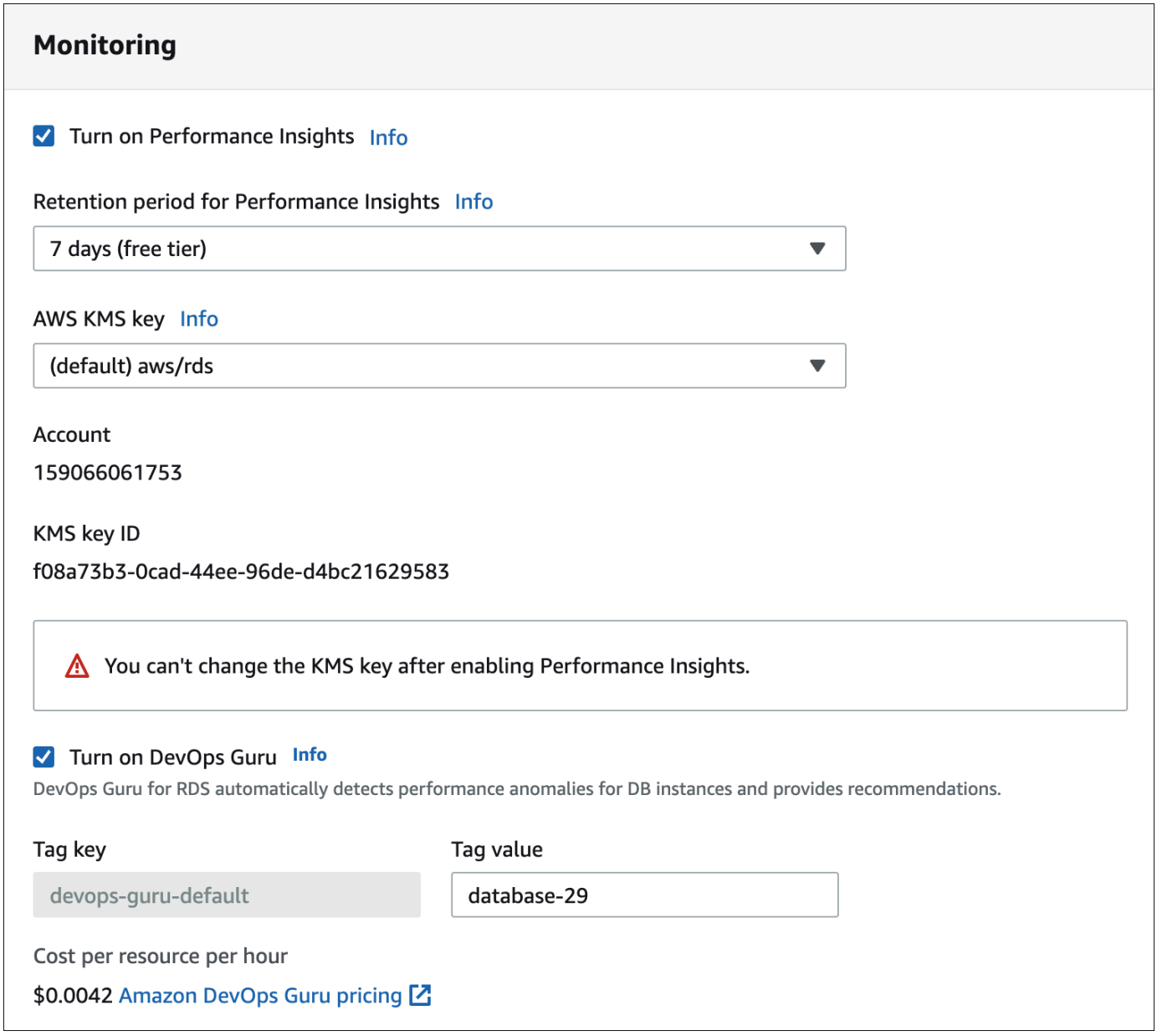
- Attivazione di DevOps Guru quando si crea un database per PostgreSQL
- Attivazione di DevOps Guru dal banner di notifica
- Risposta a un errore di autorizzazione quando attivi Guru DevOps

Attivazione di DevOps Guru quando si crea un database per PostgreSQL

Il flusso di lavoro di creazione include un'impostazione che attiva la copertura Guru per il database DevOps. Questa impostazione è abilitata per default quando scegli il modello Production (Produzione).

Per attivare DevOps Guru quando si crea un database per PostgreSQL

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Segui la procedura riportata in Creazione di un'istanza database, fino al passaggio, senza includerlo, in cui scegli le impostazioni di monitoraggio.
3. In Monitoring (Monitoraggio), scegli Turn on Performance Insights (Attiva Performance Insights). DevOpsAffinché Guru for RDS fornisca un'analisi dettagliata delle anomalie delle prestazioni, è necessario attivare Performance Insights.
4. Scegli Turn on Guru. DevOps



5. Crea un tag per il tuo database in modo che DevOps Guru possa monitorarlo. Esegui questa operazione:

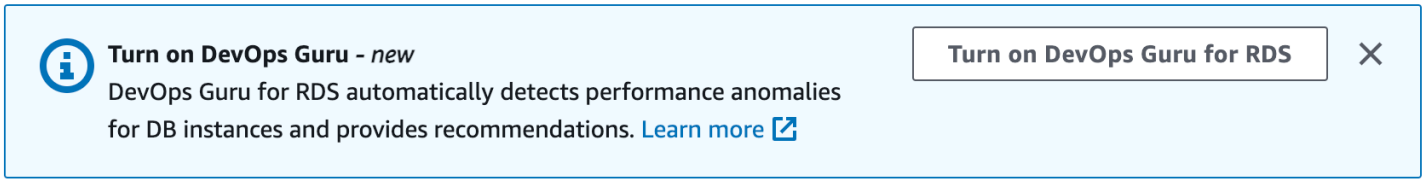
   - Nel campo di testo per Tag key (Chiave tag), inserisci un nome che inizi con **`Devops-Guru-`**.
   - Nel campo di testo per Tag value (Valore tag), inserisci qualsiasi valore. Ad esempio, se specifichi **`rds-database-1`** come nome del database RDS per PostgreSQL, puoi inserire anche **`rds-database-1`** come valore del tag.

   Per ulteriori informazioni sui tag, consulta "Usa i tag per identificare le risorse nelle tue applicazioni DevOps Guru" nella Amazon DevOps Guru User Guide.
6. Completare le fasi restanti in Creazione di un'istanza database.

Attivazione di DevOps Guru dal banner di notifica

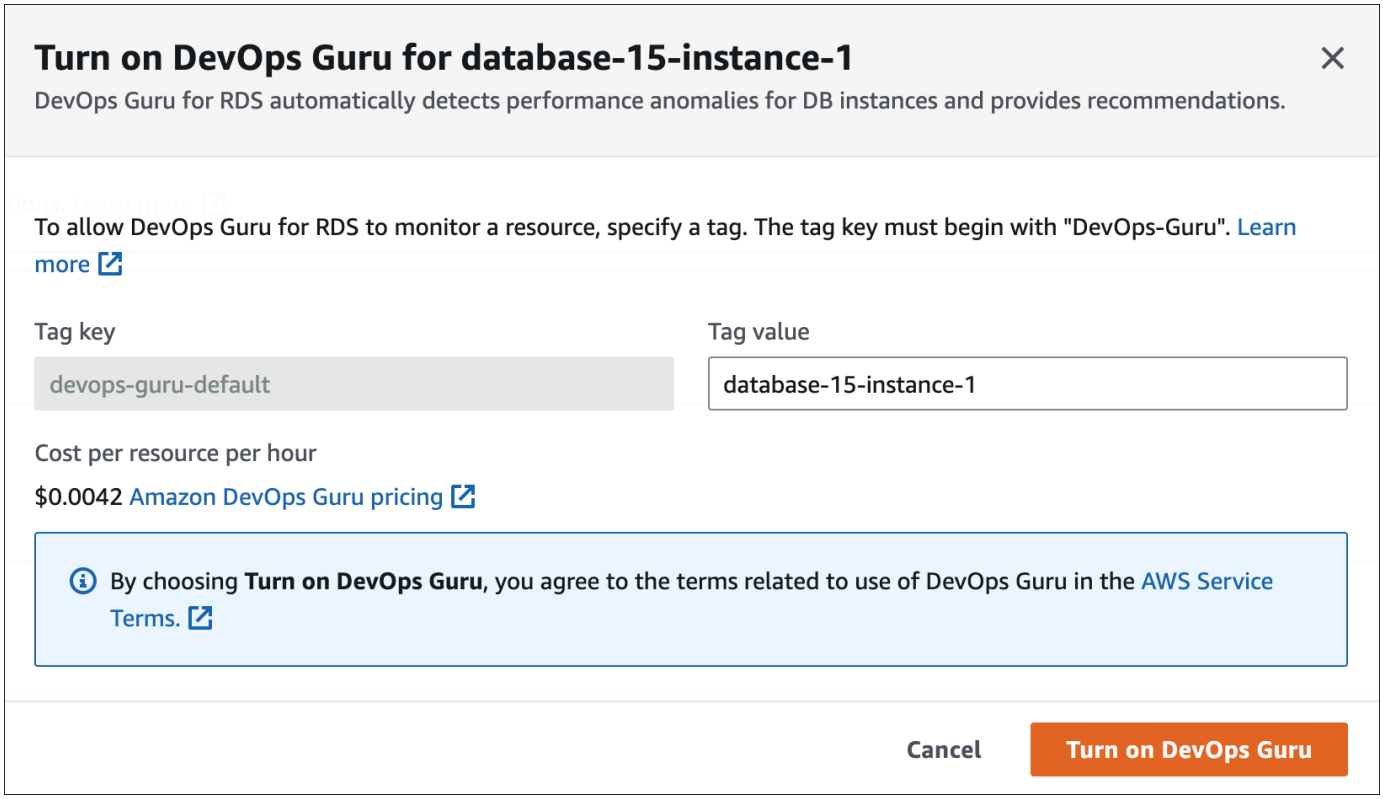
Se le tue risorse non sono coperte da DevOps Guru, Amazon RDS ti avvisa con un banner nelle seguenti posizioni:

- La scheda Monitoring (Monitoraggio) di un'istanza cluster database
- Pannello di controllo di Performance Insights



Per attivare DevOps Guru per il database per PostgreSQL

1. Nel banner, scegli Turn on Guru for RDS. DevOps
2. Immetti un nome e un valore della chiave tag. Per ulteriori informazioni sui tag, consulta "Usa i tag per identificare le risorse nelle tue applicazioni DevOps Guru" nella Amazon DevOps Guru User Guide.



3. Scegli Attiva Guru. DevOps

Risposta a un errore di autorizzazione quando attivi Guru DevOps

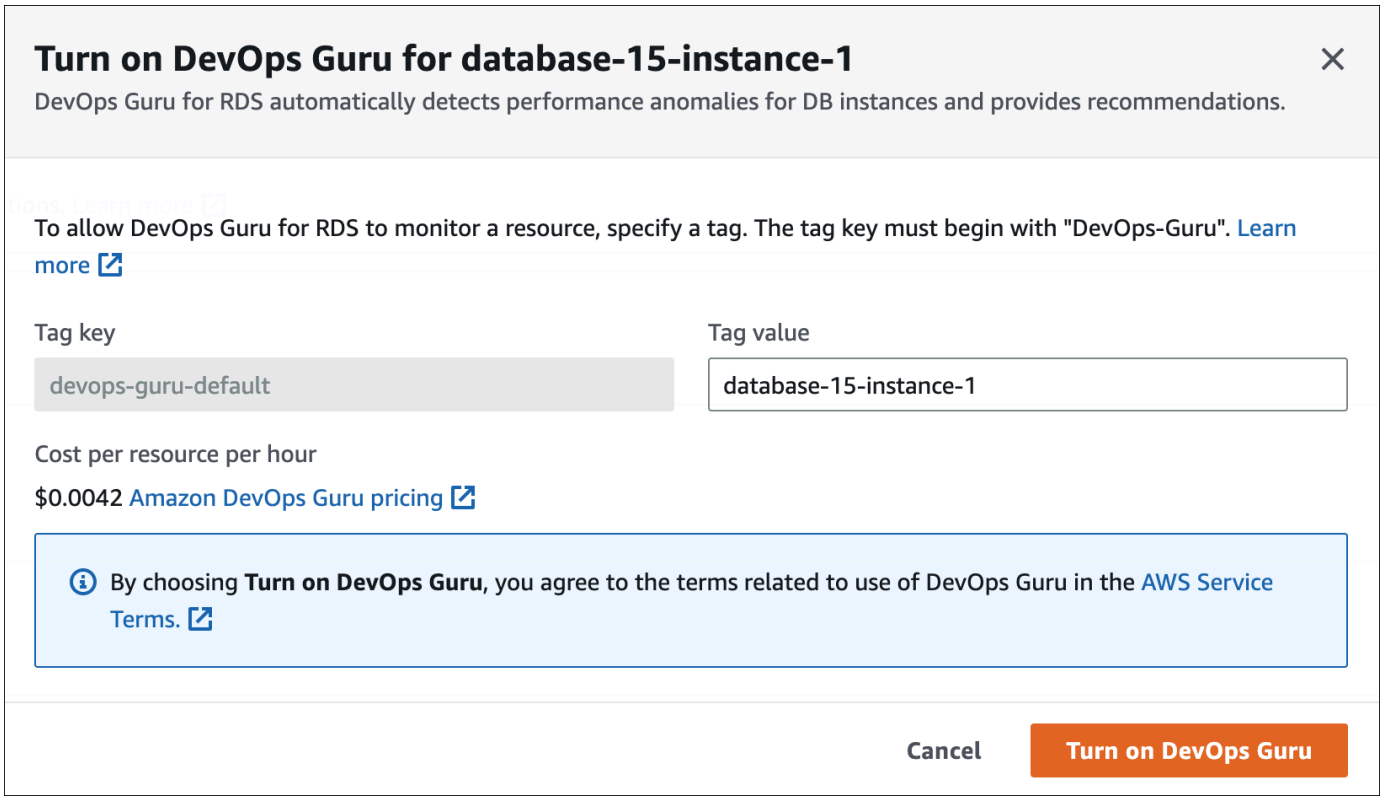
Se attivi DevOps Guru dalla console RDS quando crei un database, RDS potrebbe visualizzare il seguente banner relativo alle autorizzazioni mancanti.

Rispondere a un errore di autorizzazioni

1. Concedi all'utente o ruolo IAM il ruolo gestito dall'utente `AmazonDevOpsGuruConsoleFullAccess`. Per ulteriori informazioni, consulta Configurazione delle politiche di accesso IAM per Guru for RDS DevOps.
2. Aprire la console di RDS.
3. Nel pannello di navigazione scegli Approfondimenti sulle prestazioni.
4. Scegli un'istanza database nel cluster appena creato.
5. Scegli l'interruttore per attivare Guru for RDSDevOps.

DevOps Guru for RDS

6. Scegli un valore di tag. Per ulteriori informazioni, consulta "Usa i tag per identificare le risorse nelle tue applicazioni DevOps Guru" nella Amazon DevOps Guru User Guide.



7. Scegli Turn on Guru. DevOps

Aggiungere risorse per PostgreSQL nella console Guru DevOps

È possibile specificare la copertura delle risorse Guru sulla console DevOps Guru. DevOps Segui la procedura descritta in Specificare la copertura delle risorse DevOps Guru nella Amazon DevOps Guru User Guide. Quando modifichi le risorse analizzate, scegli una delle opzioni seguenti:

- Scegli Tutte le risorse dell'account per analizzare tutte le risorse supportate, inclusi RDS per PostgreSQL, nella tua regione. Account AWS
- Scegli CloudFormation gli stack per analizzare i database per PostgreSQL che si trovano negli stack che preferisci. Per ulteriori informazioni, consulta Usa gli AWS CloudFormation stack per identificare le risorse nelle tue applicazioni DevOps Guru nella Amazon Guru DevOps User Guide.
- Scegli Tag per analizzare i database RDS per PostgreSQL con tag. Per ulteriori informazioni, consulta Usa i tag per identificare le risorse nelle tue applicazioni DevOps Guru nella Amazon DevOps Guru User Guide.

Per ulteriori informazioni, consulta Enable DevOps Guru nella Amazon DevOps Guru User Guide.

Aggiungere risorse per PostgreSQL utilizzandoAWS CloudFormation

Puoi utilizzare i tag per aggiungere la copertura delle risorse RDS per PostgreSQL ai tuoi modelli. CloudFormation La procedura seguente presuppone che si disponga di un CloudFormation modello sia per l'istanza DB per PostgreSQL che per lo stack Guru. DevOps

Per specificare un'istanza DB RDS per PostgreSQL utilizzando un tag CloudFormation

1. Nel CloudFormation modello per l'istanza DB, definisci un tag utilizzando una coppia chiave/valore.

   L'esempio seguente assegna il valore `my-db-instance1` a `Devops-guru-cfn-default` per un'istanza database RDS per PostgreSQL.

   ```
   MyDBInstance1:
     Type: "AWS::RDS::DBInstance"
     Properties:
       DBInstanceIdentifier: my-db-instance1
       Tags:
         - Key: Devops-guru-cfn-default
           Value: devopsguru-my-db-instance1
   ```

2. Nel CloudFormation modello per il tuo stack DevOps Guru, specifica lo stesso tag nel filtro di raccolta delle risorse.

   L'esempio seguente configura DevOps Guru per fornire una copertura alla risorsa con il valore del tag. `my-db-instance1`

   ```
   DevOpsGuruResourceCollection:
     Type: AWS::DevOpsGuru::ResourceCollection
     Properties:
       ResourceCollectionFilter:
         Tags:
           - AppBoundaryKey: "Devops-guru-cfn-default"
             TagValues:
             - "devopsguru-my-db-instance1"
   ```

   Nell'esempio seguente si fornisce la copertura per tutte le risorse all'interno di `Devops-guru-cfn-default` del limite dell'applicazione.

```
DevOpsGuruResourceCollection:
  Type: AWS::DevOpsGuru::ResourceCollection
  Properties:
    ResourceCollectionFilter:
      Tags:
        - AppBoundaryKey: "Devops-guru-cfn-default"
          TagValues:
          - "*"
```

Per ulteriori informazioni, consulta AWS::DevOpsGuru::ResourceCollectione AWS: :RDS: :dbInstance nella Guida per l'utente.AWS CloudFormation

Monitoraggio dei parametri del sistema operativo con il monitoraggio avanzato

Con il monitoraggio avanzato, potete monitorare il sistema operativo del vostro database in tempo reale. I parametri di monitoraggio avanzato sono utili quando si desidera vedere come viene utilizzata la CPU in un'istanza database dai diversi processi o thread.

Argomenti

Panoramica sul monitoraggio avanzato

Amazon RDS fornisce parametri in tempo reale per il sistema operativo su cui è in esecuzione l'istanza di database. È possibile visualizzare tutti i parametri di sistema e le informazioni sui processi per le istanze del database RDS sulla console. È possibile gestire quali parametri si desidera monitorare per ogni istanza e personalizzare il pannello di controllo in base alle proprie esigenze. Per le descrizioni dei parametri di monitoraggio avanzato, consulta Parametri del sistema operativo nel monitoraggio avanzato.

RDS fornisce i parametri di Enhanced Monitoring al tuo account Amazon CloudWatch Logs. Puoi creare filtri per le metriche CloudWatch da CloudWatch Logs e visualizzare i grafici sulla dashboard. CloudWatch Puoi utilizzare l'output JSON di Enhanced Monitoring di CloudWatch Logs in un sistema di monitoraggio a tua scelta. Per ulteriori informazioni, consulta Monitoraggio avanzato nelle domande frequenti su Amazon RDS.

Argomenti

Enhanced Monitoring Availability (Disponibilità del monitoraggio avanzato)

Il monitoraggio avanzato è disponibile per i seguenti motori di database:

- Db2
- MariaDB
- Microsoft SQL Server
- MySQL
- Oracle
- PostgreSQL

Il monitoraggio avanzato è disponibile per tutte le classi di istanza database eccetto che per la classe di istanza db.m1.small.

Differenze tra CloudWatch e metriche di monitoraggio avanzato

Un hypervisor crea ed esegue macchine virtuali (VM). Utilizzando un hypervisor, un'istanza può supportare più macchine virtuali guest condividendo virtualmente memoria e CPU. CloudWatch raccoglie le metriche sull'utilizzo della CPU dall'hypervisor per un'istanza DB. Al contrario, Enhanced Monitoring raccoglie le metriche da un agente nell'istanza DB.

È possibile riscontrare differenze tra le misurazioni di Enhanced Monitoring CloudWatch e quelle di Enhanced Monitoring, poiché il livello dell'hypervisor esegue una piccola quantità di lavoro. Le differenze possono essere maggiori se le istanze DB utilizzano classi di istanza più piccole. In questo scenario, più macchine virtuali (VM) sono probabilmente gestite dal livello dell'hypervisor in una singola istanza fisica.

Per le descrizioni dei parametri di monitoraggio avanzato, consulta Parametri del sistema operativo nel monitoraggio avanzato. Per ulteriori informazioni sui CloudWatch parametri, consulta la Amazon CloudWatch User Guide.

Conservazione delle metriche di monitoraggio avanzato

Per impostazione predefinita, i parametri di Enhanced Monitoring vengono archiviati per 30 giorni nei CloudWatch log. Questo periodo di conservazione è diverso dalle metriche tipiche CloudWatch .

Per modificare la quantità di tempo in cui le metriche vengono archiviate nei CloudWatch log, modifica la conservazione per il gruppo di `RDSOSMetrics` log nella console. CloudWatch Per ulteriori

informazioni, consulta Change log data retention in CloudWatch logs nella Amazon CloudWatch Logs User Guide.

Costo di Enhanced Monitoring (monitoraggio avanzato)

I parametri di monitoraggio avanzato vengono archiviati nei CloudWatch log anziché nei parametri. CloudWatch Il costo del monitoraggio avanzato dipende dai seguenti fattori:

- L'Enhanced Monitoring ti verrà addebitato solo se superi il livello gratuito fornito da Amazon CloudWatch Logs. I costi si basano sulle CloudWatch tariffe di archiviazione e trasferimento dei dati dei log.
- La quantità di informazioni trasferite per un'istanza RDS è direttamente proporzionale alla granularità definita per la funzione di monitoraggio avanzato. Un intervallo di monitoraggio più piccolo comporta report più frequenti sui parametri del sistema operativo e aumenta i costi di monitoraggio. Per gestire i costi, imposta granularità diverse per istanze diverse nei tuoi account.
- I costi di utilizzo per il Monitoraggio avanzato vengono applicati a ciascuna istanza database per cui il monitoraggio avanzato è abilitato. Il monitoraggio di un numero elevato di istanze DB è più costoso rispetto al monitoraggio solo di pochi.
- Le istanze database che supportano un carico di lavoro ad alta intensità di elaborazione hanno più attività di processo del sistema operativo per generare report e costi più elevati per il monitoraggio avanzato.

Per ulteriori informazioni sui prezzi, consulta la pagina CloudWatch dei prezzi di Amazon.

Configurare e abilitare il monitoraggio avanzato

Per utilizzare il monitoraggio avanzato, è necessario creare un ruolo IAM e quindi abilitare il monitoraggio avanzato.

Argomenti

- Creazione di un ruolo IAM per Enhanced Monitoring
- Attivazione e disattivazione del monitoraggio avanzato
- Protezione dal problema del "confused deputy"

Creazione di un ruolo IAM per Enhanced Monitoring

Il monitoraggio avanzato richiede l'autorizzazione ad agire per conto dell'utente per inviare le informazioni sulle metriche del sistema operativo ai CloudWatch registri. Concedi le autorizzazioni di Enhanced Monitoring utilizzando un ruolo AWS Identity and Access Management (IAM). È possibile creare questo ruolo quando si abilita il monitoraggio avanzato o lo si crea in anticipo.

Argomenti

Creazione del ruolo IAM quando si attiva Enhanced Monitoring

Quando si attiva Enhanced Monitoring nella console RDS, Amazon RDS è possibile creare il ruolo IAM necessario. Il ruolo è denominato `rds-monitoring-role`. RDS utilizza questo ruolo per l'istanza database specificata, la replica di lettura o il cluster di database Multi-AZ.

Per creare il ruolo IAM quando si attiva Enhanced Monitoring

1. Segui la procedura riportata in Attivazione e disattivazione del monitoraggio avanzato.
2. Imposta Ruolo di monitoraggio su Predefinito nel passaggio in cui si sceglie un ruolo.

Creazione del ruolo IAM prima di abilitare Enhanced Monitoring

È possibile creare il ruolo richiesto prima di abilitare Enhanced Monitoring. Quando si abilita Enhanced Monitoring, specifica il nome del nuovo ruolo. Si deve creare questo ruolo necessario se si abilita il monitoraggio avanzato utilizzando AWS CLI oppure l'API RDS.

L'utente che abilita il monitoraggio avanzato deve ricevere l'autorizzazione `PassRole`. Per ulteriori informazioni, consulta l'Esempio 2 in Concessione a un utente delle autorizzazioni per il trasferimento di un ruolo a un AWS servizio nella Guida per l'utente IAM.

Per creare un ruolo IAM per Amazon RDS Enhanced Monitoring

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com.
2. Nel pannello di navigazione, seleziona Roles (Ruoli).
3. Selezionare Create role (Crea ruolo).
4. Scegliere la scheda Servizio AWS quindi seleziona RDS dall’elenco di servizi.

5. Scegli RDS - Enhanced Monitoring (RDS - Monitoraggio avanzato), quindi seleziona Next (Avanti).
6. Assicurati che le politiche di autorizzazione indichino AmazonRDS EnhancedMonitoringRole, quindi scegli Avanti.
7. In Nome ruolo, immetti un nome per il ruolo. Ad esempio, specifica **`emaccess`**.

   L'entità affidabile per il tuo ruolo è il servizio monitoring.rds.amazonaws.com. AWS
8. Scegli Crea ruolo.

Attivazione e disattivazione del monitoraggio avanzato

Puoi attivare e disattivare il monitoraggio avanzato utilizzando l'API, o RDS. AWS Management Console AWS CLI Scegli le istanze database RDS in cui desideri attivare il monitoraggio avanzato. È possibile impostare granularità diverse per la raccolta di parametri su ogni istanza database.

Console

È possibile attivare il monitoraggio avanzato quando si crea un'istanza database, un cluster di database multi-AZ, o una replica di lettura oppure quando si modifica un'istanza database o un cluster database multi-AZ. Se modifichi un'istanza database per attivare il monitoraggio avanzato, non devi riavviare l'istanza database per rendere effettive le modifiche.

È possibile abilitare il monitoraggio avanzato nella console RDS quando si esegue una delle seguenti operazioni nella pagina Databases (Database):

- Creazione di un’istanza database o un cluster di database Multi-AZ: scegli Create database (Crea database).
- Creazione di una replica di lettura: scegli Actions (Operazioni), quindi Create read replica (Crea replica di lettura).
- Modifica di un'istanza database o di un cluster di database Multi-AZ: scegli Modify (Modifica).

Per attivare o disattivare il monitoraggio avanzato nella console RDS

1. Scorri fino a Additional configuration (Configurazione aggiuntiva).
2. In Monitoring (Monitoraggio), scegli Enable Enhanced Monitoring (Abilita monitoraggio avanzato) per l’istanza database o la replica di lettura. Per disattivare il monitoraggio avanzato, scegli Disable enhanced monitoring (Disabilita monitoraggio avanzato).

3. Imposta la proprietà Monitoring Role sul ruolo IAM che hai creato per consentire ad Amazon RDS di comunicare con Amazon CloudWatch Logs per te, oppure scegli Default per fare in modo che RDS crei un ruolo per te denominato. `rds-monitoring-role`
4. Impostare la proprietà Granularity (Granularità) sull'intervallo, in secondi, tra i punti quando i parametri vengono raccolti per l'istanza database o la replica di lettura. La proprietà Granularity (Granularità) può essere impostata su uno dei valori seguenti: `1`, `5`, `10`, `15`, `30` oppure `60`.

   L'intervallo più veloce in cui la console RDS si aggiorna è ogni 5 secondi. Se si imposta la granularità su 1 secondo nella console RDS, vengono comunque visualizzati parametri aggiornati solo ogni 5 secondi. Puoi recuperare gli aggiornamenti dei parametri in 1 secondo utilizzando Logs. CloudWatch

AWS CLI

Per attivare il monitoraggio avanzato utilizzando i comandi seguenti AWS CLI, impostate l'`--monitoring-interval`opzione su un valore diverso da `0` e impostate l'`--monitoring-role-arn`opzione sul ruolo in cui avete creato. Creazione di un ruolo IAM per Enhanced Monitoring

- create-db-instance
- create-db-instance-read-replica
- modify-db-instance
- create-db-cluster(cluster DB Multi-AZ)
- modify-db-cluster(cluster DB Multi-AZ)

L'opzione `--monitoring-interval` specifica l'intervallo, in secondi, tra i punti quando vengono raccolti i parametri di monitoraggio avanzato. I valori validi per l'opzione sono `0`, `1`, `5`, `10`, `15`, `30` e `60`.

Per disattivare il monitoraggio avanzato utilizzando il AWS CLI, imposta l'`--monitoring-interval`opzione su `0` in questi comandi.

Example

Nell'esempio seguente viene attivato il monitoraggio avanzato per un'istanza database:

Per LinuxmacOS, oUnix:

```
aws rds modify-db-instance \
```

```
    --db-instance-identifier mydbinstance \
    --monitoring-interval 30 \
    --monitoring-role-arn arn:aws:iam::123456789012:role/emaccess
```

Per Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --monitoring-interval 30 ^
    --monitoring-role-arn arn:aws:iam::123456789012:role/emaccess
```

Example

Nell'esempio seguente viene attivato il monitoraggio avanzato per un cluster di database Multi-AZ:

Per LinuxmacOS, oUnix:

```
aws rds modify-db-cluster \
    --db-cluster-identifier mydbcluster \
    --monitoring-interval 30 \
    --monitoring-role-arn arn:aws:iam::123456789012:role/emaccess
```

Per Windows:

```
aws rds modify-db-cluster ^
    --db-cluster-identifier mydbcluster ^
    --monitoring-interval 30 ^
    --monitoring-role-arn arn:aws:iam::123456789012:role/emaccess
```

API RDS

Per attivare il monitoraggio avanzato utilizzando l'API RDS, imposta il parametro `MonitoringInterval` su un valore diverso da `0` e imposta il parametro `MonitoringRoleArn` sul ruolo creato in Creazione di un ruolo IAM per Enhanced Monitoring. Imposta questi parametri nelle seguenti operazioni:

- CreateDBInstance
- Creato B InstanceReadReplica
- ModifyDBInstance

- CreateDBCluster (Cluster di database Multi-AZ)
- ModifyDBCluster (Cluster di database Multi-AZ)

Il parametro `MonitoringInterval` specifica l'intervallo, in secondi, tra i punti quando vengono raccolti i parametri di monitoraggio avanzato. I valori validi sono `0`, `1`, `5`, `10`, `15`, `30` e `60`.

Per disattivare il monitoraggio avanzato utilizzando l'API RDS, imposta `MonitoringInterval` su `0`.

Protezione dal problema del "confused deputy"

Con "confused deputy" si intende un problema di sicurezza in cui un'entità che non dispone dell'autorizzazione per eseguire una certa operazione può costringere un'entità con più privilegi a eseguire tale operazione. Nel AWS, l'impersonificazione tra servizi può portare al confuso problema del vice. La rappresentazione tra servizi può verificarsi quando un servizio (il servizio chiamante) effettua una chiamata a un altro servizio (il servizio chiamato). Il servizio chiamante può essere manipolato per utilizzare le proprie autorizzazioni e agire sulle risorse di un altro cliente, a cui normalmente non avrebbe accesso. Per evitare ciò, AWS fornisce strumenti per poterti a proteggere i tuoi dati per tutti i servizi con entità di servizio a cui è stato concesso l'accesso alle risorse del tuo account. Per ulteriori informazioni, consulta Problema del "confused deputy".

Per limitare le autorizzazioni relative alle risorse che Amazon RDS può fornire a un altro servizio, si consiglia di utilizzare le chiavi di contesto delle condizioni globali `aws:SourceArn` e `aws:SourceAccount` in una policy di attendibilità per il tuo ruolo di monitoraggio avanzato. Se si utilizzano entrambe le chiavi di contesto delle condizioni globali, devono utilizzare lo stesso ID account.

Il modo più efficace per proteggersi dal problema "confused deputy" è quello di usare la chiave di contesto della condizione globale `aws:SourceArn` con l'ARN completo della risorsa. Per Amazon RDS, imposta `aws:SourceArn` su `arn:aws:rds:`*`Region`*`:`*`my-account-id`*`:db:`*`dbname`*.

L'esempio seguente usa le chiavi di contesto delle condizioni globali `aws:SourceArn` e `aws:SourceAccount` in una policy di affidabilità per prevenire il problema del "confused deputy".

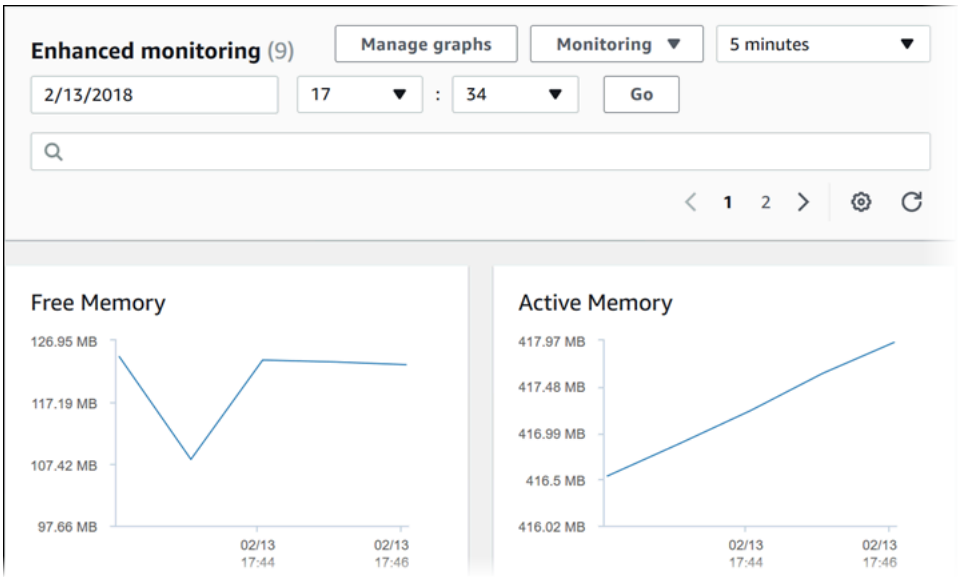
```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
```

```
        "Service": "monitoring.rds.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringLike": {
          "aws:SourceArn": "arn:aws:rds:Region:my-account-id:db:dbname"
        },
        "StringEquals": {
          "aws:SourceAccount": "my-account-id"
        }
      }
    }
  ]
}
```

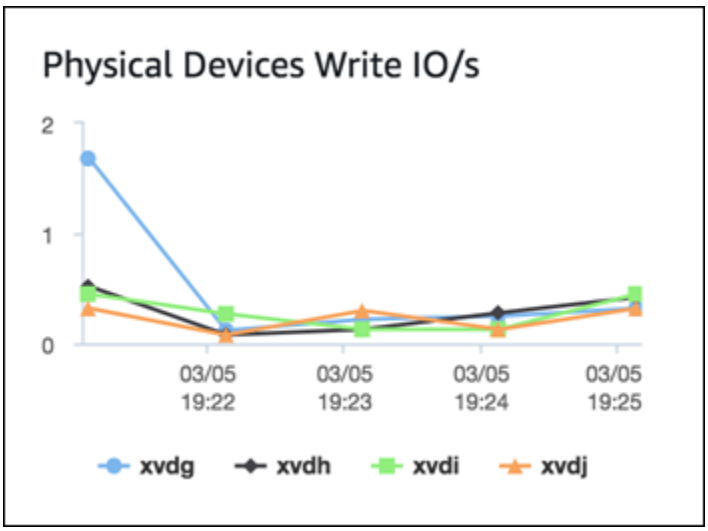
Visualizzazione dei parametri nella console RDS

Puoi visualizzare i parametri del sistema operativo segnalati dal monitoraggio avanzato nella console RDS scegliendo Enhanced monitoring (Monitoraggio avanzato) per Monitoring (Monitoraggio).

L'esempio seguente mostra la pagina Enhanced Monitoring (Monitoraggio avanzato). Per le descrizioni dei parametri di monitoraggio avanzato, consulta Parametri del sistema operativo nel monitoraggio avanzato.

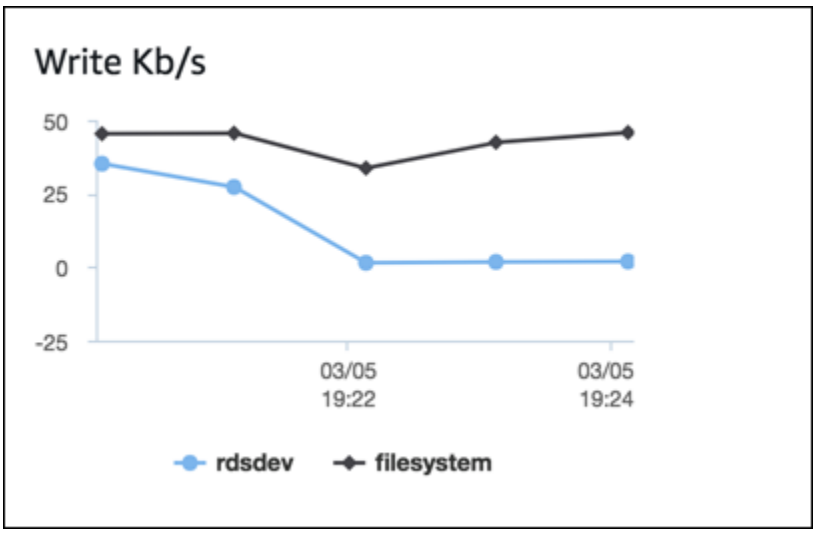


Alcune istanze database utilizzano più di un disco per il volume di storage dei dati dell'istanza database. Su tali istanze database, i grafici Physical Devices (Dispositivi fisici) mostrano i parametri per ciascun disco. Ad esempio, il grafico seguente mostra i parametri per quattro dischi.



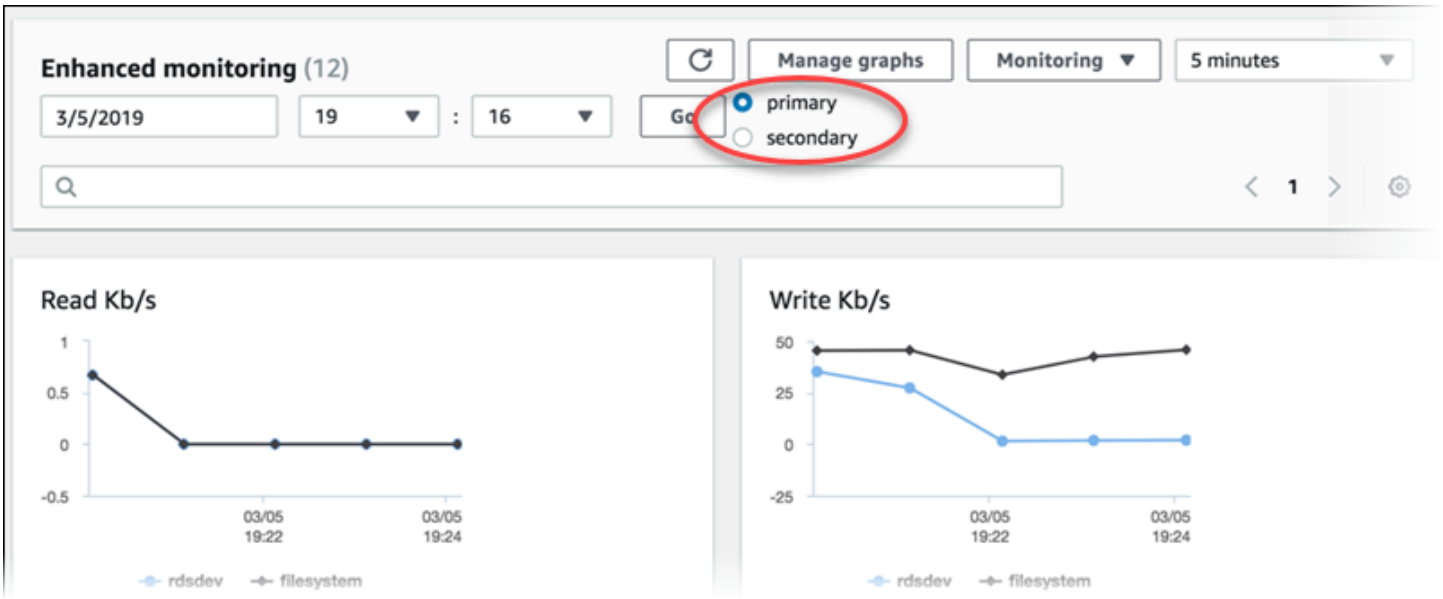
> **Note**
>
> Al momento, i grafici Physical Devices (Dispositivi fisici) non sono disponibili per le istanze database di Microsoft SQL Server.

Quando stai visualizzando grafici I/O su disco e File system aggregati, il dispositivo rdsdev fa riferimento al file system `/rdsdbdata`, dove sono archiviati tutti i log e i file del database. Il dispositivo filesystem fa riferimento al file system `/` (noto anche come root), dove sono archiviati i file correlati al sistema operativo.



Se l'istanza DB è un'implementazione Multi-AZ, puoi visualizzare i parametri del sistema operativo per l'istanza database primaria e la sua replica di standby Multi-AZ. Nella vista Enhanced monitoring (Monitoraggio avanzato), scegliere primary (primario) per visualizzare i parametri del sistema

operativo per l'istanza database primaria, altrimenti selezionare secondary (secondario) per visualizzare i parametri del sistema operativo per la replica di standby.

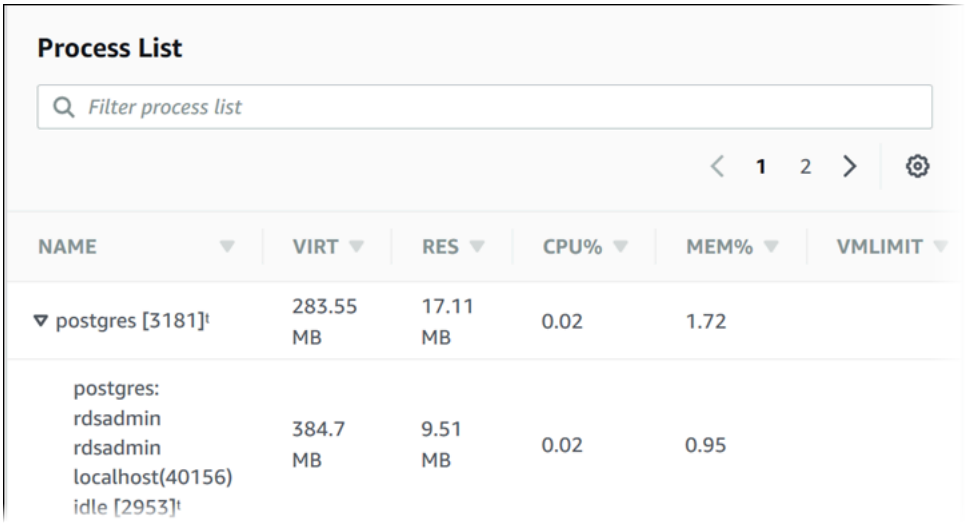


Per ulteriori informazioni sulle implementazioni Multi-AZ, consulta Configurazione e gestione di un'implementazione multi-AZ.

> **Note**
>
> Attualmente, la visualizzazione dei parametri del sistema operativo per una replica di standby Multi-AZ non è supportata per le istanze database MariaDB.

Se desideri vedere i dettagli per i processi in esecuzione nell'istanza database, scegli OS process list (Elenco processi sistema operativo) per Monitoring (Monitoraggio).

La vista Process List (Elenco processi) è mostrata di seguito.



I parametri del monitoraggio avanzato mostrati nella vista Process List (Elenco processi) sono organizzati nel seguente modo:

- RDS child processes (Processi figlio RDS) – Viene visualizzato un riepilogo dei processi RDS che supportano l'istanza database, ad esempio `mysqld` per le istanze database di MySQL. I thread del processo appaiono nidificati sotto il processo genitore. I thread del processo mostrano l'utilizzo della CPU solo quando gli altri parametri sono uguali per tutti i thread per il processo. La console visualizza un massimo di 100 processi e thread. I risultati sono una combinazione dei principali processi e thread CPU che consumano memoria. Se ci sono più di 50 processi e più di 50 thread, la console visualizza i primi 50 consumatori in ciascuna categoria. Questo display aiuta a identificare quali processi stanno avendo il maggiore impatto sulle prestazioni.
- Processi RDS: viene visualizzato un riepilogo delle risorse utilizzate dall'agente di gestione RDS, dei processi di monitoraggio della diagnostica e di altri processi AWS necessari per supportare le istanze database RDS.
- OS processes (Processi del sistema operativo) – Viene visualizzato un riepilogo dei processi del kernel e di sistema, che generalmente hanno un impatto minimo sulle prestazioni.

Gli elementi elencati per ogni processo sono:

- VIRT – Indica la dimensione virtuale del processo.
- RES – Indica la memoria fisica effettiva utilizzata dal processo.
- CPU% – Indica la percentuale della larghezza di banda totale della CPU utilizzata dal processo.
- MEM% – Indica la percentuale della memoria totale utilizzata dal processo.

I dati di monitoraggio visualizzati nella console RDS sono recuperati dalla Amazon CloudWatch Logs. È anche possibile recuperare i parametri per un'istanza database come un flusso di log CloudWatch Logs. Per ulteriori informazioni, consulta Visualizzazione dell'utilizzo dei parametri del sistema operativo CloudWatch Logs.

I parametri di monitoraggio avanzato non vengono restituiti durante:

- Un failover dell'istanza database.
- Modifica della classe di istanza dell'istanza database (dimensionamento del calcolo).

I parametri del monitoraggio avanzato vengono restituiti durante un riavvio di un'istanza DB perché viene riavviato solo il motore del database. I parametri per il sistema operativo vengono ancora segnalati.

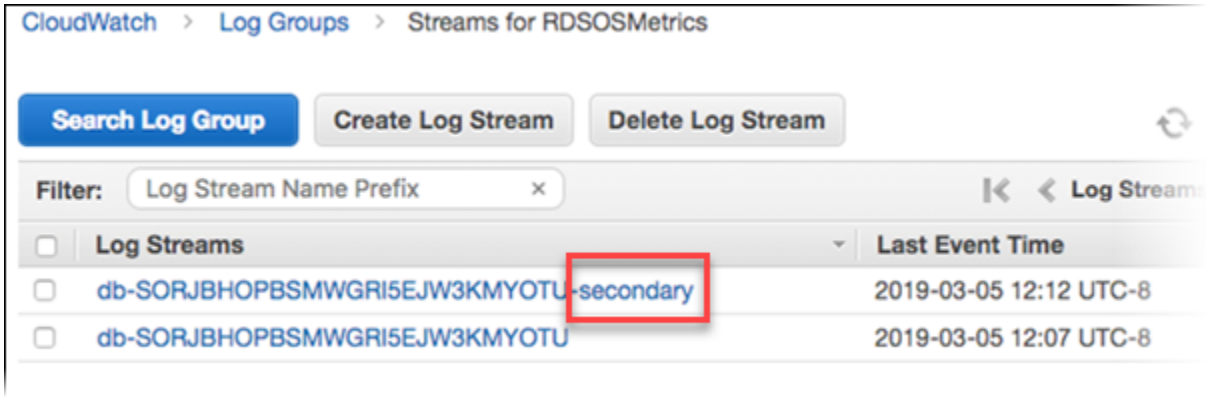
Visualizzazione dell'utilizzo dei parametri del sistema operativo CloudWatch Logs

Dopo aver abilitato il monitoraggio avanzato per l'istanza database o il cluster di database Multi-AZ, è possibile visualizzare i relativi parametri utilizzando CloudWatch Logs, con ogni flusso di log che rappresenta una singola istanza database monitorata o cluster di database monitorato. L'identificatore del flusso di log è l'identificativo della risorsa (`DbiResourceId`) per l'istanza database o il cluster di database.

Per visualizzare i dati di log del Monitoraggio avanzato

1. Aprire la console CloudWatch all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Se necessario, scegliere la Regione AWS in cui si trova l'istanza database o il cluster di database Multi-AZ. Per ulteriori informazioni, consulta la pagina relativa a regioni ed endpoint nei Riferimenti generali di Amazon Web Services.
3. Selezionare Logs (Log) nel riquadro di navigazione.
4. Selezionare RDSOSMetrics nell'elenco di gruppi di log.

   In un'implementazione istanza database Multi-AZ, i file di log con `-secondary` aggiunto al nome sono per la replica di standby Multi-AZ.



5. Scegliere il flusso di log che si desidera visualizzare dall´elenco dei flussi di log.

Riferimento per i parametri per Amazon RDS

In questo riferimento, è possibile trovare descrizioni dei parametri di Amazon RDS per Amazon CloudWatch, Performance Insights e monitoraggio avanzato.

Argomenti

CloudWatch Parametri Amazon per Amazon RDS

Amazon RDS pubblica i parametri su Amazon CloudWatch nei namespace e. `AWS/RDS` `AWS/Usage`

Argomenti

Parametri a CloudWatch livello di istanza Amazon per Amazon RDS

Il `AWS/RDS` namespace in Amazon CloudWatch include i seguenti parametri a livello di istanza.

> **Note**
>
> La console Amazon RDS potrebbe visualizzare i parametri in unità diverse da quelle inviate ad Amazon. CloudWatch Ad esempio, la console Amazon RDS potrebbe visualizzare una metrica in megabyte (MB), mentre la metrica viene inviata ad Amazon in byte. CloudWatch

Parametro	Descrizione	Si applica a	Unità
`BinLogDiskUsage`	La quantità di spazio su disco occupata dai registri binari. Se i backup automatic	MariaDB	Byte

Parametro	Descrizione	Si applica a	Unità
	i sono abilitati per le istanze MySQL e MariaDB, incluse le repliche di lettura, vengono creati i log binari.	MySQL	
`BurstBalance`	La percentuale di crediti I/O General Purpose SSD (gp2) burst-bucket disponibili.	Tutti	Percentuale
`CheckpointLag`	Il tempo trascorso dal checkpoint più recente.		Secondi
`ConnectionAttempts`	Il numero di tentativi di connessione a un'istanza, a prescindere che vadano a buon fine.	MySQL	Conteggio
`CPUUtilization`	La percentuale di utilizzo della CPU.	Tutti	Percentuale

Parametro	Descrizione	Si applica a	Unità
`CPUCreditUsage`	Il numero di crediti CPU spesi dall'istanza per l'utilizzo della CPU. Un credito CPU equivale a una vCPU in esecuzione al 100 percento di utilizzo per un minuto o una combinazione equivalente di vCPU, utilizzo e tempo. Ad esempio, potresti avere una vCPU in esecuzione al 50 percento di utilizzo per due minuti o due vCPU in esecuzione al 25 percento di utilizzo per due minuti. Questa metrica si applica solo alle istanze e. `db.t2` `db.t3` `db.t4g` **Note** Consigliamo di utilizzare le classi di istanza database T solo per i server di sviluppo e test o altri server non di produzione. Per ulteriori dettagli sulle classi di istanze T, vedere Tipi di classi di istanza database I parametri di credito CPU sono disponibili solo con una frequenza di 5 minuti. Se specifichi un periodo superiore a 5 minuti, usa la statistica `Sum` al posto di quella `Average`.		Crediti (vCPU/minuti)

Parametro	Descrizione	Si applica a	Unità
`CPUCredit Balance`	Il numero di crediti CPU ottenuti, che un'istanza ha accumulato da quando è stata lanciata o avviata. Per le T2 Standard `CPUCreditBalance` include anche il numero di crediti di lancio che sono stati accumulati. I crediti vengono accumulati nel saldo del credito dopo che sono stati ottenuti e rimossi dal saldo del credito una volta spesi. Il saldo del credito ha un limite massimo, determinato dalla dimensione dell'istanza. Una volta che il limite viene raggiunto, tutti i nuovi crediti guadagnat i vengono scartati. Per le T2 Standard, i crediti di lancio non contano per il limite. I crediti in `CPUCreditBalance` sono disponibili affinché l'istanza li spenda per andare oltre l'utilizzo di base della CPU. Quando l'istanza non è in fase di esecuzione, i crediti in `CPUCredit Balance` non scadono. Quando l'istanza si arresta, il `CPUCredit Balance` non persiste e tutti i crediti accumulati vengono persi. I parametri di credito CPU sono disponibili solo con una frequenza di 5 minuti. Questa metrica si applica solo alle `db.t2` `db.t4g` istanze e. `db.t3`		Crediti (vCPU/ minuti)

Parametro	Descrizione	Si applica a	Unità
	Note Consigliamo di utilizzare le classi di istanza database T solo per i server di sviluppo e test o altri server non di produzione. Per ulteriori dettagli sulle classi di istanze T, vedere Tipi di classi di istanza database I crediti di lancio funzionano in Amazon RDS allo stesso modo che in Amazon EC2. Per ulteriori informazioni, consulta Crediti di lancio nella Guida per l'utente di Amazon Elastic Compute Cloud per istanze Linux.		

Parametro	Descrizione	Si applica a	Unità
`CPUSurplusCreditBalance`	Il numero di crediti extra spesi da un'istanza illimitata quando il rispettivo valore `CPUCreditBalance` è pari a zero. Il valore `CPUSurplusCreditBalance` viene saldato con i crediti CPU ottenuti. Se il numero dei crediti extra va oltre il numero massimo di crediti che un'istanza può ottenere in un periodo di 24 ore, i crediti extra spesi, eccedenti il limite, incorreranno in costi aggiuntivi. I parametri di credito CPU sono disponibili solo con una frequenza di 5 minuti.	Tutti	Crediti (vCPU/minuti)

Parametro	Descrizione	Si applica a	Unità
`CPUSurplusCreditsCharged`	Il numero di crediti extra spesi da un'istanza, che non sono saldati con i crediti CPU ottenuti e che pertanto incorrono in costi aggiuntivi. I crediti extra spesi subiscono costi aggiuntivi quando si verifica uno dei seguenti casi: • I crediti extra spesi vanno oltre il numero massimo di crediti che un'istanza può ottenere in un periodo di 24 ore. I crediti extra spesi, che eccedono il limite, subiscono costi aggiuntivi alla fine dell'ora; • l'istanza viene arrestata o terminata; • l'istanza passa da `unlimited` a `standard`. I parametri di credito CPU sono disponibili solo con una frequenza di 5 minuti.	Tutti	Crediti (vCPU/minuti)

Parametro	Descrizione	Si applica a	Unità
`DatabaseConnections`	Il numero di connessioni di rete client all'istanza del database. Il numero di sessioni del database può essere superiore al valore del parametro perché il valore del parametro non include quanto segue: • Sessioni che non hanno più una connessione di rete ma che il database non ha ripulito • Sessioni create dal motore del database per i propri scopi • Sessioni create dalle funzionalità di esecuzione parallela del motore del database • Sessioni create dal pianificatore dei processi del motore del database • Connessioni ad Amazon RDS	Tutti	Conteggio
`DiskQueueDepth`	Il numero di I/O (richieste di lettura/s crittura) in sospeso che sono in attesa di accedere al disco.	Tutti	Conteggio
`DiskQueueDepthLogVolume`	Il numero di I/O (richieste di lettura/s crittura) in sospeso che sono in attesa di accedere al disco del volume di log.	Tutti	Conteggio

Parametro	Descrizione	Si applica a	Unità
`EBSByteBalance%`	La percentuale di crediti di throughput rimanenti nel bucket continuo del database RDS. Questo parametro è disponibile solo per il monitoraggio base. Il valore del parametro si basa sulla velocità di trasmissione effettiva di tutti i volumi, incluso il volume root, anziché solo sui volumi contenenti file di database. Per trovare le dimensioni delle istanze che supportano questo parametro, consulta le dimensioni delle istanze con un asterisco (*) nella tabella EBS ottimizzata per impostazione predefinita in Guida per l'utente di Amazon EC2 per le istanze Linux. La statistica `Sum` non è applicabile a questo parametro.	Tutti	Percentuale

Parametro	Descrizione	Si applica a	Unità
`EBSIOBalance%`	La percentuale di crediti I/O rimanenti nel bucket di frammentazione del database RDS. Questo parametro è disponibile solo per il monitoraggio base. Il valore del parametro si basa sull'IOPS di tutti i volumi, incluso il volume root, anziché solo sui volumi contenenti file di database. Per trovare le dimensioni delle istanze che supportano questo parametro, consulta le dimensioni delle istanze con un asterisco (*) nella tabella EBS ottimizzata per impostazione predefinita in Guida per l'utente di Amazon EC2 per le istanze Linux. La statistica `Sum` non è applicabile a questo parametro. Questo parametro è diverso da `BurstBalance` . Per informazioni su come utilizzare questo parametro, consulta Miglioramento delle prestazioni delle applicazioni e riduzione dei costi con la funzionalità di frammentazione istanze ottimizzata per Amazon EBS.	Tutti	Percentuale
`FailedSQLServerAgentJobsCount`	Il numero di processi Microsoft SQL Server Agent falliti nel corso dell'ultimo minuto.	Microsoft SQL Server	Conteggio al minuto

Parametro	Descrizione	Si applica a	Unità
`FreeableMemory`	La quantità di memoria RAM disponibile. Per le istanze di database MariaDB, MySQL, Oracle e PostgreSQL DB, questo parametro segnala il valore del campo `MemAvailable` di `/proc/meminfo` .	Tutti	Byte
`FreeLocalStorage`	La quantità di spazio di archiviazione locale disponibile. Questa metrica si applica solo alle classi di istanze database con volumi di archivio dell'istanza NVMe SSD. Per informazioni sulle istanze Amazon EC2 con volumi di archivio dell'istanza NVMe SSD, consulta Volumi dell'archivio dell'istanza. Le classi di istanze database di RDS equivalenti hanno gli stessi volumi di archivio dell'istanza. Ad esempio, le classi di istanze database db.m6gd e db.r6gd hanno volumi di archivio dell'istanza NVMe SSD.		Byte
`FreeStorageSpace`	La quantità di spazio di storage disponibile.	Tutti	Byte
`FreeStorageSpaceLogVolume`	La quantità di spazio di archiviazione disponibile nel volume di log.	Tutti	Byte
`MaximumUsedTransactionIDs`	Il numero massimo di ID di transazione che sono stati utilizzati.	PostgreSQL	Conteggio

Parametro	Descrizione	Si applica a	Unità
`NetworkReceiveThroughput`	Il traffico di rete in entrata (ricezione) sull'istanza database, inclusi il traffico del database del cliente e il traffico di Amazon RDS utilizzati per attività di monitoraggio e replica.	Tutti	Byte al secondo
`NetworkTransmitThroughput`	Il traffico di rete in uscita (trasmissione) sull'istanza database, inclusi il traffico del database del cliente e il traffico di Amazon RDS utilizzati per attività di monitoraggio e replica.	Tutti	Byte al secondo
`OldestReplicationSlotLag`	L'entità del ritardo della replica più in ritardo in termini di dati WAL ricevuti.	PostgreSQL	Byte
`ReadIOPS`	Il numero medio di operazioni di I/O di lettura del disco al secondo.	Tutti	Conteggio al secondo
`ReadIOPSLocalStorage`	Il numero medio di operazioni di I/O di lettura del disco verso l'archiviazione locale al secondo. Questa metrica si applica solo alle classi di istanze database con volumi di archivio dell'istanza NVMe SSD. Per informazioni sulle istanze Amazon EC2 con volumi di archivio dell'istanza NVMe SSD, consulta Volumi dell'archivio dell'istanza. Le classi di istanze database di RDS equivalenti hanno gli stessi volumi di archivio dell'istanza. Ad esempio, le classi di istanze database db.m6gd e db.r6gd hanno volumi di archivio dell'istanza NVMe SSD.		Conteggio al secondo

Parametro	Descrizione	Si applica a	Unità
`ReadLatency`	Il numero medio di operazioni di I/O di lettura del disco al secondo per il volume di log.	Tutti	Conteggio al secondo
`ReadIOPSLogVolume`	La quantità di tempo media che occorre per ciascuna operazione I/O su disco.	Tutti	Secondi
`ReadLatencyLocalStorage`	La quantità di tempo media che occorre per ciascuna operazione I/O su disco per archiviazione locale. Questa metrica si applica solo alle classi di istanze database con volumi di archivio dell'istanza NVMe SSD. Per informazioni sulle istanze Amazon EC2 con volumi di archivio dell'istanza NVMe SSD, consulta Volumi dell'archivio dell'istanza. Le classi di istanze database di RDS equivalenti hanno gli stessi volumi di archivio dell'istanza. Ad esempio, le classi di istanze database db.m6gd e db.r6gd hanno volumi di archivio dell'istanza NVMe SSD.		Secondi
`ReadLatencyLogVolume`	La quantità di tempo media che occorre per ciascuna operazione I/O su disco per il volume di log.	Tutti	Secondi
`ReadThroughput`	Il numero medio di byte letti dal disco al secondo.	Tutti	Byte al secondo

Parametro	Descrizione	Si applica a	Unità
`ReadThroughputLocalStorage`	Il numero medio di byte letti dal disco al secondo per l'archiviazione locale. Questa metrica si applica solo alle classi di istanze database con volumi di archivio dell'istanza NVMe SSD. Per informazioni sulle istanze Amazon EC2 con volumi di archivio dell'istanza NVMe SSD, consulta Volumi dell'archivio dell'istanza. Le classi di istanze database di RDS equivalenti hanno gli stessi volumi di archivio dell'istanza. Ad esempio, le classi di istanze database db.m6gd e db.r6gd hanno volumi di archivio dell'istanza NVMe SSD.		Byte al secondo
`ReadThroughputLogVolume`	Il numero medio di byte letti dal disco al secondo per il volume di log.	Tutti	Byte al secondo
`ReplicaLag`	Per le configurazioni di replica di lettura, il tempo di ritardo di un'istanza database di replica di lettura rispetto all'istanza database di origine. Si applica a repliche di lettura MariaDB, Microsoft SQL Server, MySQL, Oracle e PostgreSQL. Per i cluster di database Multi-AZ, la differenza di tempo tra l'ultima transazione sull'istanza database di scrittura e l'ultima transazione applicata su un'istanza database di lettura.		Secondi

Parametro	Descrizione	Si applica a	Unità
`ReplicationChannelLag`	Per le configurazioni di replica multisorgente, il periodo di ritardo di un determinato canale sulla replica multisorgente rispetto all'istanza DB di origine. Per ulteriori informazioni, consulta the section called “Monitoraggio dei canali di replica da più fonti”.	MySQL	Secondi
`ReplicationSlotDiskUsage`	Lo spazio su disco utilizzato dai file degli slot di replica.	PostgreSQL	Byte
`SwapUsage`	La quantità di spazio di swapping utilizzato sull'istanza database.	MariaDB MySQL Oracle PostgreSQL	Byte
`TransactionLogsDiskUsage`	Lo spazio su disco utilizzato dai registri delle transazioni.	PostgreSQL	Byte
`TransactionLogsGeneration`	Le dimensioni dei registri delle transazioni generati al secondo.	PostgreSQL	Byte al secondo
`WriteIOPS`	Il numero medio di operazioni di I/O di scrittura su disco al secondo.	Tutti	Conteggio al secondo

Parametro	Descrizione	Si applica a	Unità
`WriteIOPS LocalStorage`	Il numero medio di operazioni di I/O di scrittura su disco al secondo su archiviazione locale. Questa metrica si applica solo alle classi di istanze database con volumi di archivio dell'istanza NVMe SSD. Per informazioni sulle istanze Amazon EC2 con volumi di archivio dell'istanza NVMe SSD, consulta Volumi dell'archivio dell'istanza. Le classi di istanze database di RDS equivalenti hanno gli stessi volumi di archivio dell'istanza. Ad esempio, le classi di istanze database db.m6gd e db.r6gd hanno volumi di archivio dell'istanza NVMe SSD.		Conteggio al secondo
`WriteIOPS LogVolume`	Il numero medio di operazioni di I/O di scrittura su disco al secondo per il volume di log.	Tutti	Conteggio al secondo
`WriteLatency`	La quantità di tempo media che occorre per ciascuna operazione I/O su disco.	Tutti	Secondi

Parametro	Descrizione	Si applica a	Unità
`WriteLatencyLocalStorage`	La quantità di tempo media che occorre per ciascuna operazione I/O su disco su archiviazione locale. Questa metrica si applica solo alle classi di istanze database con volumi di archivio dell'istanza NVMe SSD. Per informazioni sulle istanze Amazon EC2 con volumi di archivio dell'istanza NVMe SSD, consulta Volumi dell'archivio dell'istanza. Le classi di istanze database di RDS equivalenti hanno gli stessi volumi di archivio dell'istanza. Ad esempio, le classi di istanze database db.m6gd e db.r6gd hanno volumi di archivio dell'istanza NVMe SSD.		Secondi
`WriteLatencyLogVolume`	La quantità di tempo media che occorre per ciascuna operazione I/O su disco per il volume di log.	Tutti	Secondi
`WriteThroughput`	Il numero medio di byte scritti sul disco al secondo.	Tutti	Byte al secondo
`WriteThroughputLogVolume`	Il numero medio di byte scritti sul disco al secondo per il volume di log.	Tutti	Byte al secondo

Parametro	Descrizione	Si applica a	Unità
`WriteThroughputLocalStorage`	Il numero medio di byte scritti sul disco al secondo per l'archiviazione locale. Questa metrica si applica solo alle classi di istanze database con volumi di archivio dell'istanza NVMe SSD. Per informazioni sulle istanze Amazon EC2 con volumi di archivio dell'istanza NVMe SSD, consulta Volumi dell'archivio dell'istanza. Le classi di istanze database di RDS equivalenti hanno gli stessi volumi di archivio dell'istanza. Ad esempio, le classi di istanze database db.m6gd e db.r6gd hanno volumi di archivio dell'istanza NVMe SSD.		Byte al secondo

Il `AWS/Usage` namespace in Amazon CloudWatch include parametri di utilizzo a livello di account per le quote dei servizi Amazon RDS. CloudWatch raccoglie automaticamente i parametri di utilizzo per tutti. Regioni AWS

Per ulteriori informazioni, consulta i parametri di CloudWatch utilizzo nella Amazon CloudWatch User Guide. Per ulteriori informazioni sulle quote, consulta Quote e vincoli per Amazon RDS e Requesting a quota increase nella Guida per l'utente di Service Quotas.

Parametro	Descrizione	Unità*
`AllocatedStorage`	Totale archiviazione per tutte le istanze database Dal totale sono escluse le istanze di migrazione temporanea.	Gigabyte
`DBClusterParameterGroups`	Il numero di gruppi di parametri del cluster di database nel tuo Account AWS. I gruppi di parametri di default non vengono conteggiati.	Conteggio

Parametro	Descrizione	Unità*
DBClusters	Il numero di cluster di database Amazon Aurora nel tuo Account AWS.	Conteggio
DBInstances	Il numero di istanze database nel tuo Account AWS.	Conteggio
DBParameterGroups	Il numero di gruppi di parametri database nel tuo Account AWS. I gruppi di parametri database di default non vengono conteggiati.	Conteggio
DBSecurityGroups	Il numero di gruppi di sicurezza nel tuo Account AWS. Il gruppo di sicurezza di default e quello VPC di default non vengono conteggiati.	Conteggio
DBSubnetGroups	Il numero di gruppi di sottoreti nel tuo Account AWS. Il gruppo di sottoreti predefinito non viene conteggiato.	Conteggio
ManualClusterSnapshots	Il numero di snapshot del cluster di database creati manualmente nel tuo Account AWS. Gli snapshot non validi non vengono conteggiati.	Conteggio
ManualSnapshots	Il numero di snapshot database creati manualmente nel tuo Account AWS. Gli snapshot non validi non vengono conteggiati.	Conteggio
OptionGroups	Il numero di gruppi di opzioni nel tuo Account AWS. I gruppi di opzioni di default non vengono conteggiati.	Conteggio
ReservedDBInstances	Il numero di istanze database riservate nel tuo Account AWS. Le istanze ritirate o rifiutate non vengono conteggiate.	Conteggio

> **Note**
>
> Amazon RDS non pubblica unità per le metriche di utilizzo. CloudWatch Le unità sono presenti solo nella documentazione.

Le dimensioni di Amazon CloudWatch per Amazon RDS

Puoi filtrare i dati dei parametri di Amazon RDS utilizzando qualsiasi dimensione riportata nella seguente tabella.

Dimensione	Filtra i dati richiesti per...
`DBInstanceIdentifier`	Un'istanza database specifica.
`DatabaseClass`	Tutte le istanze di una classe di database. Ad esempio, puoi aggregare i parametri per tutte le istanze che appartengono alla classe database `db.r5.large` .
`EngineName`	Solo il nome del motore identificato. Ad esempio, puoi aggregare i parametri per tutte le istanze con nome del motore `postgres`.
`SourceRegion`	Usa solo la regione specificata. Ad esempio, puoi aggregare i parametri per tutte le istanze database nella regione `us-east-1` .

CloudWatch Metriche Amazon per Performance Insights

Performance Insights pubblica automaticamente alcune metriche su Amazon. CloudWatch Gli stessi dati possono essere interrogati da Performance Insights, ma l'inserimento delle metriche CloudWatch semplifica l'aggiunta di allarmi. CloudWatch Inoltre, semplifica l'aggiunta delle metriche alle dashboard esistenti. CloudWatch

Parametro	Descrizione
DBLoad	Il numero di sessioni attive per il motore del database. Generalmente, si richiedono i dati per il numero medio di sessioni attive. In Performance Insights, questi dati sono oggetto di query come `db.load.avg` .
DBLoadCPU	Il numero di sessioni attive in cui il tipo evento di attesa è CPU. In Performance Insights, questi dati sono oggetto di query come

Parametro	Descrizione
	`db.load.avg` , filtrate per tipo di evento di attesa `CPU`.
CPU DB LoadNon	Il numero di sessioni attive in cui il tipo evento di attesa non è CPU.

> **Note**
>
> Queste metriche vengono pubblicate CloudWatch solo in caso di carico sull'istanza DB.

Puoi esaminare queste metriche utilizzando la CloudWatch console AWS CLI, l'o l' CloudWatch API. Puoi anche esaminare altre metriche dei contatori di Performance Insights utilizzando una speciale funzione matematica metrica. Per ulteriori informazioni, consulta Interrogazione di altre metriche dei contatori di Performance Insights in CloudWatch.

Ad esempio, è possibile ottenere le statistiche per la `DBLoad` metrica eseguendo il comando. get-metric-statistics

```
aws cloudwatch get-metric-statistics \
    --region us-west-2 \
    --namespace AWS/RDS \
    --metric-name DBLoad  \
    --period 60 \
    --statistics Average \
    --start-time 1532035185 \
    --end-time 1532036185 \
    --dimensions Name=DBInstanceIdentifier,Value=db-loadtest-0
```

Questo esempio genera un output simile a quello riportato di seguito.

```
{
  "Datapoints": [
  {
  "Timestamp": "2021-07-19T21:30:00Z",
  "Unit": "None",
  "Average": 2.1
  },
```

```
{
"Timestamp": "2021-07-19T21:34:00Z",
"Unit": "None",
"Average": 1.7
},
{
"Timestamp": "2021-07-19T21:35:00Z",
"Unit": "None",
"Average": 2.8
},
{
"Timestamp": "2021-07-19T21:31:00Z",
"Unit": "None",
"Average": 1.5
},
{
"Timestamp": "2021-07-19T21:32:00Z",
"Unit": "None",
"Average": 1.8
},
{
"Timestamp": "2021-07-19T21:29:00Z",
"Unit": "None",
"Average": 3.0
},
{
"Timestamp": "2021-07-19T21:33:00Z",
"Unit": "None",
"Average": 2.4
}
],
"Label": "DBLoad"
}
```


Per ulteriori informazioni su CloudWatch, consulta What is Amazon CloudWatch? nella Amazon CloudWatch User Guide.

Interrogazione di altre metriche dei contatori di Performance Insights in CloudWatch

È possibile eseguire interrogazioni, avvisi e creare grafici sulle metriche di RDS Performance Insights da. CloudWatch È possibile accedere alle informazioni sull'istanza DB utilizzando la funzione matematica `DB_PERF_INSIGHTS` metrica for. CloudWatch Questa funzione consente di utilizzare le

metriche di Performance Insights che non vengono riportate direttamente per CloudWatch creare una nuova serie temporale.

È possibile utilizzare la nuova funzione Metric Math facendo clic sul menu a discesa Aggiungi matematica nella schermata Seleziona metrica nella console. CloudWatch Puoi usarlo per creare allarmi e grafici sulle metriche di Performance Insights o su combinazioni di metriche di Performance CloudWatch Insights, inclusi allarmi ad alta risoluzione per metriche inferiori al minuto. È inoltre possibile utilizzare la funzione a livello di codice includendo l'espressione Metric Math in una richiesta. `get-metric-data` Per ulteriori informazioni, vedere Sintassi e funzioni matematiche delle metriche e Creare un allarme sulle metriche dei contatori di Performance Insights da un database. AWS

Parametri contatore di Performance Insights

I parametri contatore sono parametri prestazionali di sistema operativo e database nel pannello di controllo di Performance Insights. Per agevolare l'individuazione e l'analisi di problemi legati alle prestazioni, è possibile correlare i parametri contatore ai carichi dei database. Puoi aggiungere una funzione statistica alla metrica per ottenere i valori delle metriche. Ad esempio, le funzioni supportate per la metrica `os.memory.active` sono `.avg`, `.min`, `.max`, `.sum` e `.sample_count`.

Le metriche dei contatori vengono raccolte una volta al minuto. La raccolta delle metriche del sistema operativo dipende dall'attivazione o dalla disattivazione della funzionalità Monitoraggio avanzato. Se la funzionalità Monitoraggio avanzato è disattivata, le metriche del sistema operativo vengono raccolte una volta al minuto. Se la funzionalità Monitoraggio avanzato è attivata, le metriche del sistema operativo vengono raccolte per il periodo di tempo selezionato. Per ulteriori informazioni sull'attivazione o sulla disattivazione della funzionalità Monitoraggio avanzato, consulta Attivazione e disattivazione del monitoraggio avanzato.

Argomenti

- Contatori del sistema operativo in Performance Insights
- Contatori Performance Insights per Amazon RDS for MariaDB e MySQL
- Contatori Performance Insights per Amazon RDS for Microsoft SQL Server
- Contatori Performance Insights per Amazon RDS for Oracle
- Contatori Performance Insights per Amazon RDS for PostgreSQL

Contatori del sistema operativo in Performance Insights

I seguenti contatori del sistema operativo, con prefisso `os`, sono disponibili in Approfondimenti sulle prestazioni per tutti i motori RDS tranne RDS per SQL Server .

Puoi utilizzare l'API `ListAvailableResourceMetrics` per l'elenco delle metriche dei contatori disponibili per l'istanza database. Per ulteriori informazioni, consulta la guida ListAvailableResourceMetricsdi riferimento dell'API Amazon RDS Performance Insights.

Contatore	Tipo	Parametro	Descrizione
Attivo	Memoria	os.memory.active	La quantità di memoria assegnata, in kilobyte.
Buffer	Memoria	os.memory.buffers	La quantità di memoria utilizzata per il buffering delle richieste di I/O prima della scrittura sul dispositivo di storage, in kilobyte.
Cached	Memoria	os.memory.cached	La quantità di memoria utilizzata per la memorizzazione nella cache dell'I/O basato sul file system, in kilobyte.
DB Cache	Memoria	os.memory.db.cache	La quantità di memoria utilizzata per la cache della pagina in base al processo del database, incluso tmpfs (shmem), in byte.

Contatore	Tipo	Parametro	Descrizione
DB Resident Set Size	Memoria	os.memory.db.residentSetSize	La quantità di memoria utilizzata per la cache anonima e di swap in base al processo del database, escluso tmpfs (shmem), in byte.
DB Swap	Memoria	os.memory.db.swap	La quantità di memoria utilizzata per lo scambio dal processo del database, in byte.
Dirty	Memoria	os.memory.dirty	La quantità di pagine di memoria nella RAM che sono state modificate ma non scritte nel relativo blocco di dati nello storage, in kilobyte.
Gratuito	Memoria	os.memory.free	La quantità di memoria non assegnata, in kilobyte.
Huge Pages libere	Memoria	os.memoria.hugePagesFree	Il numero di pagine di grandi dimensioni gratuite. Le pagine di grandi dimensioni sono una caratteristica del kernel di Linux.

Contatore	Tipo	Parametro	Descrizione
Huge Pages Rsvd	Memoria	os.memoria.hugePagesRsvd	Il numero di pagine di grandi dimensioni impegnate.
Dimensioni Huge Pages	Memoria	os.memoria.hugePagesSize	La dimensione per ogni unità delle pagine di grandi dimensioni, in kilobyte.
Huge Pages Surp	Memoria	os.memoria.hugePagesSurp	Il numero di pagine di grandi dimensioni in eccesso disponibili sul totale.
Totale Huge Pages	Memoria	os.memoria.hugePagesTotal	Il numero totale di Huge Pages.
Inattivo	Memoria	os.memory.inactive	La quantità di pagine di memoria utilizzat e meno frequente mente, in kilobyte.
Mapped	Memoria	os.memory.mapped	La quantità totale di contenuti del file system mappati in memoria all'interno di uno spazio di indirizza mento del processo, in kilobyte.
Out of Memory Kill Count	Memoria	os.memoria.outOfMemoryKillCount	Il numero di interruzi oni OOM avvenute nell'ultimo intervallo di raccolta.

Contatore	Tipo	Parametro	Descrizione
Tabelle delle pagine	Memoria	os.memory.pageTables	La quantità di memoria utilizzata dalle tabelle della pagina, in kilobyte.
Slab	Memoria	os.memory.slab	La quantità di strutture dati del kernel riutilizzabili, in kilobyte.
Totale	Memoria	os.memory.total	La quantità totale di memoria, in kilobyte.
Writeback	Memoria	os.memory.writeback	La quantità di pagine sporche nella RAM che sono ancora scritte nello storage di backup, in kilobyte.
Guest	Utilizzo CPU	os.cpuUtilization.guest	La percentuale di CPU utilizzata dai programmi guest.
Idle	Utilizzo CPU	os.cpuUtilization.idle	La percentuale di tempo CPU che è inattiva.
Irq	Utilizzo CPU	os.cpuUtilization.irq	La percentuale di CPU utilizzata dalle interruzioni dei software.
Nice	Utilizzo CPU	os.cpuUtilization.nice	La percentuale di CPU utilizzata dai programmi in esecuzione con priorità più bassa.

Contatore	Tipo	Parametro	Descrizione
Steal	Utilizzo CPU	os.cpuUtilization.steal	La percentuale di CPU utilizzata da altre macchine virtuali.
System (Sistema)	Utilizzo CPU	os.cpuUtilization. system	La percentuale di CPU utilizzata dal kernel.
Totale	Utilizzo CPU	os.cpuUtilization.total	La percentuale totale del CPU utilizzata. Questo valore include il valore nice.
Utente	Utilizzo CPU	os.cpuUtilization.user	La percentuale di CPU utilizzata dai programmi utente.
Attendi	Utilizzo CPU	os.cpuUtilization.wait	La percentuale di CPU non utilizzata durante l'attesa per l'accesso I/O.
Read IOs PS	I/O del disco	os.diskIO.<nomedispositivo>.readIOsPS	Il numero di operazioni di lettura al secondo.
Write IOs PS	I/O del disco	os.diskIO.<nomedispositivo>.writeIOsPS	Il numero di operazioni di scrittura al secondo.
Avg Queue Len	I/O del disco	Sistema operativo DiskIO. <devicename>. avgQueueLen	Il numero di richieste in attesa nella coda del dispositivo I/O.

Contatore	Tipo	Parametro	Descrizione
Avg Req Sz	I/O del disco	Sistema operativo DiskIO. <devicename>. avgReqSz	Il numero di richieste in attesa nella coda del dispositivo I/O.
Await	I/O del disco	os.diskIO.<nomedispositivo>.await	Il numero di millisecondi necessari per rispondere alle richieste, compreso il tempo della coda e il tempo del servizio.
Read IOs PS	I/O del disco	os.diskIO.<nomedispositivo>.readIOsPS	Il numero di operazioni di lettura al secondo.
Read KB	I/O del disco	os.diskIO.<nomedispositivo>.readKb	Il numero totale di kilobyte letti.
Read KB PS	I/O del disco	os.diskIO.<nomedispositivo>.readKbPS	Il numero di kilobytes letti al secondo.
Rram PS	I/O del disco	os.diskIO.<nomedispositivo>.rrqmPS	Il numero di richieste di lettura unite in coda al secondo.
TPS	I/O del disco	os.diskIO.<nomedispositivo>.tps	Il numero di transazioni I/O al secondo.
Util	I/O del disco	os.diskIO.<nomedispositivo>.util	La percentuale di tempo della CPU durante il quale sono state emesse le richieste.

Contatore	Tipo	Parametro	Descrizione
Write KB	I/O del disco	os.diskIO.<nomedispositivo>.writeKb	Il numero totale di kilobyte scritti.
Write KB PS	I/O del disco	os.diskIO.<nomedispositivo>.writeKbPS	Il numero di kilobytes scritti al secondo.
Wrqm PS	I/O del disco	os.diskIO.<nomedispositivo>.wrqmPS	Il numero di richieste di scrittura unite in coda al secondo.
Bloccato	Attività	os.tasks.blocked	Il numero di attività che sono bloccate.
In esecuzione	Attività	os.tasks.running	Il numero di attività che sono in esecuzione.
Sleeping	Attività	os.tasks.sleeping	Il numero di attività che sono a riposo.
Arrestato	Attività	os.tasks.stopped	Il numero di attività che sono arrestate.
Totale	Attività	os.tasks.total	Il numero totale di attività.
Zombie	Attività	os.tasks.zombie	Il numero di attività secondarie che sono inattive con un'attività genitore attiva.
One	Media carico al minuto	os. loadAverageMinute.uno	Il numero di processi che richiedono l'ora della CPU nell'ultimo minuto.

Contatore	Tipo	Parametro	Descrizione
Fifteen	Media carico al minuto	così. loadAvera geMinute.quindici	Il numero di processi che richiedono l'ora della CPU negli ultimi 15 minuti.
Cinque	Media carico al minuto	così. loadAvera geMinute.cinque	Il numero di processi che richiedono l'ora della CPU negli ultimi 5 minuti.
Cached	Swap	os.swap.cached	La quantità di memoria di scambio, in kilobyte, utilizzata come memoria cache.
Gratuito	Swap	os.swap.free	La quantità di memoria di scambio libera, in kilobyte.
In	Swap	os.swap.in	Quantità di memoria, in kilobyte, scambiata in ingresso nel disco.
Out	Swap	os.swap.out	Quantità di memoria, in kilobyte, scambiata in uscita dal disco.
Totale	Swap	os.swap.total	La quantità totale di memoria di scambio disponibile, in kilobyte.
Max Files	File system	os.fileSys.maxFiles	Il numero massimo di file che è possibile creare per il sistema di file.

Contatore	Tipo	Parametro	Descrizione
Used Files	File system	os.fileSys.usedFiles	Il numero di file audio nel sistema di file.
Used File Percent	File system	sistema operativo. Filesys. usedFileP ercent	La percentuale di file disponibili in uso.
Used Percent	File system	os.fileSys.usedPer cent	La percentuale dello spazio su disco del sistema di file in uso.
Used	File system	os.fileSys.used	La quantità totale di spazio su disco utilizzato dai file nel sistema di file, in kilobyte.
Totale	File system	os.fileSys.total	Il numero totale di spazio su disco disponibile per il sistema di file, in kilobyte.
Rx	Rete	os.network.rx	Il numero di bytes ricevuti al secondo.
Tx	Rete	os.network.tx	Il numero di bytes caricati al secondo.
Acu Utilization	Generali	os.general.acuUtil ization	La percentuale di capacità attuale rispetto alla capacità massima configurata.
Max Configured Acu	Generali	sistema operativo generale. maxConfig uredAcu	La capacità massima configurata dall'uten te, in ACU.

Contatore	Tipo	Parametro	Descrizione
Min Configured Acu	Generali	sistema operativo generale. minConfig uredAcu	La capacità minima configurata dall'uten te, in ACU.
Num VCPUs	Generali	os.general.numVCPU s	Il numero di CPU virtuali per l'istanza database.
Serverless Database Capacity	Generali	sistema operativo generale. serverles sDatabaseCapacity	La capacità attuale, in ACU, dell'istanza.

Contatori Performance Insights per Amazon RDS for MariaDB e MySQL

I seguenti contatori del database sono disponibili in Performance Insights per Amazon RDS for MariaDB and MySQL.

Argomenti

- Contatori nativi per RDS for MariaDB e RDS for MySQL
- Contatori non nativi per Amazon RDS for MariaDB e MySQL

Contatori nativi per RDS for MariaDB e RDS for MySQL

I parametri nativi sono definiti dal motore del database e non da Amazon RDS. Per le definizioni di questi parametri nativi, consulta Variabili dello stato del server nella documentazione di MySQL.

Contatore	Tipo	Unità	Parametro
Com_analyze	SQL	Query al secondo	db.SQL.Com_analyze
Com_optimize	SQL	Query al secondo	db.SQL.Com_optimize

Contatore	Tipo	Unità	Parametro
Com_select	SQL	Query al secondo	db.SQL.Com_select
Connessioni	SQL	Il numero di tentativi di connessione al minuto (riusciti o meno) al server MySQL	db.Users.Connections
Innodb_rows_deleted	SQL	Righe al secondo	db.SQL.Innodb_rows_deleted
Innodb_rows_inserted	SQL	Righe al secondo	db.SQL.Innodb_rows_inserted
Innodb_rows_read	SQL	Righe al secondo	db.SQL.Innodb_rows_read
Innodb_rows_updated	SQL	Righe al secondo	db.SQL.Innodb_rows_updated
Select_full_join	SQL	Query al secondo	db.SQL.Select_full_join
Select_full_range_join	SQL	Query al secondo	db.SQL.Select_full_range_join
Select_range	SQL	Query al secondo	db.SQL.Select_range
Select_range_check	SQL	Query al secondo	db.SQL.Select_range_check

Contatore	Tipo	Unità	Parametro
Select_scan	SQL	Query al secondo	db.SQL.Select_scan
Slow_queries	SQL	Query al secondo	db.SQL.Slow_queries
Sort_merge_passes	SQL	Query al secondo	db.SQL.Sort_merge_passes
Sort_range	SQL	Query al secondo	db.SQL.Sort_range
Sort_rows	SQL	Query al secondo	db.SQL.Sort_rows
Sort_scan	SQL	Query al secondo	db.SQL.Sort_scan
Questions	SQL	Query al secondo	db.SQL.Questions
Innodb_row_lock_time	Locks	Millisecondi (media)	db.Locks.Innodb_row_lock_time
Table_locks_immediate	Locks	Richieste al secondo	db.Locks.Table_locks_immediate
Table_locks_waited	Locks	Richieste al secondo	db.Locks.Table_locks_waited
Aborted_clients	Utenti	Connessioni	db.Users.Aborted_clients
Aborted_connects	Utenti	Connessioni	db.Users.Aborted_connects
max_connections	Utenti	Connessioni	db.User.max_connections
Threads_created	Utenti	Connessioni	db.Users.Threads_created

Contatore	Tipo	Unità	Parametro
Threads_running	Utenti	Connessioni	db.Users.Threads_running
Innodb_data_writes	I/O	Operazioni al secondo	db.IO.Innodb_data_writes
Innodb_dblwr_writes	I/O	Operazioni al secondo	db.IO.Innodb_dblwr_writes
Innodb_log_write_requests	I/O	Operazioni al secondo	db.IO.Innodb_log_write_requests
Innodb_log_writes	I/O	Operazioni al secondo	db.IO.Innodb_log_writes
Innodb_pages_written	I/O	Pagine al secondo	db.IO.Innodb_pages_written
Created_tmp_disk_tables	Temp	Tabelle al secondo	db.Temp.Created_tmp_disk_tables
Created_tmp_tables	Temp	Tabelle al secondo	db.Temp.Created_tmp_tables
Innodb_buffer_pool_pages_data	Cache	Pagine	db.Cache.Innodb_buffer_pool_pages_data
Innodb_buffer_pool_pages_total	Cache	Pagine	db.Cache.Innodb_buffer_pool_pages_total
Innodb_buffer_pool_read_requests	Cache	Pagine al secondo	db.Cache.Innodb_buffer_pool_read_requests
Innodb_buffer_pool_reads	Cache	Pagine al secondo	db.Cache.Innodb_buffer_pool_reads
Opened_tables	Cache	Tabelle	db.Cache.Opened_tables

Contatore	Tipo	Unità	Parametro
Opened_table_definitions	Cache	Tabelle	db.Cache.Opened_table_definitions
Qcache_hits	Cache	Query	db.Cache.Qcache_hits

Contatori non nativi per Amazon RDS for MariaDB e MySQL

I parametri contatore non nativi sono contatori definiti da Amazon RDS. Un parametro non nativo può essere un parametro che si ottiene con una query specifica. Un parametro non nativo può essere un parametro derivato, dove vengono utilizzati due o più contatori nativi nei calcoli di rapporti, percentuali di riscontri o latenze.

Contatore	Tipo	Parametro	Descrizione	Definizione
innodb_buffer_pool_hits	Cache	db.Cache.innoDB_buffer_pool_hits	Il numero di letture che InnoDB potrebbe soddisfare dal pool di buffer.	`innodb_buffer_pool_read_requests - innodb_buffer_pool_reads`
innodb_buffer_pool_hit_rate	Cache	db.Cache.innoDB_buffer_pool_hit_rate	La percentuale di letture che InnoDB potrebbe soddisfare dal pool di buffer.	`100 * innodb_buffer_pool_read_requests / (innodb_buffer_pool_read_requests + innodb_buffer_pool_reads)`

Contatore	Tipo	Parametro	Descrizione	Definizione
innodb_buffer_pool_usage	Cache	db.Cache.innoDB_buffer_pool_usage	La percentuale del pool di buffer di InnoDB che contiene dati (pagine). **Note** Quando si utilizzano tabelle compres, questo valore può variare. Per ulteriori dettagli, consulta le informazioni su `Innodb ffer_p _pages ta`	`Innodb_buffer_pool_pages_data / Innodb_buffer_pool_pages_total * 100.0`

Contatore	Tipo	Parametro	Descrizione	Definizione
			e `Innodb_buffer_pool_pages_total` in Variabili dello stato del server nella documentazione di MySQL.	
query_cache_hit_rate	Cache	db.Cache.query_cache_hit_rate	La percentuale di riscontri della cache (cache query) del set di risultati MySQL.	`Qcache_hits / (QCache_hits + Com_select) * 100`

Contatore	Tipo	Parametro	Descrizione	Definizione
innodb_datafile_writes_to_disk	I/O	db.IO.innoDB_datafile_writes_to_disk	Il numero di scritture del file di dati di InnoDB su disco, escluse le operazioni di doppia scrittura e scrittura di registrazione ripetuta.	`Innodb_data_writes - Innodb_log_writes - Innodb_dblwr_writes`
innodb_rows_changed	SQL	db.SQL.innodb_rows_changed	Il totale delle operazioni delle righe di InnoDB.	`db.SQL.Innodb_rows_inserted + db.SQL.Innodb_rows_deleted + db.SQL.Innodb_rows_updated`
active_transactions	Transazioni	db.Transactions.active_transactions	Le transazioni attive totali.	`SELECT COUNT(1) AS active_transactions FROM INFORMATION_SCHEMA.INNODB_TRX`

Contatore	Tipo	Parametro	Descrizione	Definizione
trx_rseg_history_len	Transazioni	db.Transactions.trx_rseg_history_len	L'elenco delle pagine di log degli annullamenti per le transazioni confermate che viene gestito dal sistema di transazioni InnoDB per implementare il controllo della concorrenza tra più versioni. Per ulteriori informazioni sui dettagli dei record dei log degli annullamenti, consulta https://dev.mysql.com/doc/refman/8.0/en/innodb-multi-versioning.html nella	`SELECT COUNT AS trx_rseg_history_len FROM INFORMATION_SCHEMA.INNODB_METRICS WHERE NAME='trx_rseg_history_len'`

Contatore	Tipo	Parametro	Descrizione	Definizione
			documentazione di MySQL.	
innodb_deadlocks	Locks	db.Locks.innodb_deadlocks	Il numero totale di deadlock.	`SELECT COUNT AS innodb_deadlocks FROM INFORMATION_SCHEMA.INNODB_METRICS WHERE NAME='lock_deadlocks'`
innodb_lock_timeouts	Locks	db.Locks.innodb_lock_timeouts	Il numero totale di blocchi scaduti.	`SELECT COUNT AS innodb_lock_timeouts FROM INFORMATION_SCHEMA.INNODB_METRICS WHERE NAME='lock_timeouts'`
innodb_row_lock_waits	Locks	db.Locks.innodb_row_lock_waits	Il numero totale di blocchi alle righe che ha determinato un'attesa.	`SELECT COUNT AS innodb_row_lock_waits FROM INFORMATION_SCHEMA.INNODB_METRICS WHERE NAME='lock_row_lock_waits'`

Contatori Performance Insights per Amazon RDS for Microsoft SQL Server

I seguenti contatori del database sono disponibili in Performance Insights per RDS per Microsoft SQL Server.

Contatori nativi per RDS per Microsoft SQL Server

I parametri nativi sono definiti dal motore del database e non da Amazon RDS. È possibile trovare le definizioni per questi parametri nativi in Utilizzare oggetti di SQL Server nella documentazione di Microsoft SQL Server.

Contatore	Tipo	Unità	Parametro
Record inoltrati	Metodi di accesso	Record al secondo	db.Access Methods.Forwarded Records
Suddivisioni di pagina	Metodi di accesso	Suddivisioni al secondo	db.Access Methods.Page Splits
Percentuale riscontri cache buffer	Buffer Manager	Ratio	db.Buffer Manager.Buffer cache hit ratio
Permanenza presunta delle pagine	Buffer Manager	Permanenza presunta in secondi	db.Buffer Manager.Page life expectancy
Ricerche di pagina	Buffer Manager	Ricerche al secondo	db.Buffer Manager.Page lookups
Letture di pagina	Buffer Manager	Letture al secondo	db.Buffer Manager.Page reads
Scritture di pagina	Buffer Manager	Scritture al secondo	db.Buffer Manager.Page writes
Transazioni attive	Database	Transazioni	db.Databases.Active Transactions (_Total)
Byte di log scaricati	Database	Byte scaricati al secondo	db.Databases.Log Bytes Flushed (_Total)

Contatore	Tipo	Unità	Parametro
Attese scarico log	Database	Attese al secondo	db.Databases.Log Flush Waits (_Total)
Svuotamenti log	Database	Svuotamenti al secondo	db.Databases.Log Flushes (_Total)
Transazioni di scrittura	Database	Transazioni al secondo	db.Databases.Write Transactions (_Total)
Processi bloccati	Statistiche generali	Processi bloccati	db.General Statistic s.Processes blocked
Connessioni utente	Statistiche generali	Connessioni	db.General Statistic s.User Connections
Attese latch	Latch	Attese al secondo	db.Latches.Latch Waits
Numero di deadlock.	Locks	Deadlock al secondo	db.Locks.Number of Deadlocks (_Total)
Concessioni di memoria in sospeso	Memory Manager	Concessioni della memoria	db.Memory Manager.Memory Grants Pending
Richieste batch	Statistiche SQL	Richieste al secondo	db.SQL Statistic s.Batch Requests
Compilazioni SQL	Statistiche SQL	Compilazioni al secondo	db.SQL Statistics.SQL Compilations
Ricompilazioni SQL	Statistiche SQL	Ricompilazioni al secondo	db.SQL Statistics.SQL Re-Compilations

Contatori Performance Insights per Amazon RDS for Oracle

I seguenti contatori del database sono disponibili in Performance Insights per RDS for Oracle.

Contatori nativi per RDS for Oracle

I parametri nativi sono definiti dal motore del database e non da Amazon RDS. Le definizioni di questi parametri nativi sono riportate in Statistics Descriptions nella documentazione di Oracle.

> **Note**
>
> Per il parametro contatore `CPU used by this session`, l'unità è stata trasformata da centisecondi nativi a sessioni attive per agevolare l'utilizzo del valore. Ad esempio, CPU send nel grafico del carico del database rappresenta la domanda di CPU. Il parametro contatore `CPU used by this session` rappresenta la quantità di CPU utilizzata dalle sessioni Oracle. Puoi confrontare CPU send e il parametro contatore `CPU used by this session`. Quando la domanda di CPU è superiore alla CPU utilizzata, le sessioni sono in attesa di tempo CPU.

Contatore	Tipo	Unità	Parametro
CPU used by this session	Utente	Sessioni attive	db.User.CPU used by this session
SQL*Net roundtrips to/from client	Utente	Round trip al secondo	db.User.SQL*Net roundtrips to/from client
Bytes received via SQL*Net from client	Utente	Byte al secondo	db.User.bytes received via SQL*Net from client
User commits	Utente	Commit al secondo	db.User.user commits
Logons cumulative	Utente	Accessi al secondo	db.User.logons cumulative
User calls	Utente	Chiamate al secondo	Chiamate db.User.user
Bytes sent via SQL*Net to client	Utente	Byte al secondo	db.User.bytes sent via SQL*Net to client

Contatore	Tipo	Unità	Parametro
User rollbacks	Utente	Rollback al secondo	Rollback db.User.user
Redo size	Redo	Byte al secondo	Dimensioni db.Redo.redo
Parse count (total)	SQL	Analisi al secondo	db.SQL.parse count (total)
Parse count (hard)	SQL	Analisi al secondo	db.SQL.parse count (hard)
Table scan rows gotten	SQL	Righe al secondo	db.SQL.table scan rows gotten
Sorts (memory)	SQL	Ordinamenti al secondo	db.SQL.sorts (memory)
Sorts (disk)	SQL	Ordinamenti al secondo	db.SQL.sorts (disk)
Sorts (rows)	SQL	Ordinamenti al secondo	db.SQL.sorts (rows)
Physical read bytes	Cache	Byte al secondo	db.Cache.physical read bytes
DB block gets	Cache	Blocchi al secondo	db.Cache.db block gets
DBWR checkpoints	Cache	Checkpoint al minuto	db.Cache.DBWR checkpoints
Physical reads	Cache	Letture al secondo	db.Cache.physical reads
Consistent gets from cache	Cache	Recuperi al secondo	db.Cache.consistent gets from cache

Contatore	Tipo	Unità	Parametro
DB block gets from cache	Cache	Recuperi al secondo	db.Cache.db block gets from cache
Consistent gets	Cache	Recuperi al secondo	db.Cache.consistent gets

Contatori Performance Insights per Amazon RDS for PostgreSQL

I seguenti contatori del database sono disponibili in Performance Insights per Amazon RDS for PostgreSQL.

Argomenti

- Contatori nativi per Amazon RDS for PostgreSQL
- Contatori non nativi per Amazon RDS for PostgreSQL

Contatori nativi per Amazon RDS for PostgreSQL

I parametri nativi sono definiti dal motore del database e non da Amazon RDS. Le definizioni di questi parametri nativi sono riportate in Viewing Statistics nella documentazione di PostgreSQL.

Contatore	Tipo	Unità	Parametro
blks_hit	Cache	Blocchi al secondo	db.Cache.blks_hit
buffers_alloc	Cache	Blocchi al secondo	db.Cache.buffers_alloc
buffers_checkpoint	Checkpoint	Blocchi al secondo	db.Checkpoint.buffers_checkpoint
checkpoint_sync_time	Checkpoint	Millisecondi per checkpoint	db.Checkpoint.checkpoint_sync_time
checkpoint_write_time	Checkpoint	Millisecondi per checkpoint	db.Checkpoint.checkpoint_write_time

Contatore	Tipo	Unità	Parametro
checkpoints_req	Checkpoint	Checkpoint al minuto	db.Checkpoint.checkpoints_req
checkpoints_timed	Checkpoint	Checkpoint al minuto	db.Checkpoint.checkpoints_timed
maxwritten_clean	Checkpoint	Interruzioni clean lettura in background al minuto	db.Checkpoint.maxwritten_clean
deadlocks	Concorrenza	Deadlock al minuto	db.Concurrency.deadlocks
blk_read_time	I/O	Millisecondi	db.IO.blk_read_time
blks_read	I/O	Blocchi al secondo	db.IO.blks_read
buffers_backend	I/O	Blocchi al secondo	db.IO.buffers_backend
buffers_backend_fsync	I/O	Blocchi al secondo	db.IO.buffers_backend_fsync
buffers_clean	I/O	Blocchi al secondo	db.IO.buffers_clean
tup_deleted	SQL	Tuple al secondo	db.SQL.tup_deleted
tup_fetched	SQL	Tuple al secondo	db.SQL.tup_fetched
tup_inserted	SQL	Tuple al secondo	db.SQL.tup_inserted
tup_returned	SQL	Tuple al secondo	db.SQL.tup_returned
tup_updated	SQL	Tuple al secondo	db.SQL.tup_updated
temp_bytes	Temp	Byte al secondo	db.Temp.temp_bytes
temp_files	Temp	File al minuto	db.Temp.temp_files

Contatore	Tipo	Unità	Parametro
xact_commit	Transazioni	Commit al secondo	db.Transactions.xact_commit
xact_rollback	Transazioni	Rollback al secondo	db.Transactions.xact_rollback
numbackends	Utente	Connessioni	db.User.numbackends
archived_count	Registro Write-ahead (WAL)	File al minuto	db.WAL.archived_count

Contatori non nativi per Amazon RDS for PostgreSQL

I parametri contatore non nativi sono contatori definiti da Amazon RDS. Un parametro non nativo può essere un parametro che si ottiene con una query specifica. Un parametro non nativo può essere un parametro derivato, dove vengono utilizzati due o più contatori nativi nei calcoli di rapporti, percentuali di riscontri o latenze.

Contatore	Tipo	Parametro	Descrizione	Definizione
checkpoint_sync_latency	Checkpoint	db.Checkpoint.checkpoint_sync_latency	La quantità totale di tempo impiegato nella parte dell'elaborazione dei checkpoint dove i file vengono sincronizzati sul disco.	`checkpoint_sync_time / (checkpoints_timed + checkpoints_req)`
checkpoint_write_latency	Checkpoint	db.Checkpoint.checkpoint_write_latency	La quantità totale di tempo impiegato nella parte dell'elaborazione dei checkpoint dove i	`checkpoint_write_time / (checkpoints_timed`

Contatore	Tipo	Parametro	Descrizione	Definizione
			file vengono scritti sul disco.	`+ checkpoints_req)`
read_latency	I/O	db.IO.read_latency	Il tempo impiegato per la lettura dei blocchi di file di dati dai back-end in questa istanza.	`blk_read_time / blks_read`
idle_in_transaction_aborted_count	Stato	db.state.idle_in_transaction_aborted_count	Il numero di sessioni nello stato. `idle in transaction (aborted)`	–
idle_in_transaction_count	Stato	db.state.idle_in_transaction_count	Il numero di sessioni nello stato. `idle in transaction`	–
idle_in_transaction_max_time	Stato	db.state.idle_in_transaction_max_time	La durata della transazione più lunga nello stato, in secondi. `idle in transaction`	–
active_transactions	Transazioni	db.Transactions.active_transactions	Il numero di transazioni attive.	–
blocked_transactions	Transazioni	db.Transactions.blocked_transactions	Il numero di transazioni bloccate.	–

Contatore	Tipo	Parametro	Descrizione	Definizione
max_used_xact_ids	Transa ni	db.Transactions.max_used_xact_ids	Il numero di transazioni che non sono state cancellate.	–
max_connections	Utenti	db.User.max_connections	Il numero massimo di connessioni consentite per un'istanza DB, come configurato nel `max_connections` parametro.	–
archive_failed_count	WAL	db.WAL.archive_failed_count	Il numero di tentativi falliti di archiviazione dei file WAL, in file al minuto.	–

Statistiche SQL per Performance Insights

Le statistiche SQL sono parametri relativi alle prestazioni delle query SQL raccolti da Performance Insights. Performance Insights raccoglie statistiche durante ogni secondo in cui è in esecuzione una query e per ogni chiamata SQL. Le statistiche SQL sono una media per l'intervallo di tempo selezionato.

Un SQL Digest è un composito di tutte le query con un determinato modello ma che non hanno necessariamente gli stessi valori letterali. Il digest sostituisce i valori letterali con un punto interrogativo. Ad esempio, `SELECT * FROM emp WHERE lname= ?`. Questo digest può essere costituito dalle seguenti query figlio:

```
SELECT * FROM emp WHERE lname = 'Sanchez'
SELECT * FROM emp WHERE lname = 'Olagappan'
SELECT * FROM emp WHERE lname = 'Wu'
```

Tutti i motori supportano le statistiche SQL delle query a livello di digest.

Per informazioni sull'assistenza alla regione, al motore di database e alla classe di istanza per questa funzionalità, consulta Supporto di classe di istanza, regione e motore di database Amazon RDS per funzionalità Performance Insights

Argomenti

- Statistiche SQL per MariaDB e MySQL
- Statistiche SQL per Oracle
- Statistiche SQL per SQL Server
- Statistiche SQL per RDS PostgreSQL

Statistiche SQL per MariaDB e MySQL

MariaDB e MySQL raccolgono le statistiche SQL solo a livello di digest. Nessuna statistica viene mostrata a livello di istruzione.

Argomenti

- Statistiche digest per MariaDB e MySQL
- Statistiche per seconod per MariaDB e MySQL
- Statistiche per chiamata per MariaDB e MySQL

Statistiche digest per MariaDB e MySQL

Performance Insights raccoglie statistiche digest SQL dalla tabella `events_statements_summary_by_digest`. La tabella `events_statements_summary_by_digest` è gestita dal database.

La tabella digest non dispone di una policy di espulsione. Il seguente messaggio viene visualizzato in AWS Management Console quando la tabella è piena:

```
Performance Insights is unable to collect SQL Digest statistics on new queries because
 the table events_statements_summary_by_digest is full.
Please truncate events_statements_summary_by_digest table to clear the issue. Check the
 User Guide for more details.
```

In questa situazione, MariaDB e MySQL non eseguono il tracciamento, esegue il tracciamento delle query SQL. Per risolvere questo problema, Performance Insights tronca automaticamente la tabella del digest quando sono soddisfatte entrambe le condizioni seguenti:

- La tabella è piena.
- Performance Insights gestisce automaticamente Performance Schema.

  Per la gestione automatica, `performance_schema` deve essere impostato su `0` e la Fonte non deve essere impostata su `user`. Se Performance Insights non gestisce automaticamente lo schemda delle prestazioni, vedi Abilitazione di Performance Schema per Performance Insights su Amazon RDS for MariaDB o MySQL.

Nella AWS CLI, controllare l'origine di un valore di parametro eseguendo il comando describe-db-parameters.

Statistiche per seconod per MariaDB e MySQL

Le seguenti statistiche SQL sono disponibili per istanze di database MariaDB e MySQL.

Parametro	Unità
db.sql_tokenized.stats.count_star_per_sec	Chiamate al secondo
db.sql_tokenized.stats.sum_timer_wait_per_sec	Media delle esecuzioni attive al secondo (AAE, Average active executions)
db.sql_tokenized.stats.sum_select_full_join_per_sec	Seleziona full join al secondo
db.sql_tokenized.stats.sum_select_range_check_per_sec	Seleziona controllo intervallo al secondo
db.sql_tokenized.stats.sum_select_scan_per_sec	Seleziona scansione al secondo
db.sql_tokenized.stats.sum_sort_merge_passes_per_sec	Ordina i pass di unione al secondo
db.sql_tokenized.stats.sum_sort_scan_per_sec	Ordina scansioni al secondo

Parametro	Unità
db.sql_tokenized.stats.sum_sort_rang e_per_sec	Ordina intervalli al secondo
db.sql_tokenized.stats.sum_sort_rows_per_sec	Ordina righe al secondo
db.sql_tokenized.stats.sum_rows_affected_per_ sec	Righe interessate al secondo
db.sql_tokenized.stats.sum_rows_exam ined_per_sec	Righe esaminate al secondo
db.sql_tokenized.stats.sum_rows_sent_per_sec	Righe inviate al secondo
db.sql_tokenized.stats.sum_created_tmp_disk_t ables_per_sec	Tabelle disco temporanee create al secondo
db.sql_tokenized.stats.sum_created_tmp_tables _per_sec	Tabelle temporanee create al secondo
db.sql_tokenized.stats.sum_lock_time_per_sec	Tempo di blocco al secondo (in ms)

Statistiche per chiamata per MariaDB e MySQL

I seguenti parametri forniscono le statistiche per chiamata di un'istruzione SQL.

Parametro	Unità
db.sql_tokenized.stats.sum_timer_wait_per_call	Latenza media per chiamata (in ms)
db.sql_tokenized.stats.sum_select_full_join_p er_call	Seleziona full join per chiamata
db.sql_tokenized.stats.sum_select_ra nge_check_per_call	Seleziona controllo intervallo per chiamata
db.sql_tokenized.stats.sum_select_sc an_per_call	Seleziona scansioni per chiamata

Parametro	Unità
db.sql_tokenized.stats.sum_sort_merge_passes_per_call	Ordina pass di unione per chiamata
db.sql_tokenized.stats.sum_sort_scan_per_call	Ordinare scansioni per chiamata
db.sql_tokenized.stats.sum_sort_range_per_call	Ordina intervalli per chiamata
db.sql_tokenized.stats.sum_sort_rows_per_call	Ordina righe per chiamata
db.sql_tokenized.stats.sum_rows_affected_per_call	Righe interessate per chiamata
db.sql_tokenized.stats.sum_rows_examined_per_call	Righe esaminate per chiamata
db.sql_tokenized.stats.sum_rows_sent_per_call	Righe inviate per chiamata
db.sql_tokenized.stats.sum_created_tmp_disk_tables_per_call	Tabelle disco temporanee create per chiamata
db.sql_tokenized.stats.sum_created_tmp_tables_per_call	Tabelle temporanee create per chiamata
db.sql_tokenized.stats.sum_lock_time_per_call	Tempo di blocco per chiamata (in ms)

Statistiche SQL per Oracle

Amazon RDS for Oracle raccoglie statistiche SQL sia a livello di istruzione che di digest. A livello di istruzione, la colonna ID rappresenta il valore di `V$SQL.SQL_ID`. A livello di digest, la colonna ID mostra il valore di `V$SQL.FORCE_MATCHING_SIGNATURE`.

Se l'ID è `0` a livello di digest, Oracle Database ha stabilito che questa istruzione non è adatta al riutilizzo. In questo caso, le istruzioni SQL figlie potrebbero appartenere a diversi digest. Tuttavia, le istruzioni sono raggruppate sotto `digest_text` per la prima istruzione SQL raccolta.

Argomenti

- Statistiche al secondo per Oracle

- Statistiche per chiamata per Oracle

Statistiche al secondo per Oracle

I seguenti parametri forniscono statistiche al secondo per una query Oracle SQL.

Parametro	Unità
db.sql.stats.executions_per_sec	Numero di esecuzioni al secondo
db.sql.stats.elapsed_time_per_sec	Media delle esecuzioni attive (AAE, Average active executions)
db.sql.stats.rows_processed_per_sec	Righe elaborate al secondo
db.sql.stats.buffer_gets_per_sec	Letture del buffer al secondo
db.sql.stats.physical_read_requests_per_sec	Letture fisiche al secondo
db.sql.stats.physical_write_requests_per_sec	Scritture fisiche al secondo
db.sql.stats.total_sharable_mem_per_sec	Memoria condivisibile totale al secondo (in byte)
db.sql.stats.cpu_time_per_sec	Tempo CPU al secondo (in minuti)

I seguenti parametri forniscono statistiche per chiamaat per una query digest Oracle SQL.

Parametro	Unità
db.sql_tokenized.stats.executions_per_sec	Numero di esecuzioni al secondo
db.sql_tokenized.stats.elapsed_time_per_sec	Media delle esecuzioni attive (AAE, Average active executions)
db.sql_tokenized.stats.rows_processe d_per_sec	Righe elaborate al secondo
db.sql_tokenized.stats.buffer_gets_per_sec	Letture del buffer al secondo

Parametro	Unità
db.sql_tokenized.stats.physical_read_requests_per_sec	Letture fisiche al secondo
db.sql_tokenized.stats.physical_write_requests_per_sec	Scritture fisiche al secondo
db.sql_tokenized.stats.total_sharable_mem_per_sec	Memoria condivisibile totale al secondo (in byte)
db.sql_tokenized.stats.cpu_time_per_sec	Tempo CPU al secondo (in minuti)

Statistiche per chiamata per Oracle

I seguenti parametri forniscono statistiche per chiamata di un'istruzione SQL Oracle.

Parametro	Unità
db.sql.stats.elapsed_time_per_exec	Tempo trascorso per esecuzioni (in ms)
db.sql.stats.rows_processed_per_exec	Righe elaborate per esecuzione
db.sql.stats.buffer_gets_per_exec	Letture del buffer per esecuzione
db.sql.stats.physical_read_requests_per_exec	Letture fisiche per esecuzione
db.sql.stats.physical_write_requests_per_exec	Scritture fisiche per esecuzione
db.sql.stats.total_sharable_mem_per_exec	Memoria condivisibile totale per esecuzione (in byte)
db.sql.stats.cpu_time_per_exec	Tempo CPU per esecuzione (in minuti)

I seguenti parametri forniscono statistiche per chiamata per una query digest Oracle SQL.

Parametro	Unità
db.sql_tokenized.stats.elapsed_time_per_exec	Tempo trascorso per esecuzioni (in minuti)
db.sql_tokenized.stats.rows_processed_per_exec	Righe elaborate per esecuzione
db.sql_tokenized.stats.buffer_gets_per_exec	Letture del buffer per esecuzione
db.sql_tokenized.stats.physical_read_requests_per_exec	Letture fisiche per esecuzione
db.sql_tokenized.stats.physical_write_requests_per_exec	Scritture fisiche per esecuzione
db.sql_tokenized.stats.total_sharable_mem_per_exec	Memoria condivisibile totale per esecuzione (in byte)
db.sql_tokenized.stats.cpu_time_per_exec	Tempo CPU per esecuzione (in minuti)

Statistiche SQL per SQL Server

Amazon RDS per SQL Server raccoglie statistiche SQL sia a livello di istruzione che di digest. A livello di istruzione, la colonna ID rappresenta il valore di `sql_handle`. A livello di digest, la colonna ID mostra il valore di `query_hash`.

SQL Server restituisce valori NULL per `query_hash` per alcune istruzioni. Ad esempio, ALTER INDEX, CHECKPOINT, UPDATE STATISTICS, COMMIT TRANSACTION, FETCH NEXT FROM Cursor e alcune istruzioni INSERT, SELECT @<variable>, istruzioni condizionali e stored procedure eseguibili. In questo caso, il valore `sql_handle` viene visualizzato come ID a livello di digest per quell'istruzione.

Argomenti

- Statistiche al secondo per SQL Server
- Statistiche per chiamata per SQL Server

Statistiche al secondo per SQL Server

I seguenti parametri forniscono statistiche al secondo per una query SQL di SQL Server.

Parametro	Unità
db.sql.stats.execution_count_per_sec	Numero di esecuzioni al secondo
db.sql.stats.total_elapsed_time_per_sec	Tempo totale trascorso al secondo
db.sql.stats.total_rows_per_sec	Righe totali elaborate al secondo
db.sql.stats.total_logical_reads_per_sec	Letture logiche totali al secondo
db.sql.stats.total_logical_writes_per_sec	Scritture logiche totali al secondo
db.sql.stats.total_physical_reads_per_sec	Letture fisiche totali al secondo
db.sql.stats.total_worker_time_per_sec	Tempo totale della CPU (in ms)

I seguenti parametri forniscono statistiche al secondo per una query digest SQL di SQL Server.

Parametro	Unità
db.sql_tokenized.stats.execution_count_per_sec	Numero di esecuzioni al secondo
db.sql_tokenized.stats.total_elapsed_time_per_sec	Tempo totale trascorso al secondo
db.sql_tokenized.stats.total_rows_per_sec	Righe totali elaborate al secondo
db.sql_tokenized.stats.total_logical_reads_per_sec	Letture logiche totali al secondo
db.sql_tokenized.stats.total_logical_writes_per_sec	Scritture logiche totali al secondo
db.sql_tokenized.stats.total_physical_reads_per_sec	Letture fisiche totali al secondo

Parametro	Unità
db.sql_tokenized.stats.total_worker_time_per_sec	Tempo totale della CPU (in ms)

Statistiche per chiamata per SQL Server

I seguenti parametri forniscono le statistiche per chiamata di un'istruzione SQL di SQL Server.

Parametro	Unità
db.sql.stats.total_elapsed_time_per_call	Tempo totale trascorso per esecuzione
db.sql.stats.total_rows_per_call	Righe totali elaborate per esecuzione
db.sql.stats.total_logical_reads_per_call	Letture logiche totali per esecuzione
db.sql.stats.total_logical_writes_per_call	Scritture logiche totali per esecuzione
db.sql.stats.total_physical_reads_per_call	Letture fisiche totali per esecuzione
db.sql.stats.total_worker_time_per_call	Tempo CPU totale per esecuzione (in ms)

I seguenti parametri forniscono statistiche per una query digest SQL di SQL Server.

Parametro	Unità
db.sql_tokenized.stats.total_elapsed_time_per_call	Tempo totale trascorso per esecuzione
db.sql_tokenized.stats.total_rows_per_call	Righe totali elaborate per esecuzione
db.sql_tokenized.stats.total_logical_reads_per_call	Letture logiche totali per esecuzione
db.sql_tokenized.stats.total_logical_writes_per_call	Scritture logiche totali per esecuzione

Parametro	Unità
db.sql_tokenized.stats.total_physical_reads_per_call	Letture fisiche totali per esecuzione
db.sql_tokenized.stats.total_worker_time_per_call	Tempo CPU totale per esecuzione (in ms)

Statistiche SQL per RDS PostgreSQL

Per ogni chiamata SQL e per ogni secondo di esecuzione di una query, Performance Insights raccoglie statistiche SQL. RDS per PostgreSQL raccoglie statistiche SQL solo a livello di digest. Nessuna statistica viene mostrata a livello di istruzione.

Di seguito sono disponibili informazioni sulle statistiche a livello di digest per RDS per PostgreSQL.

Argomenti

- Statistiche digest per RDS PostgreSQL
- Statistiche digest al secondo per RDS PostgreSQL
- Statistiche digest per chiamata per RDS PostgreSQL

Statistiche digest per RDS PostgreSQL

Per visualizzare le statistiche digest SQL è necessario caricare la libreria `pg_stat_statements` RDS PostgreSQL. Per le istanze database PostgreSQL compatibili con PostgreSQL 11 o versioni successive, il database carica questa libreria per impostazione di default. Per le istanze database PostgreSQL compatibili con PostgreSQL 10 o versioni precedenti, è possibile abilitare questa libreria manualmente. Per abilitarla manualmente, aggiungere `pg_stat_statements` a `shared_preload_libraries` nel gruppo parametri del database associati all'istanza database. Riavviare quindi l'istanza database. Per ulteriori informazioni, consulta Utilizzo di gruppi di parametri.

> **Note**
>
> Performance Insights può raccogliere statistiche solo per le query non troncate in `pg_stat_activity`. Per impostazione predefinita, i database PostgreSQL troncano le query più lunghe di 1.024 byte. Per aumentare la dimensione della query, modificare il parametro `track_activity_query_size` nel gruppo di parametri database associato

all'istanza database. Quando si modifica questo parametro, è necessario riavviare un'istanza database.

Statistiche digest al secondo per RDS PostgreSQL

Le seguenti statistiche digest SQL sono disponibili per le istanze database PostgreSQL.

Parametro	Unità
db.sql_tokenized.stats.calls_per_sec	Chiamate al secondo
db.sql_tokenized.stats.rows_per_sec	Righe al secondo
db.sql_tokenized.stats.total_time_per_sec	Media delle esecuzioni attive al secondo (AAE, Average active executions)
db.sql_tokenized.stats.shared_blks_hit_per_sec	Richieste in blocco al secondo
db.sql_tokenized.stats.shared_blks_read_per_sec	Letture in blocco al secondo
db.sql_tokenized.stats.shared_blks_dirtied_per_sec	Blocchi sporchi al secondo
db.sql_tokenized.stats.shared_blks_written_per_sec	Scritture in blocco al secondo
db.sql_tokenized.stats.local_blks_hit_per_sec	Richieste in blocco locale al secondo
db.sql_tokenized.stats.local_blks_read_per_sec	Letture di blocchi locali al secondo
db.sql_tokenized.stats.local_blks_dirtied_per_sec	Blocco locale danneggiato al secondo
db.sql_tokenized.stats.local_blks_written_per_sec	Scritture di blocchi locali al secondo
db.sql_tokenized.stats.temp_blks_written_per_sec	Scritture temporanee al secondo

Parametro	Unità
db.sql_tokenized.stats.temp_blks_read_per_sec	Letture temporanee al secondo
db.sql_tokenized.stats.blk_read_time_per_sec	Letture medie simultanee al secondo
db.sql_tokenized.stats.blk_write_time_per_sec	Scritture medie simultanee al secondo

Statistiche digest per chiamata per RDS PostgreSQL

I seguenti parametri forniscono le statistiche per chiamata di un'istruzione SQL.

Parametro	Unità
db.sql_tokenized.stats.rows_per_call	Righe per chiamata
db.sql_tokenized.stats.avg_latency_per_call	Latenza media per chiamata (in ms)
db.sql_tokenized.stats.shared_blks_hit_per_call	Richieste in blocco per chiamata
db.sql_tokenized.stats.shared_blks_read_per_call	Letture in blocco per chiamata
db.sql_tokenized.stats.shared_blks_written_per_call	Scritture in blocco per chiamata
db.sql_tokenized.stats.shared_blks_dirtied_per_call	Blocchi danneggiati per chiamata
db.sql_tokenized.stats.local_blks_hit_per_call	Richieste in blocco locale per chiamata
db.sql_tokenized.stats.local_blks_read_per_call	Letture di blocchi locali per chiamata
db.sql_tokenized.stats.local_blks_dirtied_per_call	Blocco locale danneggiato per chiamata
db.sql_tokenized.stats.local_blks_written_per_call	Scritture di blocchi locali per chiamata

Parametro	Unità
db.sql_tokenized.stats.temp_blks_written_per_call	Scritture temporanee di blocchi per chiamata
db.sql_tokenized.stats.temp_blks_read_per_call	Letture temporanee di blocchi per chiamata
db.sql_tokenized.stats.blk_read_time_per_call	Tempo di lettura per chiamata (in ms)
db.sql_tokenized.stats.blk_write_time_per_call	Tempo di scrittura per chiamata (in ms)

Per ulteriori informazioni su questi parametri, consultare pg_stat_statements nella documentazione PostgreSQL.

Parametri del sistema operativo nel monitoraggio avanzato

Amazon RDS fornisce parametri in tempo reale per il sistema operativo sul quale è in esecuzione il cluster di . RDS fornisce i parametri di Enhanced Monitoring al tuo account Amazon CloudWatch Logs. Le tabelle seguenti elencano i parametri del sistema operativo disponibili utilizzando Amazon CloudWatch Logs.

Argomenti

- Metriche del sistema operativo per Db2, MariaDB, MySQL, Oracle e PostgreSQL
- Parametri del sistema operativo per Microsoft SQL Server

Metriche del sistema operativo per Db2, MariaDB, MySQL, Oracle e PostgreSQL

Group (Gruppo)	Parametro	Nome console	Descrizione
`General`	`engine`	Non applicabile	Il motore del database per l'istanza database.
	`instanceID`	Non applicabile	Identificatore istanze DB.

Group (Gruppo)	Parametro	Nome console	Descrizione
	`instanceResourceID`	Non applicabile	Un identificatore immutabile per l'istanza database che è univoco per una regione AWS, utilizzato anche come identificatore del flusso di log.
	`numVCPUs`	Non applicabile	Il numero di CPU virtuali per l'istanza database.
	`timestamp`	Non applicabile	L'ora in cui sono stati presi i parametri.
	`uptime`	Non applicabile	Il periodo di esecuzione dell'istanza database è stato attivato.
	`version`	Non applicabile	La versione del formato JSON del flusso dei parametri del sistema operativo.
`cpuUtilization`	`guest`	Ospite CPU	La percentuale di CPU utilizzata dai programmi guest.
	`idle`	CPU inattiva	La percentuale di tempo CPU che è inattiva.
	`irq`	IRQ CPU	La percentuale di CPU utilizzata dalle interruzioni dei software.
	`nice`	CPU Nice	La percentuale di CPU utilizzata dai programmi in esecuzione con priorità più bassa.
	`steal`	Stealing della CPU	La percentuale di CPU utilizzata da altre macchine virtuali.
	`system`	Sistema CPU	La percentuale di CPU utilizzata dal kernel.
	`total`	Totale CPU	La percentuale totale del CPU utilizzata. Questo valore include il valore `nice`.

Group (Gruppo)	Parametro	Nome console	Descrizione
	`user`	Utente CPU	La percentuale di CPU utilizzata dai programmi utente.
	`wait`	Attesa CPU	La percentuale di CPU non utilizzata durante l'attesa per l'accesso I/O.
`diskIO`	`avgQueueLen`	Dimensione media coda	Il numero di richieste in attesa nella coda del dispositivo I/O.
	`avgReqSz`	Dimensione richiesta media	La dimensione della richiesta media, in kilobyte.
	`await`	I/O su disco in attesa	Il numero di millisecondi necessari per rispondere alle richieste, compreso il tempo della coda e il tempo del servizio.
	`device`	Non applicabile	L'identificatore del dispositivo del disco in uso.
	`readIOsPS`	IO/s di lettura	Il numero di operazioni di lettura al secondo.
	`readKb`	Totale lettura	Il numero totale di kilobyte letti.
	`readKbPS`	KB/s di lettura	Il numero di kilobytes letti al secondo.
	`readLatency`	Latenza di lettura	Il tempo trascorso tra l'invio di una richiesta di I/O di lettura e il relativo completamento, espresso in millisecondi. Questo parametro è disponibile solo per Amazon Aurora.

Group (Gruppo)	Parametro	Nome console	Descrizione
	`readThroughput`	Throughput di lettura	La quantità di velocità effettiva di rete utilizzata dalle richieste al cluster di database, espressa in byte al secondo. Questo parametro è disponibile solo per Amazon Aurora.
	`rrqmPS`	Rqms	Il numero di richieste di lettura unite in coda al secondo.
	`tps`	TPS	Il numero di transazioni I/O al secondo.
	`util`	Util I/O su disco	La percentuale di tempo della CPU durante il quale sono state emesse le richieste.
	`writeIOsPS`	IO/s di scrittura	Il numero di operazioni di scrittura al secondo.
	`writeKb`	Totale scrittura	Il numero totale di kilobyte scritti.
	`writeKbPS`	KB/s di scrittura	Il numero di kilobytes scritti al secondo.
	`writeLatency`	Latenza di scrittura	Tempo medio trascorso tra l'invio di una richiesta di I/O di scrittura e il relativo completamento, in millisecondi. Questo parametro è disponibile solo per Amazon Aurora.
	`writeThroughput`	Throughput di scrittura	La quantità di throughput di rete effettivo utilizzato dalle risposte del cluster di database, espressa in byte al secondo. Questo parametro è disponibile solo per Amazon Aurora.

Group (Gruppo)	Parametro	Nome console	Descrizione
	`wrqmPS`	Wrqms	Il numero di richieste di scrittura unite in coda al secondo.
`physicalDeviceIO`	`avgQueueLen`	Dimensione media coda dispositivi fisici	Il numero di richieste in attesa nella coda del dispositivo I/O.
	`avgReqSz`	Dimensione richiesta Ave dispositivi fisici	La dimensione della richiesta media, in kilobyte.
	`await`	Dispositivi fisici disco I/O in attesa	Il numero di millisecondi necessari per rispondere alle richieste, compreso il tempo della coda e il tempo del servizio.
	`device`	Non applicabile	L'identificatore del dispositivo del disco in uso.
	`readIOsPS`	Dispositivi fisici Lettura IO/s	Il numero di operazioni di lettura al secondo.
	`readKb`	Totale lettura dispositivi fisici	Il numero totale di kilobyte letti.

Group (Gruppo)	Parametro	Nome console	Descrizione
	`readKbPS`	Dispositivi fisici Lettura Kb/s	Il numero di kilobytes letti al secondo.
	`rrqmPS`	Dispositivi fisici Rrqms	Il numero di richieste di lettura unite in coda al secondo.
	`tps`	Dispositivi fisici TPS	Il numero di transazioni I/O al secondo.
	`util`	Dispositivi fisici I/O disco Util	La percentuale di tempo della CPU durante il quale sono state emesse le richieste.
	`writeIOsPS`	Dispositivi fisici scrittura IO/s	Il numero di operazioni di scrittura al secondo.
	`writeKb`	Totale scrittura dispositivi fisici	Il numero totale di kilobyte scritti.
	`writeKbPS`	Dispositivi fisici in scrittura Kb/s	Il numero di kilobytes scritti al secondo.
	`wrqmPS`	Dispositivi fisici Wrqms	Il numero di richieste di scrittura unite in coda al secondo.

Group (Gruppo)	Parametro	Nome console	Descrizione
`fileSys`	`maxFiles`	Inode max	Il numero massimo di file che è possibile creare per il sistema di file.
	`mountPoint`	Non applicabile	Il percorso del sistema di file.
	`name`	Non applicabile	Il nome del sistema di file.
	`total`	File system totale	Il numero totale di spazio su disco disponibile per il sistema di file, in kilobyte.
	`used`	Filesystem usato	La quantità totale di spazio su disco utilizzato dai file nel sistema di file, in kilobyte.
	`usedFilePercent`	Inode usati	La percentuale di file disponibili in uso.
	`usedFiles`	% di utilizzo	Il numero di file audio nel sistema di file.
	`usedPercent`	Filesystem usato	La percentuale dello spazio su disco del sistema di file in uso.
`loadAverageMinute`	`fifteen`	Carico medio 15 min	Il numero di processi che richiedono l'ora della CPU negli ultimi 15 minuti.
	`five`	Carico medio 5 min	Il numero di processi che richiedono l'ora della CPU negli ultimi 5 minuti.
	`one`	Carico medio 1 min	Il numero di processi che richiedono l'ora della CPU nell'ultimo minuto.

Group (Gruppo)	Parametro	Nome console	Descrizione
memory	active	Memoria attiva	La quantità di memoria assegnata, in kilobyte.
	buffers	Memoria bufferizzata	La quantità di memoria utilizzata per il buffering delle richieste di I/O prima della scrittura sul dispositivo di storage, in kilobyte.
	cached	Memoria cache	La quantità di memoria utilizzata per la memorizzazione nella cache dell'I/O basato sul file system.
	dirty	Memoria sporca	La quantità di pagine di memoria nella RAM che sono state modificate ma non scritte nel relativo blocco di dati nello storage, in kilobyte.
	free	Memoria libera	La quantità di memoria non assegnata, in kilobyte.
	hugePagesFree	Huge Pages libere	Il numero di pagine di grandi dimensioni gratuite. Le pagine di grandi dimensioni sono una caratteristica del kernel di Linux.
	hugePagesRsvd	Huge Pages Rsvd	Il numero di pagine di grandi dimensioni impegnate.
	hugePagesSize	Dimensioni Huge Pages	La dimensione per ogni unità delle pagine di grandi dimensioni, in kilobyte.
	hugePagesSurp	Huge Pages Surp	Il numero di pagine di grandi dimensioni in eccesso disponibili sul totale.
	hugePagesTotal	Totale Huge Pages	Il numero totale di Huge Pages.

Group (Gruppo)	Parametro	Nome console	Descrizione
	`inactive`	Memoria inattiva	La quantità di pagine di memoria utilizzate meno frequentemente, in kilobyte.
	`mapped`	Memoria mappata	La quantità totale di contenuti del sistema di file che è mappata in memoria all'interno di uno spazio di indirizzamento del processo, in kilobyte.
	`pageTables`	Tabelle delle pagine	La quantità di memoria utilizzata dalle tabelle della pagina, in kilobyte.
	`slab`	Memoria slab	La quantità di strutture dati del kernel riutilizzabili, in kilobyte.
	`total`	Memoria totale	La quantità totale di memoria, in kilobyte.
	`writeback`	Memoria writeback	La quantità di pagine sporche nella RAM che sono ancora scritte nello storage di backup, in kilobyte.
`network`	`interface`	Non applicabile	L'identificatore per l'interfaccia di rete utilizzato per l'istanza database.
	`rx`	RX	Il numero di bytes ricevuti al secondo.
	`tx`	TX	Il numero di bytes caricati al secondo.
`processList`	`cpuUsedPc`	CPU %	La percentuale di CPU utilizzata dal processo.
	`id`	Non applicabile	L'identificatore del processo.
	`memoryUsedPc`	MEM%	Percentuale della memoria totale utilizzata dal processo.

Group (Gruppo)	Parametro	Nome console	Descrizione
	`name`	Non applicabile	Il nome del processo.
	`parentID`	Non applicabile	L'identificatore del processo per il processo genitore del processo.
	`rss`	RES	La quantità di RAM allocata al processo, in kilobyte.
	`tgid`	Non applicabile	L'identificatore del gruppo di thread, che è un numero che rappresenta l'ID del processo a cui appartiene un thread. Questo identificatore è utilizzato per raggruppa re i thread dallo stesso processo.
	`vss`	VIRT	La quantità di memoria virtuale allocata al processo, in kilobyte.
`swap`	`swap`	Swap	La quantità di memoria di scambio disponibile, in kilobyte.
	`swap in`	Swap in	Quantità di memoria, in kilobyte, scambiata in ingresso nel disco.
	`swap out`	Swap out	Quantità di memoria, in kilobyte, scambiata in uscita dal disco.
	`free`	Swap libera	La quantità di memoria di scambio libera, in kilobyte.
	`committed`	Swap occupata	La quantità di memoria di scambio, in kilobyte, utilizzat a come memoria cache.
`tasks`	`blocked`	Attività bloccate	Il numero di attività che sono bloccate.
	`running`	Attività in esecuzione	Il numero di attività che sono in esecuzione.

Group (Gruppo)	Parametro	Nome console	Descrizione
	`sleeping`	Attività inattive	Il numero di attività che sono a riposo.
	`stopped`	Attività interrotte	Il numero di attività che sono arrestate.
	`total`	Totale attività	Il numero totale di attività.
	`zombie`	Attività Zombie	Il numero di attività secondarie che sono inattive con un'attività genitore attiva.

Parametri del sistema operativo per Microsoft SQL Server

Group (Gruppo)	Parametro	Nome console	Descrizione
`General`	`engine`	Non applicabile	Il motore del database per l'istanza database.
	`instanceID`	Non applicabile	Identificatore istanze DB.
	`instanceResourceID`	Non applicabile	Un identificatore immutabile per l'istanza database che è univoco per una regione AWS, utilizzato anche come identificatore del flusso di log.
	`numVCPUs`	Non applicabile	Il numero di CPU virtuali per l'istanza database.
	`timestamp`	Non applicabile	L'ora in cui sono stati presi i parametri.

Group (Gruppo)	Parametro	Nome console	Descrizione
	`uptime`	Non applicabile	Il periodo di esecuzione dell'istanza database è stato attivato.
	`version`	Non applicabile	La versione del formato JSON del flusso dei parametri del sistema operativo.
`cpuUtilization`	`idle`	CPU inattiva	La percentuale di tempo CPU che è inattiva.
	`kern`	Kernel CPU	La percentuale di CPU utilizzata dal kernel.
	`user`	Utente CPU	La percentuale di CPU utilizzata dai programmi utente.
`disks`	`name`	Non applicabile	L'identificatore del disco.
	`totalKb`	Spazio totale su disco	Lo spazio totale del disco, in kilobyte.
	`usedKb`	Spazio su disco utilizzato	La quantità di spazio utilizzato sul disco, in kilobyte.
	`usedPc`	% di utilizzo spazio su disco	La percentuale di spazio utilizzato sul disco, in kilobyte.
	`availKb`	Spazio su disco disponibile	Lo spazio disponibile sul disco, in kilobyte.
	`availPc`	% di spazio su disco disponibile	La percentuale di spazio disponibile sul disco, in kilobyte.
	`rdCountPS`	Letture/s	Il numero di operazioni di lettura al secondo

Group (Gruppo)	Parametro	Nome console	Descrizione
	`rdBytesPS`	KB/s di lettura	Il numero di bytes letti al secondo.
	`wrCountPS`	IO/s di scrittura	Il numero di operazioni di scrittura al secondo.
	`wrBytesPS`	KB/s di scrittura	La quantità di byte scritti al secondo.
`memory`	`commitTotKb`	Totale commit	La quantità di spazio di indirizzi virtuali del supporto del foglio di pagina in uso, ovvero il carico di commit corrente. Questo valore è composto da memoria principale (RAM) e disco (file di paging).
	`commitLimitKb`	Commit massimo	Il valore massimo possibile per il parametro `commitTotKb` . Questo valore è la somma delle dimensioni correnti del file di paging e della memoria fisica disponibile per i contenuti ; esclusa la RAM assegnata alle aree non paginabili.
	`commitPeakKb`	Picco commit	Il valore più grande del parametro `commitTotKb` da quando è stato avviato il sistema operativo per l'ultima volta.
	`kernTotKb`	Memoria totale del kernel	La somma della memoria nei pool di kernel paginabili e non paginabili, in kilobyte.
	`kernPagedKb`	Memoria kernel di paging	La quantità di memoria nel pool di kernel paginabile, in kilobyte.

Group (Gruppo)	Parametro	Nome console	Descrizione
	kernNonpagedKb	Memoria Kernel non paginabile	La quantità di memoria nel pool di kernel non paginabile, in kilobyte.
	pageSize	Dimensioni pagina	La dimensione di una pagina, in byte.
	physTotKb	Memoria totale	La quantità di memoria fisica, in kilobyte.
	physAvailKb	Memoria disponibile	La quantità di memoria fisica disponibile, in kilobyte.
	sqlServerTotKb	Memoria totale di SQL Server	La quantità di memoria impegnata in SQL Server, in kilobyte.
	sysCacheKb	Cache di sistema	La quantità di memoria della cache di sistema, in kilobyte.
network	interface	Non applicabile	L'identificatore per l'interfaccia di rete utilizzato per l'istanza database.
	rdBytesPS	KB/s di lettura di rete	Il numero di bytes ricevuti al secondo.
	wrBytesPS	Scrittura di rete KB/s	Il numero di bytes inviati al secondo.
processList	cpuUsedPc	% di utilizzo	La percentuale di CPU utilizzata dal processo.
	memUsedPc	MEM%	Percentuale della memoria totale utilizzata dal processo.
	name	Non applicabile	Il nome del processo.

Group (Gruppo)	Parametro	Nome console	Descrizione
	`pid`	Non applicabile	L'identificatore del processo. Questo valore non è presente per i processi di proprietà di Amazon RDS.
	`ppid`	Non applicabile	L'identificatore del processo per il genitore di questo processo. Questo valore è presente solo per i processi secondari.
	`tid`	Non applicabile	L'identificatore del thread. Questo valore è presente solo per i thread. Il processo proprietario può essere identificato usando il valore `pid`.
	`workingSetKb`	Non applicabile	La quantità di memoria nel set di lavoro privato più la quantità di memoria che è in utilizzata dal processo e che può essere condivisa con altri processi, in kilobyte.
	`workingSetPrivKb`	Non applicabile	La quantità di memoria utilizzata da un processo, ma che non può essere condivisa con altri processi, in kilobyte.
	`workingSetShareableKb`	Non applicabile	La quantità di memoria utilizzata da un processo e che può essere condivisa con altri processi, in kilobyte.
	`virtKb`	Non applicabile	La quantità di spazio di indirizzi virtuali utilizzato dal processo, in kilobyte. L'uso dello spazio degli indirizzi virtuali non implica necessariamente l'uso corrispondente del disco o delle pagine di memoria principale.
`system`	`handles`	Handle	Il numero di handle utilizzati dal sistema.
	`processes`	Processes	Il numero totale di processi in esecuzione sul sistema.

Group (Gruppo)	Parametro	Nome console	Descrizione
	`threads`	Thread	Il numero totale di thread in esecuzione sul sistema.

Monitoraggio di eventi, registri e flussi in un'istanza di database Amazon RDS

Quando monitori i tuoi database Amazon RDS Aurora e AWS altre soluzioni, il tuo obiettivo è mantenere quanto segue:

- Affidabilità
- Disponibilità
- Prestazioni
- Sicurezza

Monitoraggio di parametri in un'istanza Amazon RDS spiega come monitorare l'istanza usando i parametri. Una soluzione completa deve inoltre monitorare gli eventi del database, i file di log e i flussi di attività. AWS fornisce i seguenti strumenti di monitoraggio:

- Amazon EventBridge è un servizio di bus eventi senza server che semplifica la connessione delle applicazioni con dati provenienti da una varietà di fonti. EventBridge offre un flusso di dati in tempo reale dalle tue applicazioni, applicazioni oftware-as-a S-Service (SaaS) e servizi. AWS EventBridge indirizza tali dati verso obiettivi come. AWS Lambda In questo modo, puoi monitorare gli eventi che si verificano nei servizi e creare architetture basate su eventi. Per ulteriori informazioni, consulta la Amazon EventBridge User Guide.
- Amazon CloudWatch Logs offre un modo per monitorare, archiviare e accedere ai file di log da istanze Amazon RDS AWS CloudTrail, e altre fonti. Amazon CloudWatch Logs può monitorare le informazioni nei file di registro e avvisarti quando vengono raggiunte determinate soglie. Puoi inoltre archiviare i dati del log in storage estremamente durevole. Per ulteriori informazioni, consulta la Amazon CloudWatch Logs User Guide.
- AWS CloudTrailacquisisce le chiamate API e gli eventi correlati effettuati da o per conto del tuo. Account AWS CloudTrail consegna i file di log a un bucket Amazon S3 specificato dall'utente. Puoi identificare quali utenti e account hanno effettuato le chiamate AWS, l'indirizzo IP di origine da cui sono state effettuate le chiamate e quando sono avvenute le chiamate. Per ulteriori informazioni, consulta la Guida per l'utente AWS CloudTrail.
- I flussi di attività di database sono una funzionalità di Amazon RDS che fornisce un flusso quasi in tempo reale dell'attività nell'istanza database. Amazon RDS inserisce le attività in Amazon Kinesis

Data Streams. Il flusso Kinesis viene creato automaticamente. Da Kinesis, puoi configurare AWS servizi come Amazon Data Firehose e consumare lo stream e AWS Lambda archiviare i dati.

Argomenti

Visualizzazione di registri, eventi e flussi nella console Amazon RDS

Amazon RDS si integra con Servizi AWS per visualizzare informazioni su registri, eventi e flussi di attività del database nella console RDS.

La scheda Logs & events (Registri ed eventi) per l'istanza di database RDS mostra le informazioni seguenti:

- Amazon CloudWatch alarms (Allarmi di Amazon CloudWatch): mostra eventuali allarmi dei parametri configurati per l'istanza database . Se non hai configurato allarmi, puoi crearli nella console di RDS. Per ulteriori informazioni, consulta Monitoraggio dei parametri di Amazon RDS con Amazon CloudWatch.
- Recent events (Eventi recenti): mostra un riepilogo degli eventi (modifiche all'ambiente) per l'istanza database RDS. Per ulteriori informazioni, consulta Visualizzazione di eventi Amazon RDS.
- Log (Registro): mostra i file di log del database generati da un'istanza database . Per ulteriori informazioni, consulta Monitoraggio dei file di log di Amazon RDS.

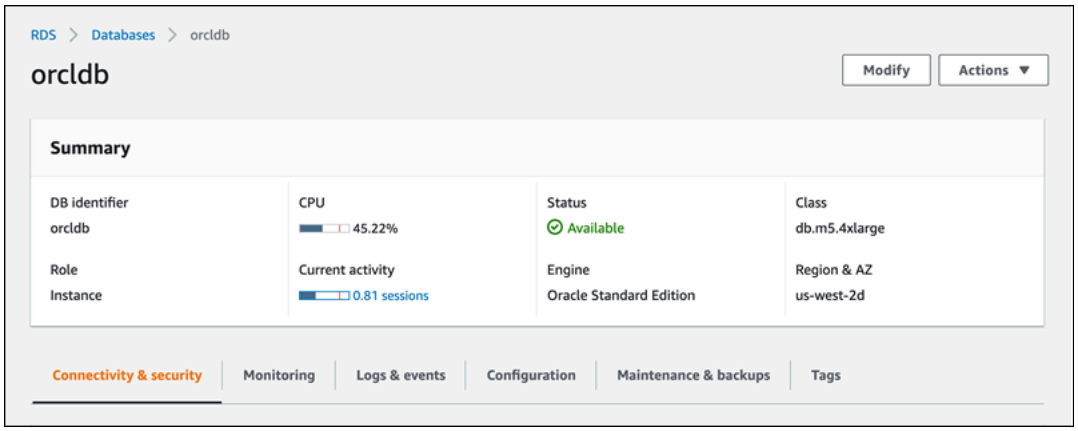
La scheda Configuration (Configurazione) mostra le informazioni sui flussi di attività di database.

Per visualizzare registri, eventi e flussi per l'istanza di database nella console RDS

1. Accedi alla AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel pannello di navigazione, scegliere Databases (Database).

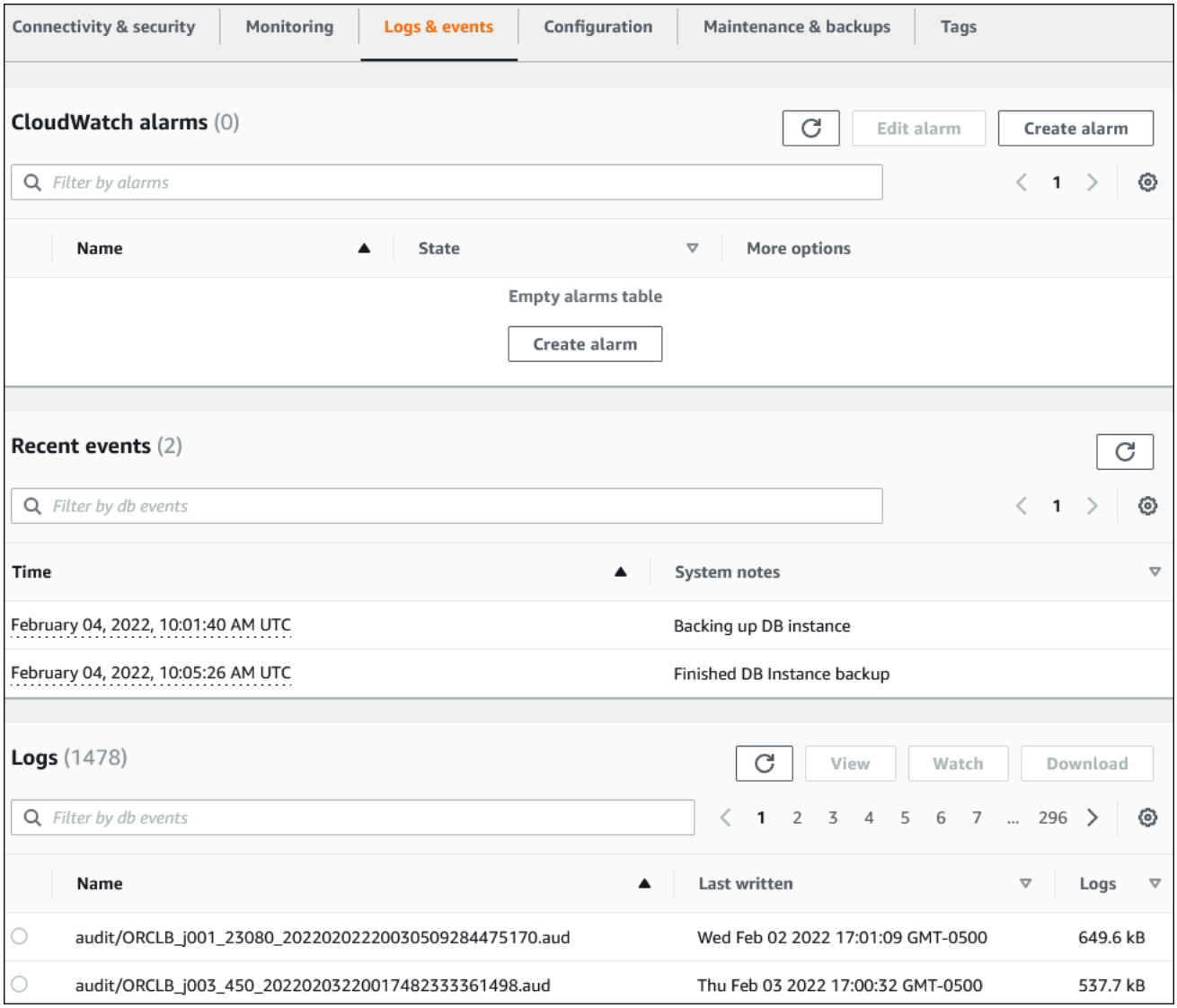
3. Scegliere il nome del di database che si desidera monitorare.

   Verrà visualizzata la pagina Databases (Database). L'esempio seguente mostra un database Oracle denominato `orclb`.



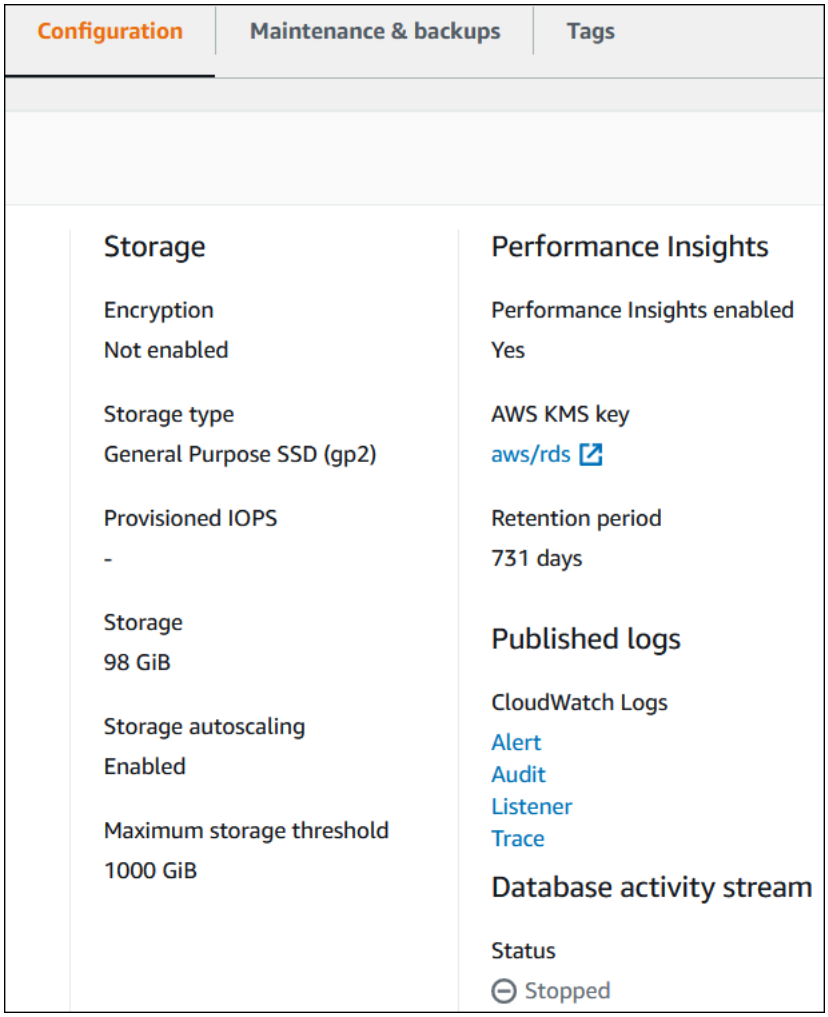
4. Scegliere Logs & events (Log ed eventi).

   Viene visualizzata la sezione Logs & events (Log ed eventi).



5. Scegliere Configuration (Configurazione).

L'esempio seguente mostra lo stato dei flussi di attività del database per l'istanza database.


Configuration
Maintenance & backups
Tags
Storage
Encryption
Not enabled
Storage type
General Purpose SSD (gp2)
Provisioned IOPS
-
Storage
98 GiB
Storage autoscaling
Enabled
Maximum storage threshold
1000 GiB
Performance Insights
Performance Insights enabled
Yes
AWS KMS key
aws/rds
Retention period
731 days
Published logs
CloudWatch Logs
Alert
Audit
Listener
Trace
Database activity stream
Status
Stopped

Monitoraggio di eventi Amazon RDS

Un evento indica una modifica in un ambiente. Questo può essere un ambiente AWS, un'applicazione o un servizio partner SaaS o uno dei servizi o applicazioni personalizzati. Per le descrizioni degli eventi RDS, consulta Categorie di eventi Amazon RDS e messaggi di evento.

Argomenti

Panoramica degli eventi per Amazon RDS

Un evento RDS indica una modifica nell'ambiente Amazon RDS. Ad esempio, Amazon Aurora di genera un evento quando lo stato di un'istanza database cambia da in sospeso a in esecuzione. Amazon RDS Aurora offre eventi quasi in tempo EventBridge reale.

> **Note**
>
> Amazon RDS emette eventi sulla base del massimo sforzo. Si consiglia di non scrivere programmi che dipendono dall'ordine o dall'esistenza di eventi di notifica, poiché potrebbero essere fuori sequenza o mancanti.

Amazon RDS registra gli eventi correlati alle seguenti risorse:

- Istanze DB

  Per un elenco degli eventi dell'istanza database, consulta Eventi di istanza database.
- Gruppi di parametri database

  Per un elenco degli eventi del gruppo parametri del database, consulta Eventi gruppo di parametri database.
- Gruppi di sicurezza DB

Per un elenco di eventi del gruppo di sicurezza DB, consulta Eventi gruppo di sicurezza DB.

- Snapshot DB

  Per un elenco di eventi degli snapshot DB, consulta Eventi degli snapshot DB.

- Eventi RDS Proxy

  Per un elenco degli eventi RDS Proxy, consulta Eventi RDS Proxy.

- Eventi di implementazione blu/verde

  Per l'elenco degli eventi di implementazione blu/verde, consulta Eventi di implementazione blu/verde.

Queste informazioni comprendono:

- La data e l'ora dell'evento
- Il nome di origine e il tipo di origine dell'evento
- Un messaggio associato all'evento
- Le notifiche degli eventi includono i tag che fanno riferimento al momento in cui il messaggio è stato inviato e potrebbero non riflettere i tag riferiti al momento in cui si è verificato l'evento.

Visualizzazione di eventi Amazon RDS

Puoi recuperare le seguenti informazioni sull'evento per le risorse Amazon RDS:

- Nome risorsa
- Tipo di risorsa
- Ora dell'evento
- Riepilogo del messaggio dell'evento

Accedi agli eventi tramite il AWS Management Console, che mostra gli eventi delle ultime 24 ore. Puoi anche recuperare gli eventi utilizzando il AWS CLI comando describe-events o l'DescribeEventsoperazione API RDS. Se utilizzi l'API RDS AWS CLI o l'API RDS per visualizzare gli eventi, puoi recuperare gli eventi relativi agli ultimi 14 giorni.

> **Note**
>
> Se devi archiviare eventi per periodi di tempo più lunghi, puoi inviare eventi Amazon RDS a EventBridge. Per ulteriori informazioni, consulta Creazione di una regola che si attiva su un evento Amazon RDS

Per le descrizioni degli eventi Amazon RDS, vedi Categorie di eventi Amazon RDS e messaggi di evento.

Per accedere a informazioni dettagliate sull'utilizzo degli eventi AWS CloudTrail, inclusi i parametri di richiesta, consultaEventi CloudTrail.

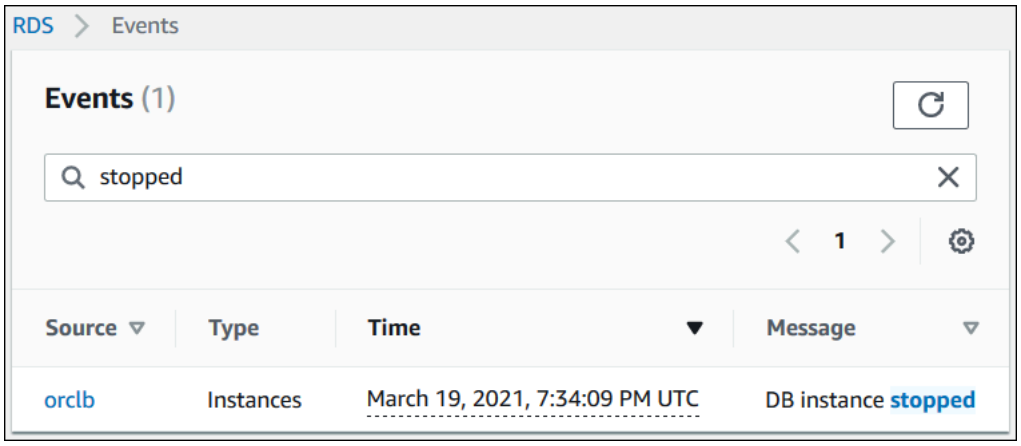
Console

Per visualizzare tutti gli eventi dell'istanza Amazon RDS delle ultime 24 ore

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel pannello di navigazione selezionare Events (Eventi).

   Gli eventi disponibili sono indicati all'interno di un elenco.
3. (Opzionale) Inserisci un termine di ricerca per filtrare i risultati.

Il seguente esempio mostra un elenco di eventi filtrati mediante i caratteri **`stopped`**.



AWS CLI

Per visualizzare tutti gli eventi generati nell'ultima ora, invoca describe-events senza parametri.

```
aws rds describe-events
```

Il seguente output di esempio mostra che un'istanza database è stata arrestata.

```
{
    "Events": [
        {
            "EventCategories": [
                "notification"
            ],
            "SourceType": "db-instance",
            "SourceArn": "arn:aws:rds:us-east-1:123456789012:db:testinst",
            "Date": "2022-04-22T21:31:00.681Z",
            "Message": "DB instance stopped",
            "SourceIdentifier": "testinst"
        }
    ]
}
```

Per visualizzare tutti gli eventi Amazon RDS degli ultimi 10080 minuti (7 giorni), chiama il AWS CLI comando describe-events e imposta il parametro su. `--duration 10080`

```
aws rds describe-events --duration 10080
```

L'esempio seguente mostra gli eventi nell'intervallo di tempo specificato per l'istanza database *test-instance*.

```
aws rds describe-events \
    --source-identifier test-instance \
    --source-type db-instance \
    --start-time 2022-03-13T22:00Z \
    --end-time 2022-03-13T23:59Z
```

L'output di esempio seguente mostra lo stato di un backup.

```
{
    "Events": [
        {
            "SourceType": "db-instance",
            "SourceIdentifier": "test-instance",
            "EventCategories": [
                "backup"
            ],
            "Message": "Backing up DB instance",
            "Date": "2022-03-13T23:09:23.983Z",
            "SourceArn": "arn:aws:rds:us-east-1:123456789012:db:test-instance"
        },
        {
            "SourceType": "db-instance",
            "SourceIdentifier": "test-instance",
            "EventCategories": [
                "backup"
            ],
            "Message": "Finished DB Instance backup",
            "Date": "2022-03-13T23:15:13.049Z",
            "SourceArn": "arn:aws:rds:us-east-1:123456789012:db:test-instance"
        }
    ]
}
```

API

Puoi visualizzare tutti gli eventi delle istanze Amazon RDS degli ultimi 14 giorni chiamando l'operazione dell'API DescribeEventsRDS e impostando il `Duration` parametro su. `20160`

Utilizzo della notifica degli eventi di Amazon RDS

Amazon RDS usa Amazon Simple Notification Service (Amazon SNS) per fornire una notifica quando si verifica un evento Amazon RDS. Queste notifiche possono essere in qualsiasi forma supportata da Amazon SNS per una regione AWS, ad esempio un'e-mail, un SMS o una chiamata a un endpoint HTTP.

Argomenti

Panoramica delle notifiche eventi di Amazon RDS

Amazon RDS raggruppa gli eventi in categorie che puoi sottoscrivere, per ricevere una notifica quando si verifica un evento di tale categoria.

Argomenti

Risorse RDS idonee per la sottoscrizione di eventi

È possibile sottoscrivere una categoria di eventi per le seguenti risorse:

- Istanza database
- snapshot di database
- DB parameter group (Gruppo di parametri database)
- Gruppo di sicurezza DB
- Server proxy per RDS
- Versioni personalizzate del motore

Ad esempio, sottoscrivendo la categoria Backup per una determinata istanza database, riceverai una notifica ogni volta che si verifica un evento relativo al backup che interessa l'istanza database. Sottoscrivendo una categoria Modifica della configurazione per un'istanza database, riceverai una notifica quando l'istanza database viene modificata. Riceverai una notifica anche quando viene modificata la sottoscrizione a una notifica eventi.

È possibile creare alcuni sottoscrizioni diverse. Per esempio, è possibile creare una sottoscrizione che riceve tutte le notifiche eventi per tutte le istanze database e un'altra sottoscrizione che include solo eventi critici per un sottoinsieme di istanze database. Per la seconda sottoscrizione, specifica una o più istanze database nel filtro.

Per sottoscrivere una notifica eventi di Amazon RDS, procedi come indicato di seguito:

Per sottoscrivere una notifica eventi Amazon RDS, procedi come indicato di seguito:

1. Puoi creare un abbonamento per la notifica di eventi Amazon RDS utilizzando la console o l'API di Amazon RDS. AWS CLI

   Amazon RDS usa l'ARN di un argomento Amazon SNS per identificare ogni sottoscrizione. La console Amazon RDS crea automaticamente l'ARN quando crei la sottoscrizione. Crea l'ARN utilizzando la console Amazon SNS, o AWS CLI l'API Amazon SNS.
2. Amazon RDS invia un SMS o un'e-mail di approvazione all'indirizzo da te specificato nella sottoscrizione.
3. Puoi confermare la sottoscrizione selezionando il collegamento nella notifica ricevuta.
4. La console di Amazon RDS aggiorna la sezione My Event Subscriptions con lo stato della sottoscrizione.
5. Amazon RDS inizia inviando le notifiche agli indirizzi forniti al momento della creazione della sottoscrizione.

Per ulteriori informazioni su Identity and Access Management quando utilizzi Amazon SNS, consultaIdentity and Access Management in Amazon SNS nella Guida per sviluppatori di Amazon Simple Notification.

Puoi utilizzarla AWS Lambda per elaborare notifiche di eventi da un'istanza DB. Per ulteriori informazioni, consulta Using AWS Lambda with Amazon RDS nella AWS Lambda Developer Guide.

Consegna delle notifiche degli eventi RDS

Amazon RDS invia le notifiche all'indirizzo fornito al momento della creazione della sottoscrizione. La notifica può inserire gli attributi del messaggio che forniscono i metadati strutturati sul messaggio. Per ulteriori informazioni sugli attributi del messaggio, consulta Categorie di eventi Amazon RDS e messaggi di evento.

La distribuzione delle notifiche degli eventi può richiedere fino a cinque minuti.

> **Important**
>
> Amazon RDS non garantisce l'ordine degli eventi inviato in un flusso di eventi. Tale ordine è soggetto a modifiche.

Quando Amazon SNS invia una notifica a un endpoint HTTP o HTTPS sottoscritto, il corpo del messaggio POST inviato all'endpoint contiene un documento JSON. Per ulteriori informazioni, consulta Formati di messaggio e JSON Amazon SNS nella Guida per gli sviluppatori di Amazon Simple Notification Service.

È possibile configurare SNS in modo che le notifiche vengano inviate tramite messaggi di testo. Per ulteriori informazioni, consulta Messaggistica SMS nella Guida per sviluppatori di Amazon Simple Notification Service.

Per disattivare le notifiche senza eliminare una sottoscrizione, scegliere No per Enabled (Abilitato) nella console Amazon RDS. Oppure puoi impostare il `Enabled` parametro per `false` utilizzare l'API AWS CLI o Amazon RDS.

Fatturazione per le notifiche eventi Amazon RDS

Fatturazione per la notifica degli eventi Amazon RDS avviene tramite Amazon SNS. L'uso della notifica degli eventi è soggetta alle tariffe di Amazon SNS. Per ulteriori informazioni sulla fatturazione di Amazon SNS, consulta prezzi di Amazon Simple Notification Service.

Esempi di eventi RDS con Amazon EventBridgei

Negli esempi seguenti vengono illustrati diversi tipi di eventi Amazon RDS in formato JSON. Per un'esercitazione che illustra come acquisire e visualizzare eventi in formato JSON, consultare Tutorial: registra le modifiche allo stato delle istanze DB utilizzando Amazon EventBridge.

Argomenti

- Esempio di evento di istanza database
- Esempio di evento del gruppo parametri del database
- Esempio di evento snapshot DB

Esempio di evento di istanza database

Di seguito è riportato un esempio di evento di istanza database in formato JSON. L'evento mostra che RDS ha eseguito un failover Multi-AZ per l'istanza denominata `my-db-instance`. L'ID evento è RDS-EVENT-0049.

```
{
  "version": "0",
  "id": "68f6e973-1a0c-d37b-f2f2-94a7f62ffd4e",
  "detail-type": "RDS DB Instance Event",
  "source": "aws.rds",
  "account": "123456789012",
  "time": "2018-09-27T22:36:43Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:rds:us-east-1:123456789012:db:my-db-instance"
  ],
  "detail": {
    "EventCategories": [
      "failover"
    ],
    "SourceType": "DB_INSTANCE",
    "SourceArn": "arn:aws:rds:us-east-1:123456789012:db:my-db-instance",
    "Date": "2018-09-27T22:36:43.292Z",
    "Message": "A Multi-AZ failover has completed.",
    "SourceIdentifier": "my-db-instance",
    "EventID": "RDS-EVENT-0049"
  }
}
```

Esempio di evento del gruppo parametri del database

Di seguito è riportato un esempio di un evento gruppo parametri del database in formato JSON. L'evento mostra che il parametro `time_zone` è stato aggiornato nel gruppo di parametri `my-db-param-group`. L'ID evento è RDS-EVENT-0037.

```
{
  "version": "0",
  "id": "844e2571-85d4-695f-b930-0153b71dcb42",
  "detail-type": "RDS DB Parameter Group Event",
  "source": "aws.rds",
  "account": "123456789012",
  "time": "2018-10-06T12:26:13Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:rds:us-east-1:123456789012:pg:my-db-param-group"
  ],
  "detail": {
    "EventCategories": [
      "configuration change"
    ],
    "SourceType": "DB_PARAM",
    "SourceArn": "arn:aws:rds:us-east-1:123456789012:pg:my-db-param-group",
    "Date": "2018-10-06T12:26:13.882Z",
    "Message": "Updated parameter time_zone to UTC with apply method immediate",
    "SourceIdentifier": "my-db-param-group",
    "EventID": "RDS-EVENT-0037"
  }
}
```

Esempio di evento snapshot DB

Di seguito è riportato un esempio di un evento snapshot DB in formato JSON. L'evento mostra l'eliminazione della snapshot denominata `my-db-snapshot`. L'ID evento è RDS-EVENT-0041.

```
{
  "version": "0",
  "id": "844e2571-85d4-695f-b930-0153b71dcb42",
  "detail-type": "RDS DB Snapshot Event",
  "source": "aws.rds",
  "account": "123456789012",
  "time": "2018-10-06T12:26:13Z",
  "region": "us-east-1",
```

```
  "resources": [
    "arn:aws:rds:us-east-1:123456789012:snapshot:rds:my-db-snapshot"
  ],
  "detail": {
    "EventCategories": [
      "deletion"
    ],
    "SourceType": "SNAPSHOT",
    "SourceArn": "arn:aws:rds:us-east-1:123456789012:snapshot:rds:my-db-snapshot",
    "Date": "2018-10-06T12:26:13.882Z",
    "Message": "Deleted manual snapshot",
    "SourceIdentifier": "my-db-snapshot",
    "EventID": "RDS-EVENT-0041"
  }
}
```

Concessione di autorizzazioni per pubblicare le notifiche in un argomento Amazon SNS

Per concedere autorizzazioni Amazon RDS per pubblicare le notifiche in un argomento Servizio di notifica semplice Amazon (Amazon SNS), collega una policy (IAM) AWS Identity and Access Management all'argomento di destinazione. Per maggiori informazioni sulle autorizzazioni, consulta Esempi di casi per il controllo degli accessi Amazon SNS nella Guida per gli sviluppatori di Amazon Simple Notification Service.

Per impostazione predefinita, un argomento Amazon SNS dispone di una policy che consente a tutte le risorse Amazon RDS nello stesso account di pubblicare notifiche nello stesso. Puoi collegare una policy personalizzata per consentire notifiche tra account o per limitare l'accesso a determinate risorse.

Di seguito è riportato un esempio di policy IAM collegata all'argomento Amazon SNS di destinazione. Limita l'argomento alle istanze database con nomi corrispondenti al prefisso specificato. Per utilizzare questa policy, specifica i seguenti valori:

- `Resource` – Il nome della risorsa Amazon (ARN) per l'argomento Amazon SNS
- `SourceARN` – L'ARN della risorsa RDS
- `SourceAccount` – L'ID Account AWS

Per visualizzare un elenco di tipi di risorse e i relativi ARN, consulta Tipi di risorsa definiti da Amazon RDS in Service Authorization Reference.

```
{
  "Version": "2008-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "events.rds.amazonaws.com"
      },
      "Action": [
        "sns:Publish"
      ],
      "Resource": "arn:aws:sns:us-east-1:123456789012:topic_name",
      "Condition": {
        "ArnLike": {
          "aws:SourceArn": "arn:aws:rds:us-east-1:123456789012:db:prefix-*"
```

```
        },
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        }
      }
    }
  ]
}
```

Sottoscrizione alle notifiche eventi di Amazon RDS

Il modo più semplice per creare una sottoscrizione è utilizzare la console RDS. Se scegli di creare sottoscrizioni delle notifiche degli eventi tramite la CLI o l'API, devi creare un argomento Amazon Simple Notification Service e sottoscrivere l'argomento con la console Amazon SNS o l'API di Amazon SNS. Dovrai inoltre annotare l'Amazon Resource Name (ARN) dell'argomento, in quanto viene utilizzato quando si inviano comandi CLI o operazioni API. Per informazioni sulla creazione di un argomento SNS e sull'abbonamento allo stesso, consulta Nozioni di base su Amazon SNS nella Guida per sviluppatori di Amazon Simple Notification Service.

Puoi specificare il tipo di origine per cui vuoi ricevere le notifiche e l'origine Amazon RDS che attiva l'evento:

Source type (Tipo di origine)

Il tipo di origine Ad esempio: Source Type (Tipo di origine) potrebbe essere Instances (Istanze). Devi scegliere un tipo di origine.

Risorse da includere

La risorsa Amazon RDS che genera gli eventi. Ad esempio, puoi scegliere Select specific instances (Seleziona istanze specifiche) e quindi myDBInstance1.

Nella tabella seguente viene illustrato il risultato quando si specificano o non si specificano ***Risorse*** da includere.

Risorse da includere	Descrizione	Esempio
Specificato	RDS invia notifiche relative a tutti gli eventi solo per la risorsa specificata.	Se Source type (Tipo di origine) è Instances (Istanze) e la risorsa è myDBInstance1, RDS invia notifiche relative a tutti gli eventi solo per myDBInstance1.
Non specificato	RDS invia notifiche relative agli eventi relativi al tipo di origine specificato per tutte le risorse Amazon RDS.	Se Source type (Tipo di origine) è Instances (Istanze), RDS invia notifiche relative a tutti gli eventi correlati alle istanze nell'account.

L'abbonato di un argomento Amazon SNS riceve per impostazione predefinita ogni messaggio pubblicato nell'argomento. Per ricevere solo un sottoinsieme dei messaggi, l'abbonato deve assegnare una policy di filtro all'abbonamento all'argomento. Per ulteriori informazioni, consulta Filtraggio messaggi di Amazon SNS nella Guida per gli sviluppatori di Amazon Simple Notification Service.

Console

Per sottoscrivere una notifica eventi RDS

1. Accedi alla AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel pannello di navigazione, scegliere Event subscriptions (Sottoscrizioni a eventi).
3. Nel riquadro Event subscriptions (Sottoscrizioni di eventi) scegliere Create event subscription (Crea sottoscrizione di eventi).
4. Inserisci i dettagli dell'abbonamento come segue:

   a. Per Name (Nome), immettere un nome per la sottoscrizione alle notifiche eventi.

   b. Nel campo Send notifications to (Invia notifica a), esegui una delle seguenti operazioni:

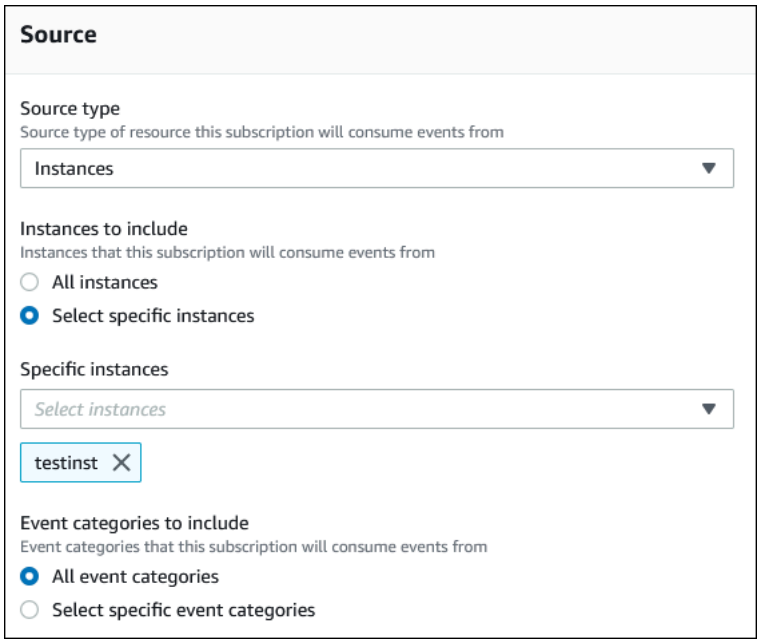
      - Scegli New email topic (Nuovo argomento e-mail). Inserisci un nome per l'argomento dell'email e un elenco di destinatari. Ti consigliamo di configurare le sottoscrizioni agli eventi con lo stesso indirizzo e-mail del contatto dell'account principale. I suggerimenti, gli eventi di assistenza e i messaggi personali vengono inviati utilizzando canali diversi. Le sottoscrizioni con lo stesso indirizzo e-mail assicurano che tutti i messaggi siano consolidati in un'unica posizione.
      - Scegli Amazon Resource Name (ARN) (Nome della risorsa Amazon (ARN)). Quindi scegli l'ARN Amazon SNS esistente per un argomento Amazon SNS.

        Se desideri utilizzare un argomento abilitato per la crittografia lato server (SSE), concedi ad Amazon RDS le autorizzazioni necessarie per accedere a AWS KMS key. Per ulteriori informazioni, consulta Abilitare la compatibilità tra le origini eventi dai servizi AWS e gli argomenti crittografati nella Guida per gli sviluppatori di Amazon Simple Notification Service.

   c. Per Source type (Tipo di origine) scegliere un tipo di origine. Ad esempio, scegli Instances (Istanze) o Parameter groups (Gruppi di parametri).

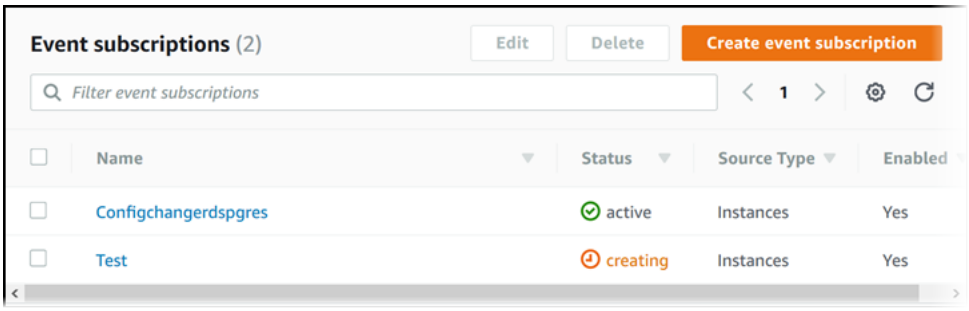
   d. Scegli le categorie di eventi e le risorse per i quali desideri ricevere notifiche eventi.

L'esempio seguente configura le notifiche degli eventi per l'istanza database denominata `testinst`.



e. Scegli Create (Crea).

La console Amazon RDS indica che è in corso la creazione della sottoscrizione.



AWS CLI

Per sottoscrivere notifiche degli eventi RDS, utilizzare il comando AWS CLI `create-event-subscription`. Includi i parametri obbligatori seguenti:

- `--subscription-name`
- `--sns-topic-arn`

Example

Per LinuxmacOS, oUnix:

```
aws rds create-event-subscription \
    --subscription-name myeventsubscription \
    --sns-topic-arn arn:aws:sns:us-east-1:123456789012:myawsuser-RDS \
    --enabled
```

Per Windows:

```
aws rds create-event-subscription ^
    --subscription-name myeventsubscription ^
    --sns-topic-arn arn:aws:sns:us-east-1:123456789012:myawsuser-RDS ^
    --enabled
```

API

Per sottoscrivere le notifiche degli eventi Amazon RDS, invoca la funzione API Amazon RDS `CreateEventSubscription`. Includi i parametri obbligatori seguenti:

- `SubscriptionName`
- `SnsTopicArn`

Tag e attributi delle notifiche eventi di Amazon RDS

Quando Amazon RDS invia una notifica di evento ad Amazon Simple Notification Service (SNS) o Amazon EventBridge, la notifica contiene gli attributi dei messaggi e i tag degli eventi. RDS invia gli attributi del messaggio separatamente insieme al messaggio, mentre i tag degli eventi sono inclusi nel corpo del messaggio. Usa gli attributi del messaggio e i tag di Amazon RDS per aggiungere metadati alle risorse. Puoi modificare questi tag con notazioni personalizzate relative alle istanze database . Per ulteriori informazioni sul tagging delle risorse di Amazon RDS, consulta Tagging delle risorse Amazon RDS.

Per impostazione predefinita, Amazon SNS e Amazon EventBridge ricevono tutti i messaggi a loro inviati. SNS ed EventBridge possono filtrare il messaggio e inviare le notifiche alla modalità di comunicazione preferita, ad esempio tramite e-mail, SMS o una chiamata a un endpoint HTTP.

> **Note**
>
> La notifica inviata tramite e-mail o SMS non includerà tag di evento.

La tabella seguente mostra gli attributi dei messaggi per gli eventi RDS inviati agli abbonati all'argomento.

Attributo per gli eventi Amazon RDS	Descrizione
EventID	Identificatore del messaggio dell'evento RDS, ad esempio RDS-EVENT-0006.
Risorsa	L'identificatore ARN della risorsa che emette l'evento, ad esempio `arn:aws:rds:ap-southeast-2:123456789012:db:database-1` .

I tag RDS forniscono i dati sulla risorsa interessata dall'evento del servizio. RDS aggiunge lo stato corrente dei tag nel corpo del messaggio quando la notifica viene inviata a SNS o EventBridge.

Per ulteriori informazioni sull'applicazioni di filtri agli attributi dei messaggi per SNS, consulta Filtraggio messaggi di Amazon SNS nella Guida per gli sviluppatori di Amazon Simple Notification Service.

Per ulteriori informazioni sull'applicazione di filtri ai tag degli eventi per EventBridge, consulta Filtraggio dei contenuti nei modelli di eventi di Amazon EventBridge nella Guida per l'utente di Amazon EventBridge.

Per ulteriori informazioni sull'applicazione di filtri ai tag basati sul payload per SNS, consulta https://aws.amazon.com/blogs/compute/introducing-payload-based-message-filtering-for-amazon-sns/.

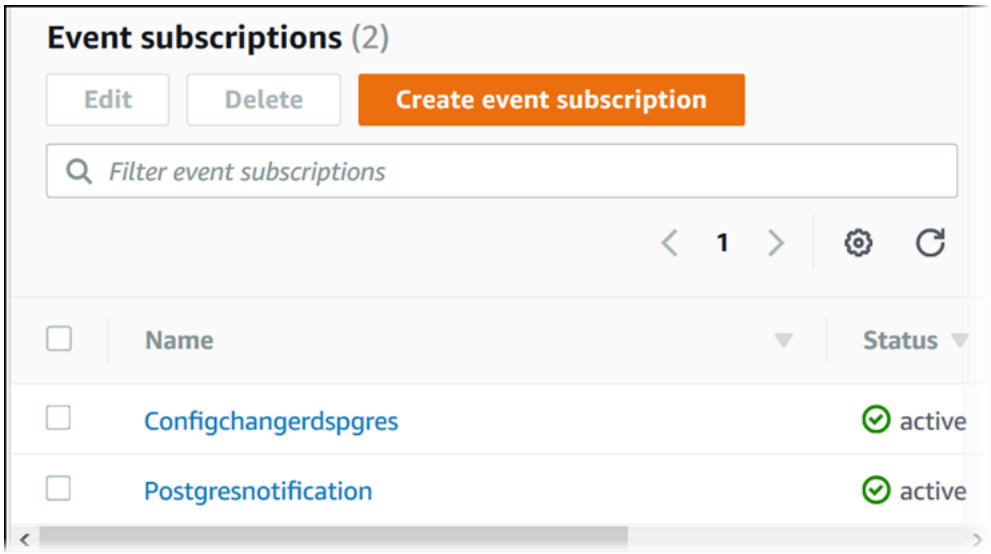
Elenco delle sottoscrizioni delle notifiche degli eventi Amazon RDS

Puoi creare un elenco delle sottoscrizioni correnti alle notifiche eventi Amazon RDS.

Console

Per creare un elenco delle sottoscrizioni correnti alle notifiche eventi Amazon RDS

1. Accedi alla AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel pannello di navigazione selezionare Event subscriptions (Sottoscrizioni di eventi). Il riquadro Event subscriptions (Sottoscrizioni di eventi) mostra tutte le sottoscrizioni delle notifiche degli eventi.



AWS CLI

Per visualizzare un elenco delle sottoscrizioni delle notifiche degli eventi Amazon RDS, utilizza il comando della AWS CLI `describe-event-subscriptions`.

Example

L'esempio seguente illustra tutte le sottoscrizioni a eventi.

```
aws rds describe-event-subscriptions
```

L'esempio seguente illustra `myfirsteventsubscription`.

```
aws rds describe-event-subscriptions --subscription-name myfirsteventsubscription
```

API

Per creare un elenco delle sottoscrizioni delle notifiche degli eventi Amazon RDS, chiamare l'operazione API Amazon RDS `DescribeEventSubscriptions`.

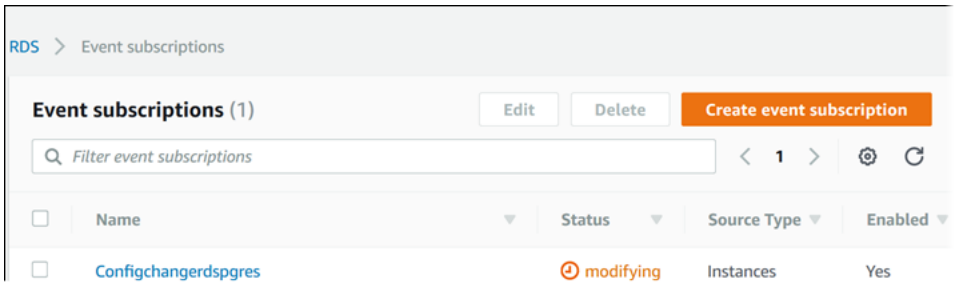
Modifica di una sottoscrizione alle notifiche eventi Amazon RDS

Dopo aver creato un abbonamento, puoi modificarne il nome, l'identificatore di origine, le categorie o l'ARN dell'argomento.

Console

Per modificare una sottoscrizione alle notifiche eventi Amazon RDS

1. Accedi alla AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel pannello di navigazione selezionare Event subscriptions (Sottoscrizioni di eventi).
3. Nel riquadro Event subscriptions (Sottoscrizioni di eventi) scegliere la sottoscrizione da modificare e selezionare Edit (Modifica).
4. Apportare le modifiche alla sottoscrizione nella sezione Target (Destinazione) o Source (Origine).
5. Seleziona Edit (Modifica). La console Amazon RDS indica che è in corso la modifica della sottoscrizione.



AWS CLI

Per modificare una sottoscrizione delle notifiche degli eventi Amazon RDS, utilizza il comando della AWS CLI `modify-event-subscription`. Includi il seguente parametro obbligatorio:

- `--subscription-name`

Example

Il codice seguente abilita `myeventsubscription`.

Per LinuxmacOS, oUnix:

```
aws rds modify-event-subscription \
    --subscription-name myeventsubscription \
    --enabled
```

Per Windows:

```
aws rds modify-event-subscription ^
    --subscription-name myeventsubscription ^
    --enabled
```

API

Per modificare un evento Amazon RDS, chiamare l'operazione API Amazon RDS `ModifyEventSubscription`. Includi il seguente parametro obbligatorio:

- `SubscriptionName`

Aggiunta di un identificatore di origine a una sottoscrizione alle notifiche eventi Amazon RDS

Puoi aggiungere un identificatore di origine Amazon RDS che genera l'evento a una sottoscrizione esistente.

Console

Puoi aggiungere o rimuovere facilmente gli identificatori di origine tramite la console Amazon RDS, selezionandoli o deselezionandoli quando modifichi una sottoscrizione. Per ulteriori informazioni, consulta Modifica di una sottoscrizione alle notifiche eventi Amazon RDS.

AWS CLI

Per aggiungere un identificatore di origine a una sottoscrizione delle notifiche degli eventi Amazon RDS, utilizza il comando della AWS CLI `add-source-identifier-to-subscription`. Includi i parametri obbligatori seguenti:

- `--subscription-name`
- `--source-identifier`

Example

L'esempio seguente aggiunge l'identificatore di origine `mysqldb` alla sottoscrizione `myrdseventsubscription`.

Per LinuxmacOS, oUnix:

```
aws rds add-source-identifier-to-subscription \
    --subscription-name myrdseventsubscription \
    --source-identifier mysqldb
```

Per Windows:

```
aws rds add-source-identifier-to-subscription ^
    --subscription-name myrdseventsubscription ^
    --source-identifier mysqldb
```

API

Per aggiungere un identificatore di origine a una sottoscrizione delle notifiche degli eventi Amazon RDS, chiamare l'operazione API Amazon RDS `AddSourceIdentifierToSubscription`. Includi i parametri obbligatori seguenti:

- `SubscriptionName`
- `SourceIdentifier`

Rimozione di un identificatore di origine da una sottoscrizione alle notifiche eventi Amazon RDS

Per smettere di ricevere notifiche relative agli eventi di un'origine, puoi rimuovere un identificatore di origine Amazon RDS che genera l'evento da una sottoscrizione.

Console

Puoi aggiungere o rimuovere facilmente gli identificatori di origine tramite la console Amazon RDS, selezionandoli o deselezionandoli quando modifichi una sottoscrizione. Per ulteriori informazioni, consulta Modifica di una sottoscrizione alle notifiche eventi Amazon RDS.

AWS CLI

Per rimuovere un identificatore di origine da una sottoscrizione delle notifiche degli eventi Amazon RDS, utilizza il comando della AWS CLI `remove-source-identifier-from-subscription`. Includi i parametri obbligatori seguenti:

- `--subscription-name`
- `--source-identifier`

Example

L'esempio seguente rimuove l'identificatore dell'origine `mysqldb` dalla sottoscrizione `myrdseventsubscription`.

Per LinuxmacOS, oUnix:

```
aws rds remove-source-identifier-from-subscription \
    --subscription-name myrdseventsubscription \
    --source-identifier mysqldb
```

Per Windows:

```
aws rds remove-source-identifier-from-subscription ^
    --subscription-name myrdseventsubscription ^
    --source-identifier mysqldb
```

API

Per rimuovere un identificatore di origine da una sottoscrizione delle notifiche degli eventi Amazon RDS, utilizzare l'operazione API Amazon RDS `RemoveSourceIdentifierFromSubscription`. Includi i parametri obbligatori seguenti:

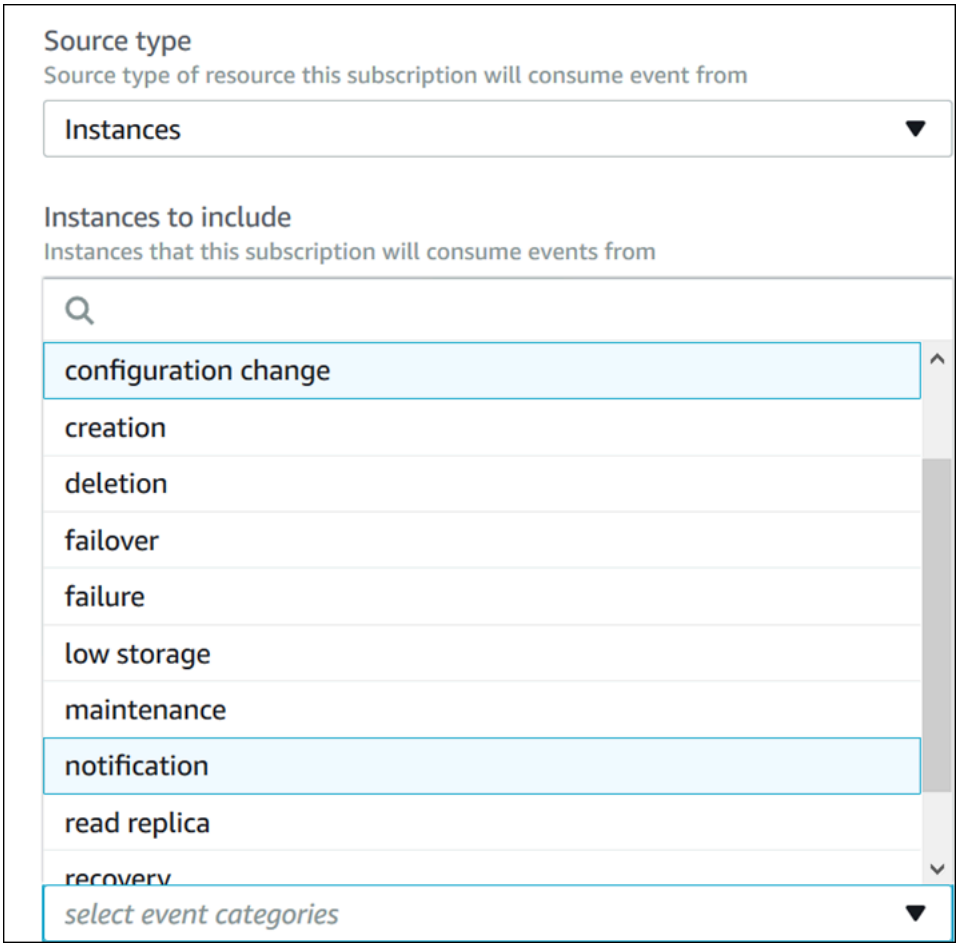
- `SubscriptionName`
- `SourceIdentifier`

Creazione di un elenco delle categorie di notifiche eventi Amazon RDS

Tutti gli eventi per un tipo di risorsa sono raggruppati in categorie. Per visualizzare l'elenco delle categorie disponibili, utilizza le procedure riportate di seguito.

Console

Quando crei o modifichi una sottoscrizione alle notifiche eventi, le categorie di eventi vengono visualizzate nella console Amazon RDS. Per ulteriori informazioni, consulta Modifica di una sottoscrizione alle notifiche eventi Amazon RDS.



AWS CLI

Per visualizzare un elenco delle categorie delle notifiche degli eventi Amazon RDS, utilizza il comando della AWS CLI `describe-event-categories`. Questo comando non prevede parametri obbligatori.

Example

```
aws rds describe-event-categories
```

API

Per elencare le categorie delle notifiche degli eventi Amazon RDS, utilizzare l'operazione API Amazon RDS `DescribeEventCategories`. Questo comando non prevede parametri obbligatori.

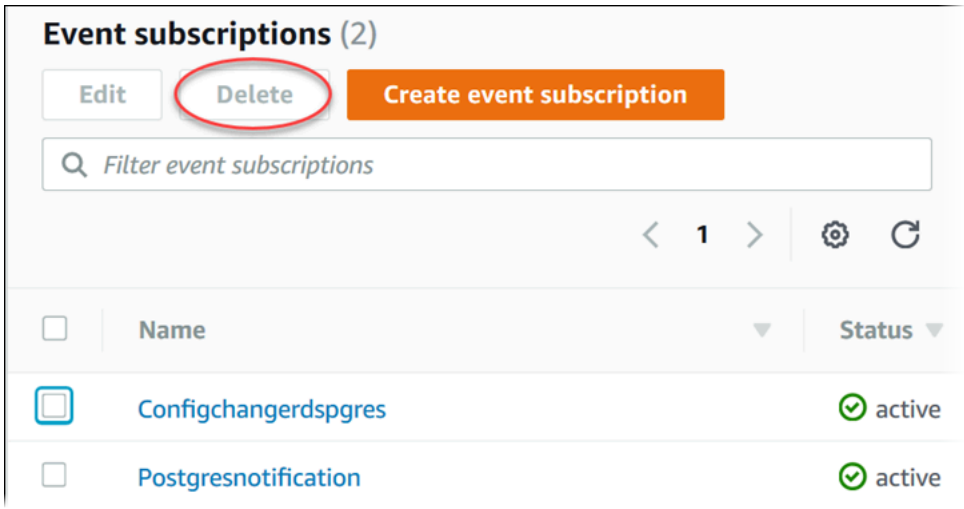
Eliminazione di una sottoscrizione alle notifiche eventi Amazon RDS

Puoi eliminare un abbonamento quando questo non è più necessario. Tutti gli abbonati all'argomento non riceveranno più le notifiche di eventi specificate dall'abbonamento.

Console

Per eliminare una sottoscrizione alle notifiche eventi Amazon RDS

1. Accedi alla AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel pannello di navigazione scegliere DB Event Subscriptions (Sottoscrizioni di eventi database).
3. Nel riquadro My DB Event Subscriptions (Sottoscrizioni di eventi database personali) scegliere la sottoscrizione che si vuole eliminare.
4. Scegli Delete (Elimina).
5. La console Amazon RDS indica che è in corso l'eliminazione della sottoscrizione.



AWS CLI

Per eliminare una sottoscrizione delle notifiche degli eventi Amazon RDS, utilizza il comando AWS CLI `delete-event-subscription`. Includi il seguente parametro obbligatorio:

- `--subscription-name`

Example

L'esempio seguente elimina la sottoscrizione `myrdssubscription`.

```
aws rds delete-event-subscription --subscription-name myrdssubscription
```

API

Per eliminare una sottoscrizione delle notifiche degli eventi Amazon RDS, utilizzare l'operazione API RDS `DeleteEventSubscription`. Includi il seguente parametro obbligatorio:

- `SubscriptionName`

Creazione di una regola che si attiva su un evento Amazon RDS

Con Amazon EventBridge, puoi automatizzare AWS i servizi e rispondere a eventi di sistema come problemi di disponibilità delle applicazioni o modifiche delle risorse.

Argomenti

Creazione di regole per inviare eventi Amazon RDS ad Amazon EventBridge

Puoi scrivere semplici regole che indichino quali eventi Amazon RDS ti interessano e quali operazioni automatizzate eseguire quando si verifica un evento previsto da una regola. Puoi impostare una varietà di obiettivi, come una AWS Lambda funzione o un argomento Amazon SNS, che ricevono eventi in formato JSON. Ad esempio, puoi configurare Amazon RDS Amazon ogni volta che viene creata o eliminata un'istanza DB. Per ulteriori informazioni, consulta la Amazon CloudWatch Events User Guide e la Amazon EventBridge User Guide.

Per creare una regola che si attiva su un evento RDS:

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. In Events (Eventi) nel pannello di navigazione, scegli Rules (Regole).
3. Scegli Create rule (Crea regola).
4. Per Event Source (Origine evento) procedi nel seguente modo:
    a. Seleziona Event Pattern (Modello di eventi).
    b. Per Service Name (Nome servizio), scegli Relational Database Service (RDS).
    c. Per Event Type (Tipo di evento), scegli il tipo di risorsa Amazon RDS che attiva l'evento. Ad esempio, se un'istanza database attiva l'evento, scegli RDS DB Instance Event (Evento istanza database RDS).
5. Per Targets, scegli Add Target e scegli il AWS servizio che deve agire quando viene rilevato un evento del tipo selezionato.
6. Negli altri campi di questa sezione, inserisci informazioni specifiche di questo tipo di destinazione, se necessarie.

7. Per molti tipi di oggetto, EventBridge sono necessarie le autorizzazioni per inviare eventi alla destinazione. In questi casi, EventBridge puoi creare il ruolo IAM necessario per l'esecuzione del tuo evento:

   - Per creare un ruolo IAM automaticamente, seleziona Create a new role for this specific resource (Crea un nuovo ruolo per questa risorsa specifica).
   - Per utilizzare un ruolo IAM creato in precedenza, seleziona Use existing role (Utilizza un ruolo esistente).

8. Facoltativamente, ripeti le fasi 5-7 per aggiungere un'altra destinazione per questa regola
9. Scegli Configure details (Configura dettagli). In Rule definition (Definizione regola), digita un nome e una descrizione della regola.

   Il nome della regola deve essere univoco all'interno di questa regione.
10. Scegli Create rule (Crea regola).

Per ulteriori informazioni, consulta Creazione di una EventBridge regola che si attiva su un evento nella Amazon CloudWatch User Guide.

Tutorial: registra le modifiche allo stato delle istanze DB utilizzando Amazon EventBridge

In questo tutorial, crei una AWS Lambda funzione che registra le modifiche di stato per un'istanza Amazon RDS. Successivamente crei una regola che esegua la funzione ogni volta che si verifica un cambiamento di stato di un'istanza database RDS esistente. Il tutorial presuppone che si dispone di una piccola istanza di test in esecuzione che è possibile arrestare temporaneamente.

> **Important**
>
> Non eseguire questo tutorial su un'istanza database di produzione in esecuzione.

Argomenti

- Passaggio 1: creare una funzione AWS Lambda
- Fase 2: Creazione di una regola
- Fase 3: Test della regola

Passaggio 1: creare una funzione AWS Lambda

Crea una funzione Lambda per registrare gli eventi di modifica dello stato. È necessario specificare questa funzione alla creazione della regola.

Per creare una funzione Lambda

1. Apri la AWS Lambda console all'indirizzo https://console.aws.amazon.com/lambda/.
2. Se è la prima volta che utilizzi Lambda, verrà visualizzata una pagina di benvenuto. Selezionare Get Started Now (Inizia subito). Altrimenti, scegliere Create function (Crea funzione).
3. Scegli Author from scratch (Crea da zero).
4. Nella pagina Create function (Crea funzione), procedere come segue:
   a. Digitare un nome e una descrizione per la funzione Lambda. Ad esempio, denomina la funzione **`RDSInstanceStateChange`**.
   b. In Runtime, seleziona Node.js 16x.
   c. In Architecture (Architettura), scegli x86_64.
   d. In Execution role (Ruolo di esecuzione), esegui una delle operazioni seguenti:
      - Scegliere Create a new role with basic Lambda permissions (Crea un nuovo ruolo con le autorizzazioni Lambda di base).
      - In Execution role (Ruolo di esecuzione), sceglie Use an existing role (Utilizza un ruolo esistente). Scegli il ruolo che desideri usare.
   e. Scegli Crea funzione.
5. Nella InstanceStateChange pagina RDS, procedi come segue:
   a. In Origine codice, seleziona index.js.
   b. Nel riquadro di index.js, elimina il codice esistente.
   c. Immetti il seguente codice:

```
console.log('Loading function');

exports.handler = async (event, context) => {
    console.log('Received event:', JSON.stringify(event));
};
```

   d. Selezionare Deploy (Distribuisci).

Fase 2: Creazione di una regola

Crea una regola per l'esecuzione della funzione Lambda ogni volta che avvii un'istanza Amazon RDS.

Per creare la regola EventBridge

1. Apri la EventBridge console Amazon all'indirizzo https://console.aws.amazon.com/events/.
2. Nel pannello di navigazione, scegli Regole.
3. Scegli Create rule (Crea regola).
4. Immettere un nome e una descrizione per la regola. Ad esempio, specifica **`RDSInstanceStateChangeRule`**.
5. Scegli Rule with an event pattern (Regola con un modello di eventi), quindi seleziona Next (Successivo).
6. Per Event source, scegli AWS eventi o eventi EventBridge partner.
7. Scorri verso il basso fino alla sezione Event pattern (Modello di eventi).
8. In Event source (Origine eventi), selezionare Servizi AWS.
9. In AWS service (Servizio AWS), scegli Relational Database Service (RDS).
10. Per Tipo di evento, seleziona Evento istanza database RDS.
11. Lascia il modello di eventi predefinito. Quindi scegli Successivo.
12. Per Target types (Tipi di destinazione), scegli AWS service (Servizio).
13. Per Select a target (Seleziona destinazione), scegli Lambda function (Funzione Lambda).
14. In Function (Funzione), seleziona la funzione Lambda che hai creato. Quindi scegli Successivo.
15. In Configure tags (Configura tag), scegli Next (Successivo).
16. Esamina i passaggi nella regola. Quindi scegli Create rule (Crea regola).

Fase 3: Test della regola

Per verificare la regola, arresta un'istanza database RDS. Dopo aver atteso alcuni minuti perché l'istanza venga avviata e inizializzata, verifica che la funzione Lambda sia stata richiamata.

Per effettuare il test della regola arrestando un'istanza database

1. Apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Arresta un'istanza database RDS.

3. Apri la EventBridge console Amazon all'indirizzo https://console.aws.amazon.com/events/.
4. Nel pannello di navigazione, seleziona Regole, scegli il nome della regola creata.
5. In Dettagli della regola scegli Monitoraggio.

   Verrai reindirizzato alla CloudWatch console Amazon. Se non vieni reindirizzato, fai clic su Visualizza le metriche in. CloudWatch
6. In Tutti i parametri, seleziona il nome della regola creata.

   Il grafico deve indicare che la regola è stata richiamata.
7. Nel pannello di navigazione, selezionare Log groups (Gruppi di log).
8. Scegli il nome del gruppo di log per la funzione Lambda (/aws/lambda/***nome-funzione***).
9. Scegliere il nome del flusso di log per visualizzare i dati forniti dalla funzione per l'istanza avviata. Sarà visualizzato un evento ricevuto simile a quello seguente:

```
{
    "version": "0",
    "id": "12a345b6-78c9-01d2-34e5-123f4ghi5j6k",
    "detail-type": "RDS DB Instance Event",
    "source": "aws.rds",
    "account": "111111111111",
    "time": "2021-03-19T19:34:09Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:rds:us-east-1:111111111111:db:testdb"
    ],
    "detail": {
        "EventCategories": [
            "notification"
        ],
        "SourceType": "DB_INSTANCE",
        "SourceArn": "arn:aws:rds:us-east-1:111111111111:db:testdb",
        "Date": "2021-03-19T19:34:09.293Z",
        "Message": "DB instance stopped",
        "SourceIdentifier": "testdb",
        "EventID": "RDS-EVENT-0087"
    }
}
```

Per altri esempi di eventi RDS in formato JSON, vedere Panoramica degli eventi per Amazon RDS.

10. (Facoltativo) Al termine, puoi aprire la console Amazon RDS e avviare l'istanza terminata.

Categorie di eventi Amazon RDS e messaggi di evento

Amazon RDS genera un numero significativo di eventi in categorie a cui puoi abbonarti utilizzando la console Amazon RDS o AWS CLI l'API.

Argomenti

Eventi di cluster di database

La tabella seguente riporta la categoria di eventi e un elenco di eventi applicabili quando il tipo di origine è un cluster database.

Per ulteriori informazioni sulle implementazioni di cluster Multi-AZ DB, consulta. Implementazioni cluster di database multi-AZ

Categoria	ID evento RDS	Messaggio	Note
modifica della configurazione	RDS-EVENT-0016	Reimpostare le credenziali master.	
creazione	RDS-EVENT-0170	Cluster di database creato.	
failover	RDS-EVENT-0069	Failover del cluster non riuscito. Verificare lo stato delle istanze del cluster e riprovare.	

Categoria	ID evento RDS	Messaggio	Note
failover	RDS-EVENT-0070	Nuova promozione del cluster primario precedente: *nome*.	
failover	RDS-EVENT-0071	Failover sull'istanza database completato: *nome*.	
failover	RDS-EVENT-0072	Failover stessa AZ avviato sull'istanza database: *nome*.	
failover	RDS-EVENT-0073	Failover multi-AZ avviato sull'istanza database: *nome*.	
errore	RDS-EVENT-0354	Non è possibile creare il cluster DB a causa di risorse incompatibili. *messaggio* .	*messaggio* include dettagli sull'operazione non riuscita.
errore	RDS-EVENT-0355	Il cluster DB non può essere creato a causa di limiti di risorse insufficienti. *messaggio* .	*messaggio* include dettagli sull'operazione non riuscita.

Categoria	ID evento RDS	Messaggio	Note
Failover globale	RDS-EVENT-0181	Switchover globale sul cluster database *nome* nella regione *nome* avviato.	Questo evento riguarda un'operazione di switchover (precedentemente denominata "failover pianificato gestito"). Il processo può essere ritardato perché altre operazioni sono in esecuzione sul cluster di database.
Failover globale	RDS-EVENT-0182	Arresto del cluster database primario precedente *nome* nella regione *nome* riuscito.	Questo evento riguarda un'operazione di switchover (precedentemente denominata "failover pianificato gestito"). La vecchia istanza principale nel database globale non accetta scritture. Tutti i volumi sono sincronizzati.
Failover globale	RDS-EVENT-0183	In attesa della sincronizzazione dei dati tra i membri del cluster globale. Ritardi attuali del cluster database primario: *motivo*.	Questo evento riguarda un'operazione di switchover (precedentemente denominata "failover pianificato gestito"). Si verifica un ritardo di replica durante la fase di sincronizzazione del failover globale del database.

Categoria	ID evento RDS	Messaggio	Note
Failover globale	RDS-EVENT-0184	Promozione del nuovo cluster database primario *nome* nella regione *nome* riuscita.	Questo evento riguarda un'operazione di switchover (precedentemente denominata "failover pianificato gestito"). La topologia del volume del database globale viene ristabilita con il nuovo volume primario.
Failover globale	RDS-EVENT-0185	Switchover globale sul cluster database *nome* nella regione *nome* terminato.	Questo evento riguarda un'operazione di switchover (precedentemente denominata "failover pianificato gestito"). Lo switchover globale del database è stato completato nel cluster database primario. Le repliche potrebbero richiedere molto tempo per arrivare online dopo il completamento del failover.
Failover globale	RDS-EVENT-0186	Switchover globale sul cluster database *nome* nella regione *nome* annullato.	Questo evento riguarda un'operazione di switchover (precedentemente denominata "failover pianificato gestito").

Categoria	ID evento RDS	Messaggio	Note
Failover globale	RDS-EVENT-0187	Switchover globale sul cluster database *nome* nella regione *nome* non riuscito.	Questo evento riguarda un'operazione di switchover (precedentemente denominata "failover pianificato gestito").
Failover globale	RDS-EVENT-0238	Failover globale sul cluster database *nome* nella regione *nome* completato.	
Failover globale	RDS-EVENT-0239	Failover globale sul cluster database *nome* nella regione *nome* non riuscito.	
Failover globale	RDS-EVENT-0240	Risincronizzazione dei membri del cluster database *nome* nella regione *nome* dopo il failover globale avviata.	
Failover globale	RDS-EVENT-0241	Risincronizzazione dei membri del cluster database *nome* nella regione *nome* dopo il failover globale terminata.	
manutenzione	RDS-EVENT-0156	È disponibile un aggiornamento della versione secondaria del motore del database per il cluster di database.	
manutenzione	RDS-EVENT-0176	La versione principale del motore del cluster database è stata aggiornata.	

Categoria	ID evento RDS	Messaggio	Note
manutenzione	RDS-EVENT-0286	Aggiornamento della versione del motore del cluster database avviato.	
manutenzione	RDS-EVENT-0287	Requisito di aggiornamento del sistema operativo rilevato.	
manutenzione	RDS-EVENT-0288	Avvio dell'aggiornamento del sistema operativo del cluster in corso.	
manutenzione	RDS-EVENT-0289	Aggiornamento del sistema operativo del cluster completato.	
manutenzione	RDS-EVENT-0290	Applicazione delle patch al cluster database completata: versione di origine *numero_versione* => *numero_nuova_versione* .	
notification	RDS-EVENT-0172	Cluster rinominato da *nome* in *nome*.	

Eventi di istanza database

La tabella seguente riporta la categoria di eventi e un elenco di eventi applicabili quando il tipo di origine è un'istanza database.

Categoria	ID evento RDS	Messaggio	Note
disponibilità	RDS-EVENT-0004	L'istanza database è stata arrestata.	

Categoria	ID evento RDS	Messaggio	Note
availability	RDS-EVENT-0006	L'istanza database è stata riavviata.	
disponibilità	RDS-EVENT-0022	Errore durante il riavvio di mysql: *messaggio* .	Si è verificato un errore durante il riavvio di MySQL.
disponibilità	RDS-EVENT-0221	L'istanza database ha raggiunto la soglia di completamento dell'archiviazione e il database è stato arrestato. È possibile aumentare lo spazio di archiviazione allocato per risolvere questo problema.	
disponibilità	RDS-EVENT-0222	Lo spazio di archiviazione libero per l'istanza database *nome* è basso rispetto al valore *percentuale* dell'archiviazione allocata (archiviazione allocata *valore*, spazio di archiviazione libero: *valore*]. Il database verrà arrestato per evitare il danneggiamento se lo spazio di archiviazione libero è inferiore a *valore*. È possibile aumentare lo spazio di archiviazione allocato per risolvere questo problema.	Per ulteriori informazioni, consulta Storage delle istanze di database Amazon RDS.

Categoria	ID evento RDS	Messaggio	Note
disponibilità	RDS-EVENT-0330	*La capacità di archiviazione gratuita del volume dedicato del registro delle transazioni è troppo bassa per il nome dell'istanza DB.* Lo spazio di archiviazione gratuito del volume di registro è la *percentuale* dello spazio di archiviazione allocato. *[Archiviazione allocata: quantità, Spazio di archiviazione gratuito: quantità] Il database verrà chiuso per evitare il danneggiamento se lo spazio di archiviazione gratuito è inferiore alla quantità.* È possibile disabilitare il volume dedicato del registro delle transazioni per risolvere questo problema.	Per ulteriori informazioni, consulta Volume di registro dedicato (DLV).

Categoria	ID evento RDS	Messaggio	Note
disponibilità	RDS-EVENT-0331	*La capacità di archiviazione gratuita del volume dedicato del registro delle transazioni è troppo bassa per il nome dell'istanza DB.* Lo spazio di archiviazione gratuito per il volume di log è la *percentuale* dello storage assegnato. [Storage fornito: *importo*, spazio di archiviazione gratuito: *importo*] È possibile disabilitare il volume dedicato del registro delle transazioni per risolvere questo problema.	Per ulteriori informazioni, consulta Volume di registro dedicato (DLV).
backup	RDS-EVENT-0001	Viene eseguito il backup dell'istanza database.	
backup	RDS-EVENT-0002	Il backup dell'istanza database è terminato.	

Categoria	ID evento RDS	Messaggio	Note
backup	RDS-EVENT-0086	Impossibile associare il gruppo di opzioni *nome* all'istanza database *nome*. Verificare che il gruppo di opzioni *nome* sia supportato nella classe di istanza database e nella configurazione. In caso affermativo, verificare tutte le impostazioni del gruppo di opzioni e riprovare.	Per ulteriori informazioni, consulta Uso di gruppi di opzioni.
modifica della configurazione	RDS-EVENT-0011	Aggiornato per utilizzare il ParameterGroup *nome del* DB.	
modifica della configurazione	RDS-EVENT-0012	Vengono applicate le modifiche alla classe di istanza database.	
modifica della configurazione	RDS-EVENT-0014	Applicazione delle modifiche alla classe di istanza database completata.	
modifica della configurazione	RDS-EVENT-0016	Reimpostare le credenziali master.	
modifica della configurazione	RDS-EVENT-0017	Applicazione delle modifiche all'archiviazione allocata completata.	

Categoria	ID evento RDS	Messaggio	Note
modifica della configurazione	RDS-EVENT-0018	Applicazione delle modifiche allo spazio di archiviazione allocato in corso.	
modifica della configurazione	RDS-EVENT-0024	Applicazione di modifiche per la conversione in un'istanza database Multi-AZ in corso.	
modifica della configurazione	RDS-EVENT-0025	Applicazione delle modifiche per la conversione in un'istanza database Multi-AZ completata.	
modifica della configurazione	RDS-EVENT-0028	Backup automatici disabilitati.	
modifica della configurazione	RDS-EVENT-0029	Applicazione delle modifiche per la conversione in un'istanza database standard (Single-AZ) completata.	
modifica della configurazione	RDS-EVENT-0030	Applicazione delle modifiche per la conversione in un'istanza database standard (Single-AZ) in corso.	
modifica della configurazione	RDS-EVENT-0032	Backup automatici abilitati.	

Categoria	ID evento RDS	Messaggio	Note
modifica della configurazione	RDS-EVENT-0033	Ci sono *numero* utenti che corrispondono al nome utente master. Reimpostazione dell'unico nome utente non collegato a un host specifico in corso.	
modifica della configurazione	RDS-EVENT-0067	Impossibile reimpostare la password. Informazioni sull'errore: *messaggio* .	
modifica della configurazione	RDS-EVENT-0078	Intervallo di monitoraggio modificato in *numero*.	La configurazione Enhanced Monitoring è stata modificata.
modifica della configurazione	RDS-EVENT-0092	Aggiornamento gruppo di parametri DB terminato.	
modifica della configurazione	RDS-EVENT-0217	Applicazione di modifiche avviate dal ridimensionamento automatico allo storage allocato.	
modifica della configurazione	RDS-EVENT-0218	Terminata l'applicazione della modifica avviata dal ridimensionamento automatico allo storage allocato.	
modifica della configurazione	RDS-EVENT-0295	L'aggiornamento della configurazione dell'archiviazione è iniziato.	

Categoria	ID evento RDS	Messaggio	Note
modifica della configurazione	RDS-EVENT-0296	L'aggiornamento della configurazione dell'archiviazione è completato.	
modifica della configurazione	RDS-EVENT-0332	Il volume di registro dedicato è disabilitato.	Per ulteriori informazioni, consulta Volume di registro dedicato (DLV).
modifica della configurazione	RDS-EVENT-0333	La disabilitazione del volume di registro dedicato è iniziata.	Per ulteriori informazioni, consulta Volume di registro dedicato (DLV).
modifica della configurazione	RDS-EVENT-0334	L'attivazione del volume di registro dedicato è iniziata.	Per ulteriori informazioni, consulta Volume di registro dedicato (DLV).
modifica della configurazione	RDS-EVENT-0335	Il volume di registro dedicato è abilitato.	Per ulteriori informazioni, consulta Volume di registro dedicato (DLV).
creazione	RDS-EVENT-0005	Viene creata l'istanza database.	
eliminazione	RDS-EVENT-0003	Istanza database eliminata.	
failover	RDS-EVENT-0013	Failover dell'istanza Multi-AZ avviato.	È stato avviato un failover Multi-AZ che ha comportato la promozione di un'istanza database in standby.

Categoria	ID evento RDS	Messaggio	Note
failover	RDS-EVENT-0015	Failover da Multi-AZ a standby completato: la propagazione DNS può richiedere alcuni minuti.	È stato completato un failover Multi-AZ che ha comportato la promozione di un'istanza database in standby. Potrebbero essere necessari alcuni minuti per il trasferimento del DNS alla nuova istanza database principale.
failover	RDS-EVENT-0034	Abbandono in corso del failover richiesto dall'utente perché di recente si è verificato un failover nell'istanza database.	Amazon RDS non sta tentando di effettuare un failover richiesto perché di recente si è verificato un failover nell'istanza database.
failover	RDS-EVENT-0049	Failover dell'istanza Multi-AZ completato.	
failover	RDS-EVENT-0050	Attivazione dell'istanza Multi-AZ avviata.	Un'attivazione Multi-AZ è iniziata dopo il corretto ripristino dell'istanza DB.
failover	RDS-EVENT-0051	Attivazione dell'istanza Multi-AZ completata.	È stata completata un'attivazione Multi-AZ. Ora il database dovrebbe essere accessibile.
failover	RDS-EVENT-0065	Ripristino riuscito da un failover parziale.	

Categoria	ID evento RDS	Messaggio	Note
errore	RDS-EVENT-0031	Istanza database impostata sullo stato *nome*. RDS consiglia di avviare un. point-in-time-restore	L'istanza database ha avuto esito negativo a causa di una configura zione non compatibile o di un problema di storage sottostante. Inizia a point-in-time-restore per l'istanza DB.
errore	RDS-EVENT-0035	Istanza database impostata sullo stato *nome*. *messaggio* .	L'istanza database include parametri non validi. Ad esempio, se non è stato possibile avviare l'istanza database perché un parametro relativo alla memoria è troppo elevato per questa classe di istanze, è necessario modificare il parametro relativo alla memoria e riavviare l'istanza database.
errore	RDS-EVENT-0036	Lo stato dell'istanza database è *stato*. *messaggio* .	L'istanza database si trova in una rete non compatibi le. Alcuni degli ID sottorete specificati non sono validi o non esistono.
errore	RDS-EVENT-0058	Installazione di Statspack non riuscita. *messaggio* .	Errore durante la creazione dell'account utente Oracle Statspack `PERFSTAT`. Eliminare l'account prima di aggiungere l'opzione `STATSPACK` .

Categoria	ID evento RDS	Messaggio	Note
errore	RDS-EVENT-0079	Amazon RDS non è stato in grado di creare le credenziali per un monitoraggio avanzato e questa funzionalità è stata disattivata. Ciò è probabilmente dovuto al fatto che rds-monitoring-role non è presente e non è configurato correttamente nel tuo account. Per ulteriori dettagli, consulta la sezione relativa alla risoluzione dei problemi nella documentazione di Amazon RDS.	Impossibile abilitare la funzionalità Monitoraggio avanzato senza il ruolo IAM di monitoraggio avanzato. Per ulteriori informazioni sulla creazione del ruolo IAM, consulta Per creare un ruolo IAM per Amazon RDS Enhanced Monitoring.
errore	RDS-EVENT-0080	Amazon RDS non è stato in grado di configurare il monitoraggio avanzato nell'istanza: *nome* e questa funzionalità è stata disattivata. Ciò è probabilmente dovuto al fatto che rds-monitoring-role non è presente e non è configurato correttamente nel tuo account. Per ulteriori dettagli, consulta la sezione relativa alla risoluzione dei problemi nella documentazione di Amazon RDS.	La funzionalità Monitoraggio avanzato è stata disabilitata a causa di un errore che ha provocato la modifica della configurazione. È possibile che il ruolo IAM di monitoraggio avanzato non sia configurato correttamente. Per informazioni sulla creazione del ruolo IAM di monitoraggio avanzato, consulta Per creare un ruolo IAM per Amazon RDS Enhanced Monitoring.

Categoria	ID evento RDS	Messaggio	Note
errore	RDS-EVENT-0081	Amazon RDS non è stato in grado di creare le credenziali per l'opzione *nome*. Ciò è dovuto al fatto che il ruolo IAM *nome* non è configurato correttam ente nel tuo account. Per ulteriori dettagli, consulta la sezione relativa alla risoluzione dei problemi nella documentazione di Amazon RDS.	Il ruolo IAM utilizzato per accedere al bucket di Amazon S3 per il backup e ripristino nativo di SQL Server non è configurato correttamente. Per ulteriori informazioni, consulta Configurazione di backup e ripristino nativi.
errore	RDS-EVENT-0165	L'istanza RDS Custom DB si trova al di fuori del perimetro di supporto.	È tua responsabilità risolvere i problemi di configurazione che portano l'istanza database di RDS Custom nello stato `unsupported-configuration` . Se il problema riguarda l' AWS infrastruttura, puoi utilizzare la console o AWS CLI risolverlo. Se il problema riguarda il sistema operativo o la configurazione del database, è possibile accedere all'host per risolverlo. Per ulteriori informazi oni, consulta Perimetro di supporto RDS Custom.

Categoria	ID evento RDS	Messaggio	Note
errore	RDS-EVENT-0188	Lo stato dell'istanza database non può essere aggiornato. *messaggio*	Amazon RDS non è riuscito ad aggiornare un'istanza database MySQL dalla versione 5.7 alla versione 8.0 a causa di incompatibilità correlate al dizionario dati. L'istanza database è stata ripristinata a MySQL versione 5.7. Per ulteriori informazioni, consulta Rollback dopo l'errore di aggiornamento da MySQL 5.7 a 8.0.
errore	RDS-EVENT-0219	Lo stato dell'istanza database non è valido. Nessuna operazione richiesta. Il ridimensionamento automatico verrà riprovato in un secondo momento.	
errore	RDS-EVENT-0220	L'istanza database è nel periodo di raffreddamento per una precedente operazione di dimensionamento dell'archiviazione. Ottimizzazione dell'istanza database in corso. Questa operazione può richiedere almeno 6 ore. Nessuna operazione richiesta. La scalabilità automatica si riprova dopo il periodo di raffreddamento.	

Categoria	ID evento RDS	Messaggio	Note
errore	RDS-EVENT-0223	Il dimensionamento automatico dell'archiviazione non è in grado di dimensionare l'archiviazione per il seguente motivo: *motivo*	
errore	RDS-EVENT-0224	Il dimensionamento automatico dell'archiviazione ha attivato un'attività di dimensionamento dell'archiviazione in sospeso che raggiungerà o supererà la soglia massima di archiviazione. Aumenta la soglia massima di archiviazione.	
errore	RDS-EVENT-0237	Il tipo di archiviazione dell'istanza database non è attualmente disponibile nella zona di disponibilità. Il ridimensionamento automatico verrà riprovato in un secondo momento.	
errore	RDS-EVENT-0254	La quota di archiviazione sottostante per questo account cliente ha superato il limite. Aumentare la quota di archiviazione consentita per consentire il dimensionamento dell'istanza.	

Categoria	ID evento RDS	Messaggio	Note
errore	RDS-EVENT-0278	La creazione dell'istanza database non è riuscita. *messaggio*	*messaggio* include dettagli sull'operazione non riuscita.
errore	RDS-EVENT-0279	La promozione della replica di lettura RDS Custom non è riuscita. *messaggio*	*messaggio* include dettagli sull'operazione non riuscita.
errore	RDS-EVENT-0280	RDS Custom non è riuscito ad aggiornare l'istanza database perché il controllo preliminare non è riuscito. *messaggio*	*messaggio* include dettagli sull'operazione non riuscita.
errore	RDS-EVENT-0281	RDS Custom non è riuscito a modificare l'istanza database perché il controllo preliminare non è riuscito. *messaggio*	*messaggio* include dettagli sull'operazione non riuscita.
errore	RDS-EVENT-0282	RDS Custom non è riuscito a modificare l'istanza database perché le autorizzazioni degli indirizzi IP elastici non sono corrette. Verificare che l'indirizzo IP elastico sia contrassegnato con `AWSRDSCustom` .	

Categoria	ID evento RDS	Messaggio	Note
errore	RDS-EVENT-0283	RDS Custom non è riuscito a modificare l'istanza database perché è stato raggiunto il limite di indirizzi IP elastici nell'account. Rilasciare gli indirizzi IP elastici non utilizzati o richiedere un aumento della quota per il limite di indirizzi IP elastici.	
errore	RDS-EVENT-0284	RDS Custom non è riuscito a convertire l'istanza in istanza a disponibilità elevata perché il controllo preliminare non è riuscito. *`messaggio`*	*`messaggio`* include dettagli sull'operazione non riuscita.
errore	RDS-EVENT-0285	RDS Custom non è riuscito a creare uno snapshot finale per l'istanza database per il sgeuente motivo: *`messaggio`* .	*`messaggio`* include dettagli sull'operazione non riuscita.
errore	RDS-EVENT-0306	L'aggiornamento della configurazione dell'arch iviazione non è riuscito. Prova a eseguire nuovamente l'aggiorn amento.	

Categoria	ID evento RDS	Messaggio	Note
errore	RDS-EVENT-0315	Impossibile spostare il database di rete *nome* non compatibile allo stato disponibile: *messaggio*	La configurazione di rete del database non è valida. Impossibile spostare il database da una rete non compatibile a una rete disponibile.
errore	RDS-EVENT-0328	Impossibile aggiungere un host a un dominio. Lo stato di appartenenza al dominio, ad esempio *instancename* , è stato impostato su Failed.	
errore	RDS-EVENT-0329	Impossibile aggiungere un host al tuo dominio. Durante il processo di aggiunta al dominio, Microsoft Windows ha restituito il *messaggio* del codice di errore. Verifica le configurazioni di rete e di autorizzazione ed invia una `modify-db-instance` richiesta per ripetere l'accesso al dominio.	Quando si utilizza un Active Directory autogestito, vedere. Risoluzione dei problemi di Active Directory gestito dal cliente
errore	RDS-EVENT-0353	L'istanza DB non può essere creata a causa di limiti di risorse insufficienti. *messaggio* .	*messaggio* include dettagli sull'operazione non riuscita.

Categoria	ID evento RDS	Messaggio	Note
errore	RDS-EVENT-0356	RDS non è riuscito a configurare l'endpoint Kerberos nel tuo dominio. Ciò potrebbe impedire l'autenticazione Kerberos per l'istanza DB. Verifica la configurazione di rete tra l'istanza DB e i controller di dominio.	
storage insufficiente	RDS-EVENT-0007	Lo spazio di archiviazione allocato è stato esaurito. Allocare spazio di archiviazione aggiuntivo per risolvere il problema.	Lo storage allocato per l'istanza database è esaurito. Per risolvere questo problema, assegna ulteriore storage per l'istanza database. Per ulteriori informazioni, consulta le domande frequenti su RDS. Puoi monitorare lo spazio di storage per un'istanza database usando il parametro Free Storage Space (Spazio di storage libero).

Categoria	ID evento RDS	Messaggio	Note
storage insufficiente	RDS-EVENT-0089	Lo spazio di archiviazione libero per l'istanza database: *nome* è basso rispetto al valore *percentuale* dell'archiviazione assegnata [archiviazione assegnata: *dimensione*, spazio di archiviazione libero: *dimensione*]. È possibile aumentare lo spazio di archiviazione assegnato per risolvere questo problema.	L'istanza database ha utilizzato oltre il 90% dello storage allocato. Puoi monitorare lo spazio di storage per un'istanza database usando il parametro Free Storage Space (Spazio di storage libero).
storage insufficiente	RDS-EVENT-0227	Lo spazio di archiviazione del cluster Aurora è pericolosamente basso con solo *valore* di terabyte rimanenti. Prendere misure idonee per ridurre il carico di archiviazione sul cluster.	Il sottosistema di archiviazione di Aurora sta esaurendo lo spazio.
manutenzione	RDS-EVENT-0026	Applicazione di patch offline all'istanza database in corso.	È in corso la manutenzione offline dell'istanza database. L'istanza database non è attualmente disponibile.
manutenzione	RDS-EVENT-0027	Applicazione di patch offline all'istanza database terminata.	La manutenzione offline dell'istanza database è stata completata. L'istanza database è ora disponibile.

Categoria	ID evento RDS	Messaggio	Note
manutenzione	RDS-EVENT-0047	Applicazione delle patch all'istanza database completata.	
manutenzione	RDS-EVENT-0155	È disponibile un aggiornamento della versione secondaria del motore di database per l'istanza database.	
manutenzione	RDS-EVENT-0264	Il controllo preliminare per l'aggiornamento della versione del motore di database è iniziato.	
manutenzione	RDS-EVENT-0265	Il controllo preliminare per l'aggiornamento della versione del motore di database è terminato.	
manutenzione	RDS-EVENT-0266	Il tempo di inattività per l'istanza database è iniziato.	
manutenzione	RDS-EVENT-0267	L'aggiornamento della versione del motore è iniziato.	
manutenzione	RDS-EVENT-0268	L'aggiornamento della versione del motore è terminato.	
manutenzione	RDS-EVENT-0269	Le attività successive all'aggiornamento sono in corso.	

Categoria	ID evento RDS	Messaggio	Note
manutenzione	RDS-EVENT-0270	L'aggiornamento della versione del motore di database non è riuscito. Il rollback dell'aggiornamento della versione del motore è riuscito.	
manutenzione, guasto	RDS-EVENT-0195	*message*	L'aggiornamento del file dei fusi orari non riuscito. Per ulteriori informazioni, consulta Aggiornamento automatico dei file di fuso orario Oracle.
manutenzione, notifica	RDS-EVENT-0191	Una nuova versione del file dei fusi orari è disponibile per l'aggiornamento.	Se il motore RDS per Oracle DB viene aggiornato, Amazon RDS genera questo evento se non è stato scelto un aggiornamento del file dei fusi orari e il database non utilizza il file dei fusi orari più recente relativo all'ora legale disponibile nell'istanza. Per ulteriori informazioni, consulta Aggiornamento automatico dei file di fuso orario Oracle.

Categoria	ID evento RDS	Messaggio	Note
manutenzione, notifica	RDS-EVENT-0192	L'aggiornamento del file dei fusi orari è stato avviato.	L'aggiornamento del file del fuso orario Oracle è iniziato. Per ulteriori informazioni, consulta Aggiornamento automatico dei file di fuso orario Oracle.
manutenzione, notifica	RDS-EVENT-0193	Nessun aggiornamento disponibile per la versione corrente del file dei fusi orari.	L'istanza database Oracle utilizza la versione più recente del file dei fusi orari e una delle seguenti affermazioni è vera: • Di recente è stata aggiunta l'opzione `TIMEZONE_FILE_AUTOUPGRADE` . • Il motore database Oracle è in fase di aggiornamento. Per ulteriori informazioni, consulta Aggiornamento automatico dei file di fuso orario Oracle.
manutenzione, notifica	RDS-EVENT-0194	L'aggiornamento del file dei fusi orari è terminato.	L'aggiornamento del file dei fusi orari Oracle è stato completato. Per ulteriori informazioni, consulta Aggiornamento automatico dei file di fuso orario Oracle.

Categoria	ID evento RDS	Messaggio	Note
notifica	RDS-EVENT-0044	*message*	Si tratta di una notifica emessa dall'operatore. Per ulteriori informazioni, consulta il messaggio di evento.
notifica	RDS-EVENT-0048	Ritardo dell'aggiornamento del motore di database in corso perché questa istanza contiene repliche di lettura che prima devono essere aggiornate.	L'applicazione di patch all'istanza database è stata ritardata.
notifica	RDS-EVENT-0054	*message*	Il motore di storage MySQL in uso non è InnoDB, ovvero il motore di storage MySQL consiglia to per Amazon RDS. Per informazioni sui motori di archiviazione MySQL, consulta Motori di storage supportati per RDS for MySQL.

Categoria	ID evento RDS	Messaggio	Note
notification	RDS-EVENT-0055	*message*	Il numero di tabelle disponibili per l'istanza database supera le best practice consiglia te per Amazon RDS. Riduci il numero di tabelle nell'istanza database. Per informazioni sulle best practice consiglia te, consulta Linee guida operative di base per Amazon RDS.
notifica	RDS-EVENT-0056	*message*	Il numero di database disponibili per l'istanza database supera le best practice consigliate per Amazon RDS. Riduci il numero di database nell'istanza database. Per informazioni sulle best practice consiglia te, consulta Linee guida operative di base per Amazon RDS.
notifica	RDS-EVENT-0064	Rotazione della chiave di crittografia TDE riuscita.	Per informazioni sulle best practice consiglia te, consulta Linee guida operative di base per Amazon RDS.

Categoria	ID evento RDS	Messaggio	Note
notifica	RDS-EVENT-0084	Impossibile convertire l'istanza database in Multi-AZ: *messaggio* .	Hai tentato di convertir e un'istanza database in Multi-AZ, ma contiene gruppi di file in memoria non supportati per il Multi-AZ. Per ulteriori informazi oni, consulta Implement azioni Multi-AZ per Amazon RDS for Microsoft SQL Server.
notifica	RDS-EVENT-0087	Istanza database arrestata.	
notification	RDS-EVENT-0088	Istanza database creata.	
notification	RDS-EVENT-0154	L'istanza database è in fase di avvio perché ha superato il tempo massimo concesso per l'arresto.	

Categoria	ID evento RDS	Messaggio	Note
notification	RDS-EVENT-0157	Impossibile modificar e la classe dell'istanza database. *messaggio* .	RDS non può modificar e la classe di istanza database perché la classe di istanza di destinazi one non può supportar e il numero di database esistenti nell'istanza database di origine. Viene visualizzato il messaggio di errore: "The instance has N databases, but after conversion it would only support N" (L'istanza presenta N database, ma in seguito alla conversione potrebbe supportarne solo N). Per ulteriori informazi oni, consulta Restrizioni per le istanze database di Microsoft SQL Server.
notifica	RDS-EVENT-0158	Lo stato dell'istanza database non può essere aggiornato: *messaggio*	
notification	RDS-EVENT-0167	*message*	La configurazione perimetrale di supporto RDS Custom è stata modificata.

Categoria	ID evento RDS	Messaggio	Note
notification	RDS-EVENT-0189	I crediti del saldo di burst gp2 per l'istanza database RDS sono bassi. Per risolvere questo problema, ridurre l'utilizzo di IOPS o modificare le impostazioni di archiviazione per consentire prestazioni superiori.	I crediti del saldo di burst gp2 per l'istanza database RDS sono bassi. Per risolvere questo problema, ridurre l'utilizzo di IOPS o modificare le impostazioni di archiviazione per consentire prestazioni superiori. Per ulteriori informazioni, consulta Crediti I/O e prestazioni espandibili nella Guida per l'utente di Amazon Elastic Compute Cloud.
notification	RDS-EVENT-0225	La dimensione dell'archiviazione allocata *valore* GB sta raggiungendo la soglia massima di archiviazione *valore* GB. Aumenta la soglia massima di archiviazione.	Questo evento viene richiamato quando l'archiviazione allocata raggiunge l'80% della soglia massima di occupazione. Per evitare l'evento, aumentare la soglia massima di occupazione dello spazio di archiviazione.

Categoria	ID evento RDS	Messaggio	Note
notification	RDS-EVENT-0231	La modifica dell'arch iviazione dell'istanza database ha riscontrato un errore interno. La richiesta di modifica è in sospeso e la sua esecuzione verrà riprovata in seguito.	Si è verificato un errore interno nel processo di replica di lettura. Per ulteriori informazioni, consulta il messaggio di evento. Inoltre, vedere la sezione relativa alla risoluzione dei problemi per le repliche di lettura per il motore DB. • Risoluzione dei problemi relativi a una replica di lettura MariaDB • Risoluzione dei problemi relativi a una replica di lettura SQL Server • Risoluzione dei problemi relativi a una replica di lettura MySQL • Risoluzione dei problemi relativi alle repliche Oracle

Categoria	ID evento RDS	Messaggio	Note
notification	RDS-EVENT-0253	Il database utilizza il buffer di doppia scrittura. *`messaggio`* . Per ulteriori informazioni, consulta la documentazione di Scritture ottimizzate per Amazon RDS per *`nome`*.	RDS con scritture ottimizzate non è compatibile con la configurazione dell'architettura dell'istanza. Per ulteriori informazioni, consulta Prestazioni di scrittura migliorate con Scritture ottimizzate per Amazon RDS per MySQL e Prestazioni di scrittura migliorate con Scritture ottimizzate per Amazon RDS per MariaDB. È possibile eseguire l'aggiornamento della configurazione dell'architettura per abilitare Scritture ottimizzate creando un'implementazione blu/verde.
notification	RDS-EVENT-0297	La configurazione dell'architettura per l'istanza database *`nome`* supporta una dimensione massima di 16384 GiB. Esegui un aggiornamento della configurazione dell'architettura per supportare dimensioni di archiviazione superiori a 16384 GiB.	Non è possibile aumentare la dimensione dell'architettura allocata dell'istanza database oltre 16384 GiB. Per superare questa limitazione, esegui un aggiornamento della configurazione dell'architettura. Per ulteriori informazioni, consulta Aggiornamento del file system di archiviazione per un'istanza DB.

Categoria	ID evento RDS	Messaggio	Note
notification	RDS-EVENT-0298	La configurazione dell'archiviazione per l'istanza database *nome* supporta una dimensione di tabella massima di 2048 GiB. Esegui un aggiornamento della configurazione dell'archiviazione per supportare dimensioni di tabella superiori a 2048 GiB.	Le istanze RDS MySQL e MariaDB con questa limitazione non possono avere una dimensione della tabella superiore a 2048 GiB. Per superare questa limitazione, esegui un aggiornamento della configurazione dell'archiviazione. Per ulteriori informazioni, vedere Aggiornamento del file system di archiviazione per un'istanza DB.
notification	RDS-EVENT-0327	Amazon RDS non è riuscito a trovare il SECRET *ARN segreto. messaggio.*	
replica di lettura	RDS-EVENT-0045	La replica è stata interrotta.	La replica sull'istanza database è stata arrestata a causa di uno spazio di archiviazione insufficiente. Ridimensionare lo spazio di archiviazione o ridurre la dimensione massima dei redo log per consentire la continuazione della replica. Per ospitare redo log di dimensioni (*MiB)*, è necessaria almeno una quantità di spazio di archiviazione libero in *MiB*.

Categoria	ID evento RDS	Messaggio	Note
replica di lettura	RDS-EVENT-0046	Replica della replica di lettura ripristinata.	Questo messaggio viene visualizzato quando crei per la prima volta una replica di lettura o come messaggio di monitoraggio che conferma il corretto funzionamento della replica. Se il messaggio segue una notifica `RDS-EVENT-0045` , la replica viene ripristinata in seguito a un errore o dopo l'arresto della replica.
replica di lettura	RDS-EVENT-0057	Lo streaming della replica è stata interrotta.	
replica di lettura	RDS-EVENT-0062	La replica della replica di lettura è stata arrestata manualmente.	
replica di lettura	RDS-EVENT-0063	La replica da un'istanza non RDS è stata reimpostata.	
replica di lettura	RDS-EVENT-0202	Creazione della replica di lettura non riuscita.	
replica di lettura	RDS-EVENT-0357	*`Nome del canale di replica avviato.`*	Per informazioni sui canali di replica, vedere. the section called “Configurazione della replica da più fonti”

Categoria	ID evento RDS	Messaggio	Note
replica di lettura	RDS-EVENT-0358	*Nome del canale di replica interrotto.*	Per informazioni sui canali di replica, vedere. the section called “Configur azione della replica da più fonti”
replica di lettura	RDS-EVENT-0359	*Il nome del canale di replica è stato interrotto manualmente.*	Per informazioni sui canali di replica, vedere. the section called “Configur azione della replica da più fonti”
replica di lettura	RDS-EVENT-0360	*Il nome del canale di replica è stato reimpostato.*	Per informazioni sui canali di replica, vedere. the section called “Configur azione della replica da più fonti”
recupero	RDS-EVENT-0020	È stato avviato il recupero dell'istanza database. La durata del recupero varia in funzione della quantità di dati da recuperare.	
recupero	RDS-EVENT-0021	È stato completato il recupero dell'istanza database.	
recupero	RDS-EVENT-0023	Richiesta snapshot emergente: *messaggio* .	È stato richiesto un backup manuale, ma in Amazon RDS è in corso la creazione di uno snapshot DB. Invia di nuovo la richiesta quando Amazon RDS avrà completato lo snapshot DB.

Categoria	ID evento RDS	Messaggio	Note
recupero	RDS-EVENT-0052	Ripristino dell'istanza Multi-AZ avviato.	La durata del recupero varia in funzione della quantità di dati da recuperare.
recupero	RDS-EVENT-0053	Ripristino dell'istanza Multi-AZ completato. Failover o attivazione in sospeso.	
recupero	RDS-EVENT-0066	Mentre il mirroring viene ristabilito, l'istanza verrà degradata: *`messaggio`* .	L'istanza database di SQL Server sta ristabile ndo il relativo mirror. Le prestazioni subiranno un peggioramento fino al termine dell'operazione. È stato trovato un database con un modello di recupero non FULL. Il modello di ripristino è stato impostato di nuovo su FULL ed è stato avviato il ripristino del mirroring (<dbname>: <recovery model found>[,…])"
recupero	RDS-EVENT-0166	*`message`*	L'istanza RDS Custom DB si trova all'interno del perimetro di supporto.
ripristino	RDS-EVENT-0019	Ripristino dall'istanza database *`nome`* in *`nome`* eseguito.	L'istanza DB è stata ripristinata da un point-in-time backup.

Categoria	ID evento RDS	Messaggio	Note
sicurezza	RDS-EVENT-0068	Decrittografia della password della partizione hsm in corso per aggiornar e l'istanza.	RDS sta decriptando la password della AWS CloudHSM partizione per aggiornare l'istanza DB. Per ulteriori informazi oni, consulta Oracle Database Transparent Data Encryption (TDE) con AWS CloudHSM nella Guida per l'utente di AWS CloudHSM .
applicazione di patch di sicurezza	RDS-EVENT-0230	È disponibile un aggiornam ento del sistema per l'istanza database. Per informazioni sull'appl icazione degli aggiornam enti, consulta "Manutenz ione di un'istanza database" nella Guida per l'utente di RDS.	È disponibile un nuovo aggiornamento per il sistema operativo. È disponibile un nuova versione secondaria dell'aggiornamento del sistema operativo per l'istanza database. Per informazioni sull'appl icazione degli aggiornam enti, consulta Utilizzo degli aggiornamenti del sistema operativo.

Eventi gruppo di parametri database

La tabella seguente riporta la categoria di eventi e un elenco di eventi applicabili quando il tipo di origine è un gruppo dei parametri database.

Categoria	ID evento RDS	Messaggio	Note
modifica della configurazione	RDS-EVENT-0037	Parametro *name* aggiornato a *value* con il metodo di applicazione *method*.	

Eventi gruppo di sicurezza DB

La tabella seguente riporta la categoria di eventi e un elenco di eventi applicabili quando il tipo di origine è un gruppo di sicurezza DB.

> **Note**
>
> I gruppi di sicurezza del database sono risorse per EC2-Classic. EC2-Classic è stato ritirato il 15 agosto 2022. Se non hai eseguito la migrazione da EC2-Classic a un VPC, ti consigliamo di eseguirla il prima possibile. Per ulteriori informazioni, consulta Eseguire la migrazione da EC2-Classic a un VPC nella Guida per l'utente di Amazon EC2 e il blog EC2-Classic Networking is Retiring – Here's How to Prepare (Il networking EC2-Classic viene ritirato: ecco come prepararsi).

Categoria	ID evento RDS	Messaggio	Note
modifica della configurazione	RDS-EVENT-0038	Modifica al gruppo di sicurezza applicata.	
errore	RDS-EVENT-0039	Revoca dell'autorizzazione come *utente*.	Il gruppo di sicurezza di proprietà di *utente* non esiste. L'autorizzazione per gruppo di sicurezza è stata revocata perché non è valida.

Eventi degli snapshot DB

La tabella seguente riporta la categoria di eventi e un elenco di eventi applicabili quando il tipo di origine è uno snapshot DB.

Categoria	ID evento RDS	Messaggio	Note
creazione	RDS-EVENT-0040	Creazione di uno snapshot manuale.	
creazione	RDS-EVENT-0042	Creazione di uno snapshot manuale completata.	
creazione	RDS-EVENT-0090	Creazione di uno snapshot automatizzato.	
creazione	RDS-EVENT-0091	Creazione di uno snapshot automatizzato completata.	
eliminazione	RDS-EVENT-0041	Snapshot utente eliminato.	
notification	RDS-EVENT-0059	Copia dello snapshot *nome* dalla regione *nome* avviata.	Questa è una copia di snapshot tra regioni.
notifica	RDS-EVENT-0060	Copia dello snapshot *nome* della regione *nome* terminata in *numero* minuti.	Questa è una copia di snapshot tra regioni.
notifica	RDS-EVENT-0061	Richiesta di copia dello snapshot *nome* dalla regione *nome* annullata.	Questa è una copia di snapshot tra regioni.
notifica	RDS-EVENT-0159	Attività di esportazione dello snapshot non riuscita.	
notification	RDS-EVENT-0160	Attività di esportazione dello snapshot annullata.	
notification	RDS-EVENT-0161	Attività di esportazione dello snapshot completata.	

Categoria	ID evento RDS	Messaggio	Note
notification	RDS-EVENT-0196	Copia dello snapshot *nome* nella regione *nome* avviata.	Questa è una copia di snapshot locale.
notifica	RDS-EVENT-0197	Copia dello snapshot *nome* nella regione *nome* terminata.	Questa è una copia di snapshot locale.
notifica	RDS-EVENT-0190	Richiesta di copia dello snapshot *nome* nella regione *nome* annullata.	Questa è una copia di snapshot locale.
ripristino	RDS-EVENT-0043	Ripristino dallo snapshot *nome* eseguito.	È in corso il ripristino di un'istanza database da uno snapshot DB.

Eventi snapshot cluster di database

La tabella seguente riporta la categoria di eventi e un elenco di eventi applicabili quando il tipo di origine è uno snapshot del cluster database.

Categoria	ID evento RDS	Messaggio	Note
backup	RDS-EVENT-0074	Creazione di uno snapshot del cluster manuale in corso.	
backup	RDS-EVENT-0075	Snapshot del cluster manuale creato.	
backup	RDS-EVENT-0168	Creazione snapshot cluster automatizzato.	
backup	RDS-EVENT-0169	Snapshot di cluster automatizzato creato.	

Eventi RDS Proxy

La tabella seguente riporta la categoria di eventi e un elenco di eventi applicabili quando il tipo di origine è un proxy RDS.

Categoria	ID evento RDS	Messaggio	Note
modifica della configurazione	RDS-EVENT-0204	Proxy DB *nome* modificato da RDS.	
modifica della configurazione	RDS-EVENT-0207	RDS ha modificato l'endpoint del proxy DB *nome*.	
modifica della configurazione	RDS-EVENT-0213	RDS ha rilevato l'aggiunta dell'istanza database e l'ha aggiunta automaticamente al gruppo di destinazione del proxy DB *nome*.	
modifica della configurazione	RDS-EVENT-0213	RDS ha rilevato la creazione dell'istanza database *nome* e l'ha rimossa automaticamente dal gruppo di destinazione *nome* del proxy DB *nome*.	
modifica della configurazione	RDS-EVENT-0214	RDS ha rilevato l'eliminazione dell'istanza database *nome* e l'ha rimossa automaticamente dal gruppo di destinazione *nome* del proxy DB *nome*.	
modifica della configurazione	RDS-EVENT-0215	RDS ha rilevato l'eliminazione del cluster database *nome* e l'ha rimosso	

Categoria	ID evento RDS	Messaggio	Note
		automaticamente dal gruppo di destinazione *nome* del proxy DB *nome*.	
creazione	RDS-EVENT-0203	RDS ha creato il proxy DB *nome*.	
creazione	RDS-EVENT-0206	RDS ha creato l'endpoint *nome* del proxy DB *nome*.	
eliminazione	RDS-EVENT-0205	RDS ha eliminato il proxy DB *nome*.	
eliminazione	RDS-EVENT-0208	RDS ha eliminato l'endpoint *nome* per il proxy DB *nome*.	
errore	RDS-EVENT-0243	RDS non è riuscito ad eseguire il provisioning della capacità per il proxy *nome* perché non ci sono sufficienti indirizzi IP disponibili nelle sottoreti : *nome*. Per risolvere il problema, assicurarsi che le sottoreti abbiano il numero minimo di indirizzi IP non utilizzati come consigliato nella documentazione di Server proxy per Amazon RDS.	Per determinare il numero consigliato per la classe di istanza, consulta Pianificazione della capacità degli indirizzi IP.

Categoria	ID evento RDS	Messaggio	Note
errore	RDS-EVENT-0275	*RDS ha limitato alcune connessioni al nome del proxy DB.* Il numero di richieste di connessione simultane e dal client al proxy ha superato il limite.	

Eventi di implementazione blu/verde

Nella tabella seguente sono indicati la categoria di evento e un elenco di eventi quando l'implementazione blu/verde è un tipo di origine.

Per ulteriori informazioni sulle implementazioni blu/verde, consulta Utilizzo delle implementazioni blu/verde Amazon RDS per gli aggiornamenti del database.

Categoria	ID evento Amazon RDS	Messaggio	Note
creazione	RDS-EVENT-0244	Le attività di implement azione blu/verde sono state completate. È possibile apportare ulteriori modifiche ai database dell'ambiente verde o passare all'imple mentazione.	
errore	RDS-EVENT-0245	La creazione dell'impl ementazione blu/verde non è riuscita perché il database (origine/destinazi one) (istanza/cluster) non è stato trovato.	

Categoria	ID evento Amazon RDS	Messaggio	Note
eliminazione	RDS-EVENT-0246	L'implementazione blu/verde è stata eliminata.	
notification	RDS-EVENT-0247	Lo switchover da *blu* a *verde* è iniziato.	
notification	RDS-EVENT-0248	Lo switchover è stato completato per l'impleme ntazione blu/verde.	
errore	RDS-EVENT-0249	Lo switchover è stato annullato per l'impleme ntazione blu/verde.	
notification	RDS-EVENT-0250	Lo switchover della replica di lettura primaria da *blu* a *verde* è iniziato.	
notification	RDS-EVENT-0251	Lo switchover della replica di lettura primaria da *blu* a *verde* è completato. È stato rinominato *blu* in *blu-precedente* e *verde* in *blu*.	
errore	RDS-EVENT-0252	Lo switchover della replica di lettura primaria da *blu* a *verde* è stato annullato a causa del *motivo*.	

Categoria	ID evento Amazon RDS	Messaggio	Note
notification	RDS-EVENT-0307	La sincronizzazione della sequenza per lo switchover del da *blu* a *verde* è iniziata. Quando si utilizzano le sequenze lo switchover può comportare tempi di inattività prolungati.	
notification	RDS-EVENT-0308	La sincronizzazione della sequenza per lo switchover del da *blu* a *verde* è completata.	
errore	RDS-EVENT-0310	La sincronizzazione della sequenza dello switchover del da *blu* a *verde* è stata annullata perché le sequenze non sono state sincronizzate.	

Eventi di versioni personalizzate del motore

La tabella seguente mostra la categoria di evento e un elenco di eventi quando il tipo di origine è una versione personalizzata del motore.

Categoria	ID evento Amazon RDS	Messaggio	Note
creazione	RDS-EVENT-0316	Preparazione per la creazione del *nome* della versione del motore personalizzato. Il completamento dell'intero processo di creazione può	

Categoria	ID evento Amazon RDS	Messaggio	Note
		richiedere fino a quattro ore.	
creazione	RDS-EVENT-0317	Creazione del *nome* della versione del motore personalizzato.	
creazione	RDS-EVENT-0318	Convalida del *nome* della versione del motore personalizzato.	
creazione	RDS-EVENT-0319	Il *nome* della versione del motore personalizzato è stato creato correttamente.	
creazione	RDS-EVENT-0320	RDS non può creare il *nome* della versione del motore personalizzato a causa di un problema interno. Stiamo risolvendo il problema e ti contatteremo se necessario. Per ulteriore assistenza, contatta AWS Premium Support.	
errore	RDS-EVENT-0198	Creazione non riuscita per la versione personalizzata del motore *nome*. *messaggio*	*messaggio* include i dettagli sull'operazione non riuscita, ad esempio file mancanti.
errore	RDS-EVENT-0277	Errore durante l'eliminazione del *nome* della versione del motore personalizzato. *messaggio*	*messaggio* include dettagli sull'operazione non riuscita.

Categoria	ID evento Amazon RDS	Messaggio	Note
ripristino	RDS-EVENT-0352	Il numero massimo di database supportato per il ripristino è cambiato. point-in-time	Il *`messaggio`* include dettagli sull'evento.

Monitoraggio dei file di log di Amazon RDS

Ogni motore di database RDS genera registri cui è possibile accedere per il controllo e la risoluzione dei problemi. Il tipo di registri dipende dal motore di database.

Puoi accedere ai registri del database tramite la AWS Management Console, AWS Command Line Interface (AWS CLI) o l'API di Amazon RDS. Non puoi visualizzare, controllare o scaricare registri delle transazioni.

Argomenti

Visualizzazione ed elenco dei file di log del database

Puoi visualizzare i file di log del database per il motore DB Amazon RDS utilizzando la AWS Management Console. Puoi elencare i file di log disponibili per il download o il monitoraggio tramite AWS CLI o l'API di Amazon RDS.

> **Note**
>
> Se non riesci a visualizzare l'elenco dei file di log per un'istanza database Oracle esistente, riavvia l'istanza per visualizzare l'elenco.

Console

Per visualizzare un file di log di database

1. Apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, scegliere Databases (Database).
3. Scegliere il nome dell'istanza di database che ha il file di log che si desidera visualizzare.
4. Scegliere la scheda Logs & events (Log ed eventi).
5. Scorrere fino alla sezione Logs (Log).
6. (Opzionale) Inserisci un termine di ricerca per filtrare i risultati.
7. Scegli il log che desideri visualizzare, quindi seleziona View (Visualizza).

AWS CLI

Per elencare i file di log del database disponibili per un'istanza database, utilizza il comando AWS CLI `describe-db-log-files`.

Il seguente esempio restituisce un elenco di file di log per un'istanza database denominata `my-db-instance`.

Example

```
aws rds describe-db-log-files --db-instance-identifier my-db-instance
```

API RDS

Per elencare i file di log del database disponibili per un'istanza database usa l'operazione API Amazon RDS `DescribeDBLogFiles`.

Download di un file di log di database

Puoi usare la AWS Management Console, AWS CLI o l'API per scaricare un file di log del database.

Console

Per scaricare un file di log di database

1. Apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.

2. Nel pannello di navigazione, scegliere Databases (Database).
3. Scegliere il nome dell'istanza di database che ha il file di log che si desidera visualizzare.
4. Scegliere la scheda Logs & events (Log ed eventi).
5. Scorrere fino alla sezione Logs (Log).
6. Nella sezione Logs (Log), selezionare il pulsante accanto al log che si desidera scaricare, quindi selezionare Download (Scarica).
7. Aprire il menu contestuale (clic con il tasto destro del mouse) per il collegamento fornito, quindi scegliere Save Link As (Salva collegamento come). Immettere l'ubicazione in cui si intende salvare il file di log, quindi scegliere Save (Salva).

AWS CLI

Per scaricare un file di log del database, utilizzare il comando AWS CLI `download-db-log-file-portion`. Per impostazione predefinita, questo comando scarica la porzione più recente di un file di log. Tuttavia, puoi scaricare un intero file specificando il parametro `--starting-token 0`.

L'esempio seguente mostra come scaricare tutto il contenuto di un file di log denominato log/ERROR.4 e come archiviarlo in un file locale denominato errorlog.txt.

Example

Per LinuxmacOS, oUnix:

```
aws rds download-db-log-file-portion \
    --db-instance-identifier myexampledb \
```

```
    --starting-token 0 --output text \
    --log-file-name log/ERROR.4 > errorlog.txt
```

Per Windows:

```
aws rds download-db-log-file-portion ^
    --db-instance-identifier myexampledb ^
    --starting-token 0 --output text ^
    --log-file-name log/ERROR.4 > errorlog.txt
```

API RDS

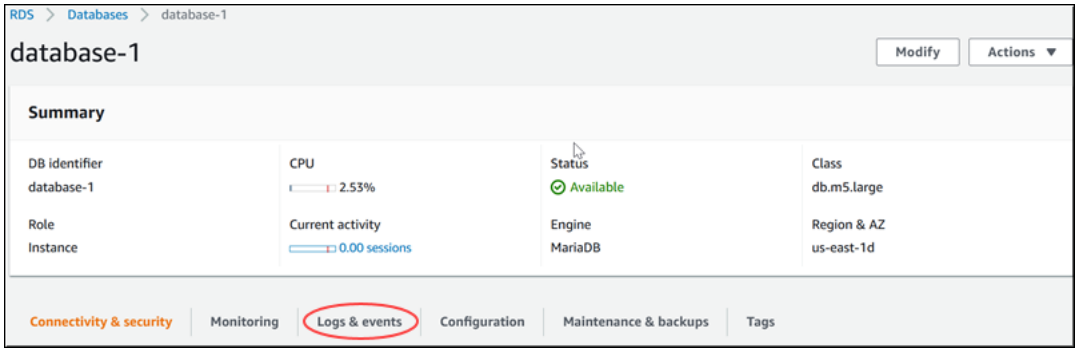
Per scaricare un file di log del database, utilizzare l'operazione API Amazon RDS `DownloadDBLogFilePortion`.

Controllo di un file di log di database

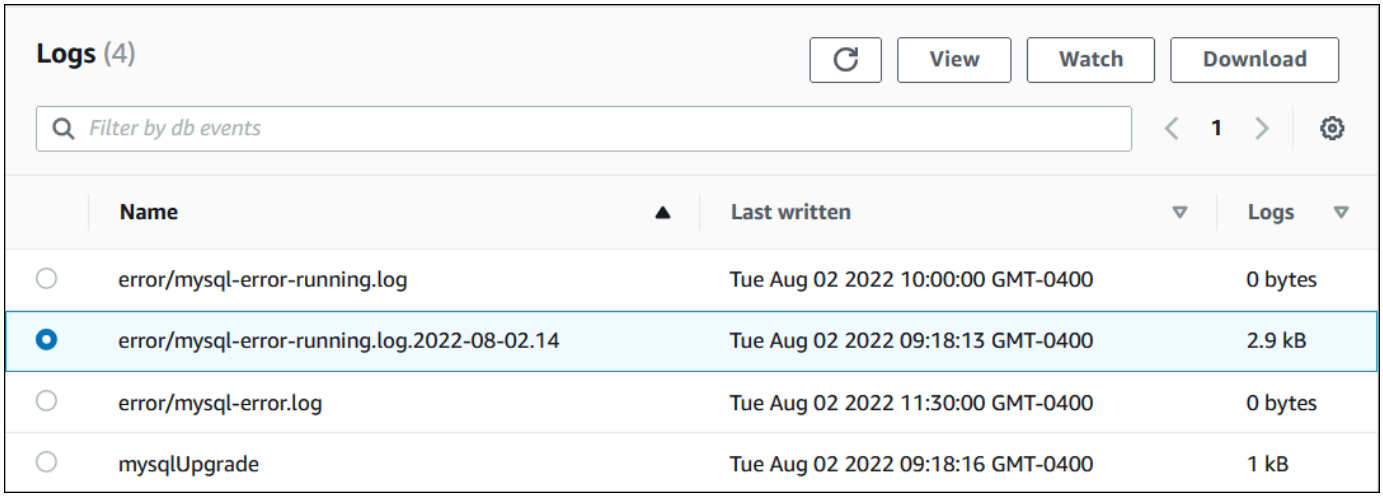
Controllare un file di registro del database equivale a eseguire l'accodamento del file su un sistema UNIX o Linux. Puoi controllare un file di registro usando la AWS Management Console. RDS aggiorna la coda del registro ogni 5 secondi.

Per controllare un file di log di database

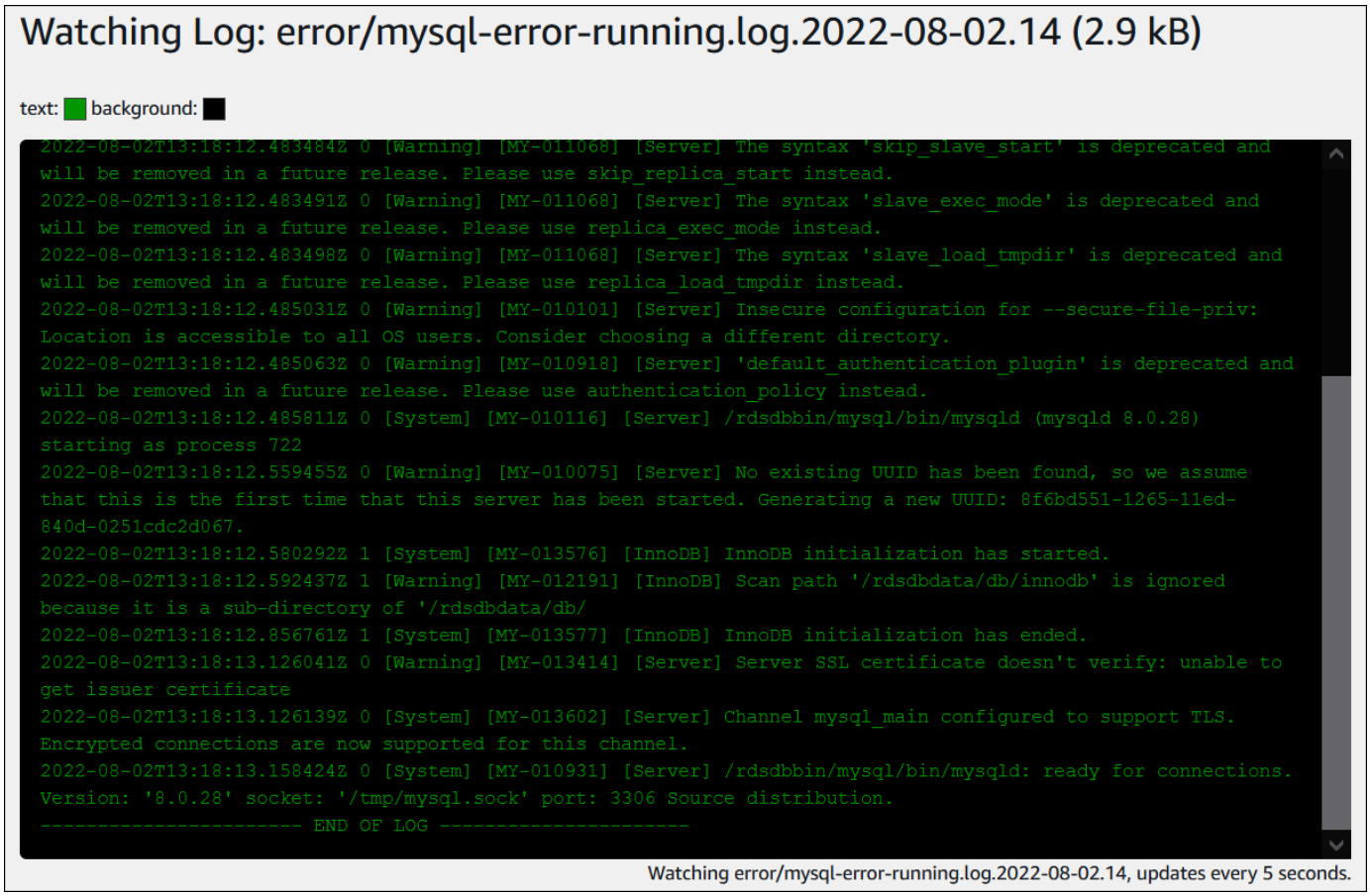
1. Apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, scegliere Databases (Database).
3. Scegliere il nome dell'istanza di database che ha il file di log che si desidera visualizzare.
4. Scegliere la scheda Logs & events (Log ed eventi).



5. Nella sezione Logs (Log), scegliere un file di log, quindi selezionare Watch (Controlla).



RDS mostra la coda del registro, come nel seguente esempio MySQL.



Pubblicazione di log di database su Amazon CloudWatch Logs

In un database on-premise, i registri del database risiedono nel file system. Amazon RDS non fornisce accesso host ai registri del database sul file system dell'istanza database. Per questo motivo,

Amazon RDS consente di esportare i registri del database nei file di log Amazon CloudWatch. Con File di log CloudWatch, puoi eseguire analisi in tempo reale dei dati dei registri. Puoi anche archiviare i dati in un archivio estremamente durevole e gestirli con l'agente File di log CloudWatch.

Argomenti

- Panoramica dell'integrazione RDS con i file di log CloudWatch
- Decidere quali registri pubblicare nei file di log CloudWatch
- Specifica dei registri da pubblicare nei file di log CloudWatch
- Ricerca e filtraggio dei registri nei file di log CloudWatch

Panoramica dell'integrazione RDS con i file di log CloudWatch

Nei file di log CloudWatch, un flusso di log è una sequenza di eventi di log che condividono la stessa origine. Ciascuna origine di registri in CloudWatch Logs costituisce un flusso di log distinto. Un gruppo di log è un gruppo di flussi di log che condividono le stesse impostazioni di conservazione, monitoraggio e controllo degli accessi.

Amazon RDS esegue lo streaming continuo dei record di log dell'istanza database in un gruppo di log. Ad esempio, disponi di un gruppo di registri `/aws/rds/instance/`*`instance_name`*`/`*`log_type`* per ogni tipo di registro che pubblichi. Questo gruppo di log si trova nella stessa regione AWS dell'istanza database che genera il log.

AWS conserva i dati di registro pubblicati nei file di log CloudWatch per un periodo di tempo indefinito a meno che non venga specificato un periodo di conservazione. Per ulteriori informazioni, consulta la pagina relativa alla modifica del periodo di conservazione dei dati dei log in CloudWatch Logs.

Decidere quali registri pubblicare nei file di log CloudWatch

Ogni motore di database RDS supporta il proprio set di registri. Per informazioni sulle opzioni per il motore di database, consulta i seguenti argomenti:

- the section called “Pubblicazione dei log di MariadB su Amazon Logs CloudWatch ”
- the section called “Pubblicazione dei log MySQL su Amazon Logs CloudWatch ”
- the section called “Pubblicazione dei log Oracle su Amazon CloudWatch Logs”
- the section called “Pubblicazione dei log PostgreSQL su Amazon Logs CloudWatch ”
- the section called “Pubblicazione dei log di SQL Server su Amazon CloudWatch Logs”

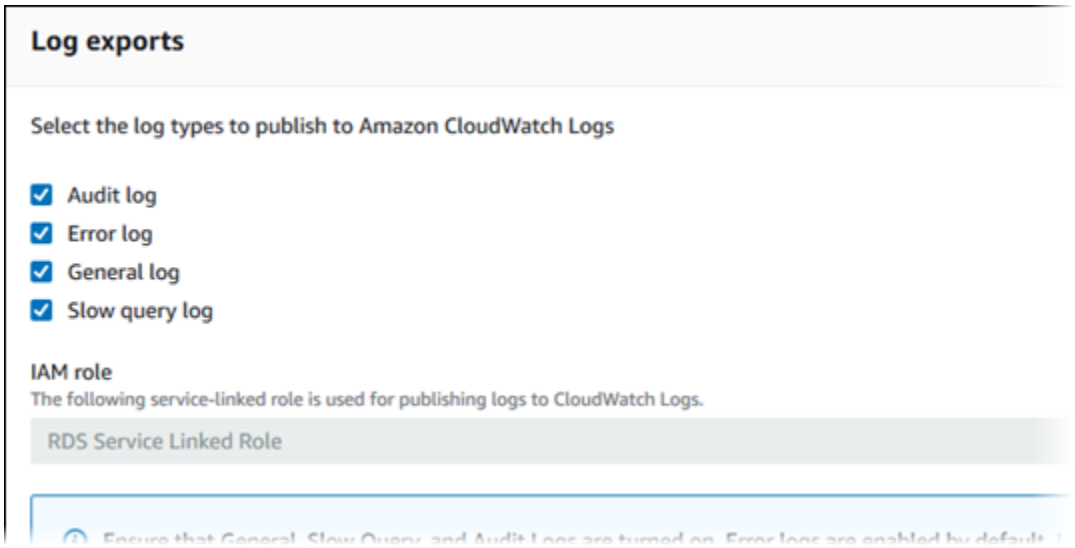
Specifica dei registri da pubblicare nei file di log CloudWatch

Puoi specificare quali registri pubblicare nella console. Assicurati di disporre di un ruolo collegato al servizio in AWS Identity and Access Management (IAM). Per ulteriori informazioni sui ruoli collegati al servizio, consulta Utilizzo di ruoli collegati ai servizi per Amazon RDS.

Per specificare i registri da pubblicare

1. Apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, scegliere Databases (Database).
3. Eseguire una delle operazioni seguenti:
   - Scegliere Crea database.
   - Scegli un database dall'elenco, quindi scegli Modify (Modifica).
4. In Logs exports (Esportazioni di log), scegli quali registri pubblicare.

   Nell'esempio seguente viene specificato il registro di controllo, i registri di errore, il registro generale e il registro query lente.



Ricerca e filtraggio dei registri nei file di log CloudWatch

Puoi cercare voci di registro che soddisfino un criterio specificato utilizzando la console File di log CloudWatch. Puoi accedere ai registri tramite la console RDS, che porta alla console File di log CloudWatch, o direttamente dalla console File di log CloudWatch.

Per cercare registri RDS utilizzando la console RDS

1. Apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, scegliere Databases (Database).
3. Scegli un'istanza database.
4. Scegliere Configuration (Configurazione).
5. In Published logs (Log pubblicati), scegli il registro del database che desideri visualizzare.

Per cercare i registri RDS utilizzando la console File di log CloudWatch

1. Aprire la console CloudWatch all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel pannello di navigazione, seleziona Log groups (Gruppi di log).
3. Nella casella del filtro, immetti **`/aws/rds`**.
4. In Log Groups (Gruppi di log), seleziona il nome del gruppo di log contenente il flusso di log da cercare.
5. In Log Streams (Flussi di log), seleziona il nome del flusso di log da cercare.
6. In Eventi di log, immettere la sintassi del filtro da utilizzare.

Per ulteriori informazioni, consulta Ricerca e filtraggio dei dati di registro nella Guida per l'utente di File di log Amazon CloudWatch. Per un blog tutorial su come monitorare i registri RDS, consulta la sezione relativa alla creazione di un monitoraggio proattivo del database per Amazon RDS con File di log Amazon CloudWatch, AWS Lambda e Amazon SNS.

Lettura dei contenuti del file di log con REST

Amazon RDS fornisce un endpoint REST che consente l'accesso ai file di log dell'istanza database. Questo è utile se hai necessità di scrivere un'applicazione per eseguire lo streaming dei contenuti del file di log Amazon RDS.

La sintassi è:

```
GET /v13/downloadCompleteLogFile/DBInstanceIdentifier/LogFileName HTTP/1.1
Content-type: application/json
host: rds.region.amazonaws.com
```

I parametri seguenti sono obbligatori:

- *DBInstanceIdentifier* — Nome dell'istanza database che contiene il file di log che vuoi scaricare.
- *LogFileName* — Nome del file di log da scaricare.

La risposta contiene i contenuti del file di log richiesto, come stream.

L'esempio seguente scarica il file di log denominato log/ERROR.6 per l'istanza database denominata sample-sql nella regione us-west-2.

```
GET /v13/downloadCompleteLogFile/sample-sql/log/ERROR.6 HTTP/1.1
host: rds.us-west-2.amazonaws.com
X-Amz-Security-Token: AQoDYXdzEIH//////////
wEa0AIXLhngC5zp9CyB1R6abwKrXHVR5efnAVN3XvR7IwqKYalFSn6UyJuEFTft9nObglx4QJ+GXV9cpACkETq=
X-Amz-Date: 20140903T233749Z
X-Amz-Algorithm: AWS4-HMAC-SHA256
X-Amz-Credential: AKIADQKE4SARGYLE/20140903/us-west-2/rds/aws4_request
X-Amz-SignedHeaders: host
X-Amz-Content-SHA256: e3b0c44298fc1c229afbf4c8996fb92427ae41e4649b934de495991b7852b855
X-Amz-Expires: 86400
X-Amz-Signature: 353a4f14b3f250142d9afc34f9f9948154d46ce7d4ec091d0cdabbcf8b40c558
```

Se specifichi un'istanza database non esistente, otterrai il seguente errore:

- DBInstanceNotFound—*DBInstanceIdentifier* non fa riferimento a un'istanza database esistente. (Codice di stato HTTP: 404)

File di log del database MariaDB

Puoi monitorare il log degli errori, il log delle query lente e il log generale di MariaDB. Il log di errori di MariaDB viene generato per impostazione predefinita. È possibile generare la query lenta e i log generali impostando i parametri nel gruppo parametri del database. Amazon RDS ruota tutti i file di log MariaDB; gli intervalli per ciascun tipo sono indicati di seguito.

Puoi monitorare i log di MariaDB direttamente tramite la console Amazon RDS, l'API Amazon RDS, la CLI di Amazon RDS o gli SDK. AWS Puoi anche eseguire l'accesso ai log MariaDB indirizzando i log a una tabella del database nel database principale e facendo una ricerca in tale tabella. Puoi utilizzare la utility mysqlbinlog per scaricare un log binario.

Per ulteriori informazioni sulla visualizzazione, il download e la visione di log di database basati su file, consulta Monitoraggio dei file di log di Amazon RDS.

Argomenti

Accesso al log degli errori MariaDB

Il log degli errori MariaDB è scritto sul file `<host-name>.err`. Puoi visualizzare questo file utilizzando la console Amazon RDS. Puoi anche recuperare il log utilizzando l'API Amazon RDS, la CLI di Amazon RDS o gli SDK. AWS Il file `<host-name>.err` viene svuotato ogni 5 minuti e i suoi contenuti vengono aggiunti a `mysql-error-running.log`. Il file `mysql-error-running.log` viene quindi ruotato ogni ora e i file che vengono generati ogni ora durante le ultime 24 ore vengono conservati. Ogni file di log ha l'ora di creazione (in UTC) accodata al nome. I file di log hanno anche un timestamp che ti aiuta a determinare quando le voci del log sono state scritte.

MariaDB scrive il log di errori solo durante l'avvio, l'arresto e quando si verificano errori. Un'istanza database può andare avanti ore senza che ci siano nuove voci scritte nel file di log degli errori. Se non vedi voci recenti, significa che il server non ha riscontrato errori per generare una voce di log.

Accesso al log delle query lente e al log generale del MariaDB

Il registro delle query lente MariaDB e quello generale possono essere scritti in un file o una tabella di database impostando i parametri nel gruppo di parametri del database. Per informazioni sulla creazione e la modifica di un gruppo di parametri database, consulta Utilizzo di gruppi di parametri. È necessario impostare questi parametri prima di poter visualizzare il log delle query lente o il registro generale nella console Amazon RDS o utilizzando l'API Amazon RDS o AWS gli AWS CLI SDK.

Puoi controllare la registrazione di MariaDB utilizzando i parametri in questo elenco:

- `slow_query_log`oppure`log_slow_query`: per creare il log delle query lente, imposta su 1. Il valore predefinito è 0.
- `general_log`: per creare il log generale, imposta su 1. Il valore predefinito è 0.
- `long_query_time`oppure`log_slow_query_time`: per evitare che le query a esecuzione rapida vengano registrate nel registro delle query lente, specificate un valore per il tempo di esecuzione della query più breve da registrare, in secondi. Il valore predefinito è 10 secondi, il minimo è 0 secondi. Se log_output = FILE, puoi specificare un valore in virgola mobile con risoluzione al microsecondo. Se log_output = TABLE, devi specificare un valore intero con risoluzione al secondo. Vengono registrate solo le query il cui tempo di esecuzione supera il valore o. `long_query_time` `log_slow_query_time` Ad esempio, l'impostazione `long_query_time` o `log_slow_query_time` su 0.1 impedisce la registrazione di qualsiasi query eseguita per meno di 100 millisecondi.
- `log_queries_not_using_indexes`: per registrare tutte le query che non usano un indice sul log delle query lente, imposta il parametro su 1. Il valore predefinito è 0. Le query che non usano un indice vengono registrate anche se il loro tempo di registrazione è inferiore al valore del parametro `long_query_time`.
- `log_output` *`option`*: puoi specificare una delle seguenti opzioni per il parametro `log_output`:
  - TABLE (predefinito) – Scrive le query generali nella tabella `mysql.general_log` e le query lente nella tabella `mysql.slow_log`.
  - FILE – Scrive sia i log generali sia i log delle query lente nel file system. I file di log vengono ruotati ogni ora.
  - NONE – Disabilita il logging.

Quando la registrazione è abilitata, Amazon RDS ruota i log delle tabelle o elimina i file di log a intervalli regolari. Questa è una precauzione per ridurre la possibilità che un file di log di grandi dimensioni blocchi l'uso del database o influisca sulle prestazioni. I logging di tipo `FILE` e `TABLE` gestiscono la rotazione e l'eliminazione in questo modo:

- Quando la registrazione `FILE` è abilitata, i file di log vengono esaminati ogni ora e quelli più vecchi di 24 ore vengono eliminati. In alcuni casi, la dimensione del file di log combinato restante dopo l'eliminazione supera la soglia del 2 per cento di spazio assegnato a un'istanza database. In questi casi, i file di log più grandi vengono eliminati fino a che le dimensioni del file di log non rimangono inferiori alla soglia.
- Quando la registrazione `TABLE` è abilitata, in alcuni casi, le tabelle di log vengono ruotate ogni 24 ore. Questa rotazione avviene se lo spazio usato dai registri delle tabelle è più del 20% dello spazio di archiviazione assegnato oppure se la dimensione di tutti i log combinati è maggiore di 10 GB. Se la quantità di spazio utilizzato per un'istanza database è maggiore del 90% dello spazio di archiviazione assegnato per l'istanza database, le soglie di rotazione dei registri vengono ridotte. Le tabelle dei registri vengono ruotate se lo spazio utilizzato dai registri delle tabelle supera il 10% dello spazio di archiviazione assegnato oppure se la dimensione di tutti i log combinati è maggiore di 5 GB.

  Quando le tabelle di log sono convertite, la tabella di log corrente è copiata in una tabella di logo di backup e le voci nella tabella di log corrente sono eliminate. Se esiste già una tabella di log di backup, questa viene eliminata prima che la tabella di log corrente sia copiata nel backup. Puoi eseguire una query sulla tabella di log di backup, se necessario. La tabella di log di backup per la tabella `mysql.general_log` è denominata `mysql.general_log_backup`. La tabella di log di backup per la tabella `mysql.slow_log` è denominata `mysql.slow_log_backup`.

  Puoi ruotare la tabella `mysql.general_log` chiamando la procedura `mysql.rds_rotate_general_log`. Puoi ruotare la tabella `mysql.slow_log` chiamando la procedura `mysql.rds_rotate_slow_log`.

  I log della tabella vengono ruotati durante l'aggiornamento della versione del database.

Amazon RDS registra la rotazione di log `TABLE` e `FILE` in un evento Amazon RDS e invia una notifica.

Per utilizzare i log della console Amazon RDS, dell'API Amazon RDS, della CLI di Amazon RDS o degli AWS SDK, imposta il parametro su FILE. `log_output` Come il log degli errori MariaDB, questi

file di log vengono ruotati ogni ora. I file di log che sono stati generati durante le precedenti 24 ore vengono conservati.

Per ulteriori informazioni sui log delle query lente e i log generali, consulta i seguenti argomenti nella documentazione del MariaDB:

- Log delle query lente
- Log delle query generali

Pubblicazione dei log di MariadB su Amazon Logs CloudWatch

Puoi configurare la tua istanza MariaDB DB per pubblicare i dati di log in un gruppo di log in Amazon Logs. CloudWatch Con CloudWatch Logs, puoi eseguire analisi in tempo reale dei dati di log e utilizzarli CloudWatch per creare allarmi e visualizzare metriche. È possibile utilizzare CloudWatch Logs per archiviare i record di registro in un archivio altamente durevole.

Amazon RDS pubblica ogni log di database MariaDB come flusso di database separato nel gruppo di log. Supponi, ad esempio, di configurare la funzione di esportazione per includere il registro delle query lente. I dati relativi alle query lente vengono archiviati in un flusso di log delle query lente nel gruppo di log `/aws/rds/instance/my_instance/slowquery`.

Il log degli errori è abilitato per impostazione predefinita. La tabella seguente fornisce un riepilogo dei requisiti per gli altri log MariaDB.

Log	Requisito
Log di controllo	L'istanza database deve usare un gruppo di opzioni personalizzato con l'opzione `MARIADB_AUDIT_PLUGIN` .
Log generale	L'istanza database deve usare un gruppo di parametri personalizzato con l'imposta zione `general_log = 1` per abilitare il log generale.
Log delle query lente	L'istanza DB deve utilizzare un gruppo di parametri personalizzato con l'impostazione dei parametri `slow_query_log = 1` o

Log	Requisito
	`log_slow_query = 1` abilitare lo slow query log.
Output log	L'istanza DB deve utilizzare un gruppo di parametri personalizzato con l'impostazione dei parametri `log_output = FILE` per scrivere i log nel file system e pubblicarli in CloudWatch Logs.

Console

Per pubblicare i log di MariadB su Logs CloudWatch dalla console

1. Apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, scegliere Databases (Database) e selezionare l'istanza database da modificare.
3. Scegliere Modify (Modifica).
4. Nella sezione Esportazioni dei log, scegli i log che desideri iniziare a pubblicare su Logs. CloudWatch
5. Scegliere Continue (Continua) e quindi Modify DB Instance (Modifica istanza database) nella pagina di riepilogo.

AWS CLI

Puoi pubblicare un registro di MariadB con. AWS CLI Puoi chiamare il comando `modify-db-instance` con i parametri seguenti:

- `--db-instance-identifier`
- `--cloudwatch-logs-export-configuration`

> **Note**
>
> Viene sempre applicata all'istanza database una modifica all'opzione `--cloudwatch-logs-export-configuration` immediatamente. Pertanto, le opzioni `--apply-immediately` e `--no-apply-immediately` non hanno alcun effetto.

Puoi anche pubblicare i log di MariadB chiamando i seguenti comandi: AWS CLI

- `create-db-instance`
- `restore-db-instance-from-db-snapshot`
- `restore-db-instance-from-s3`
- `restore-db-instance-to-point-in-time`

Esegui uno di questi AWS CLI comandi con le seguenti opzioni:

- `--db-instance-identifier`
- `--enable-cloudwatch-logs-exports`
- `--db-instance-class`
- `--engine`

Potrebbero essere necessarie altre opzioni a seconda del AWS CLI comando eseguito.

Example

L'esempio seguente modifica un'istanza di MariaDB DB esistente per pubblicare i file di registro in Logs. CloudWatch Il valore `--cloudwatch-logs-export-configuration` è un oggetto JSON. La chiave per questo oggetto è `EnableLogTypes` e il suo valore è una matrice di stringhe con qualsiasi combinazione di `audit`, `error`, `general` e `slowquery`.

PerLinux, o: macOS Unix

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --cloudwatch-logs-export-configuration '{"EnableLogTypes":
["audit","error","general","slowquery"]}'
```

Per Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --cloudwatch-logs-export-configuration '{"EnableLogTypes":
["audit","error","general","slowquery"]}'
```

Example

Il comando seguente crea un'istanza di MariaDB DB e pubblica i file di registro in Logs. CloudWatch Il valore `--enable-cloudwatch-logs-exports` è una matrice di stringhe JSON. Le stringhe possono essere una qualsiasi combinazione di `audit`, `error`, `general` e `slowquery`.

PerLinux, o: macOS Unix

```
aws rds create-db-instance \
    --db-instance-identifier mydbinstance \
    --enable-cloudwatch-logs-exports '["audit","error","general","slowquery"]' \
    --db-instance-class db.m4.large \
    --engine mariadb
```

Per Windows:

```
aws rds create-db-instance ^
    --db-instance-identifier mydbinstance ^
    --enable-cloudwatch-logs-exports '["audit","error","general","slowquery"]' ^
    --db-instance-class db.m4.large ^
    --engine mariadb
```

API RDS

Puoi pubblicare i log MariaDB con l'API di RDS. Chiama l'operazione `ModifyDBInstance` con i parametri seguenti:

- `DBInstanceIdentifier`
- `CloudwatchLogsExportConfiguration`

> **Note**
>
> Viene sempre applicata all'istanza database una modifica al parametro `CloudwatchLogsExportConfiguration` immediatamente. Pertanto, il parametro `ApplyImmediately` non ha alcun effetto.

Puoi anche pubblicare i log MariaDB eseguendo una delle seguenti azioni API RDS:

- `CreateDBInstance`
- `RestoreDBInstanceFromDBSnapshot`
- `RestoreDBInstanceFromS3`
- `RestoreDBInstanceToPointInTime`

Esegui una di queste azioni API RDS con i seguenti parametri:

- `DBInstanceIdentifier`
- `EnableCloudwatchLogsExports`
- `Engine`
- `DBInstanceClass`

Potrebbero essere necessari altri parametri a seconda del AWS CLI comando eseguito.

Dimensione del file di registro

Le dimensioni dei file di log delle query lente, degli errori e generale di MariaDB sono limitate a un massimo del 2 per cento dello spazio di storage assegnato per un'istanza database. Per rispettare questa soglia i log vengono ruotati automaticamente ogni ora e i file di log più vecchi di 24 ore vengono rimossi. Se le dimensioni del file di log combinato superano tale soglia dopo la rimozione dei file di log vecchi, i file di log più grandi vengono eliminati fino a che le dimensioni del file di log non rimangono inferiori alla soglia.

Gestione dei log MariaDB basati su tabella

Puoi indirizzare i log delle query lente e generali alle tabelle nell'istanza database. Per farlo devi creare un gruppo di parametri database e impostare il parametro server `log_output` su `TABLE`.

Le query generali vengono quindi registrate sulla tabella `mysql.general_log`, mentre le query lente vengono registrate sulla tabella `mysql.slow_log`. Puoi eseguire query sulle tabelle per avere accesso alle informazioni di log. L'abilitazione di questa registrazione aumenta il numero di dati scritti sul database, il che potrebbe compromettere le performance.

Sia il log generale che quello delle query lente sono disattivati per impostazione predefinita. Per abilitare la registrazione alle tabelle, è inoltre necessario impostare i seguenti parametri del server su: `1`

- `general_log`
- `slow_query_log` o `log_slow_query`

Le tabelle di log continuano a crescere fino a che le rispettive attività di registrazione non vengono disattivate eseguendo la reimpostazione del parametro appropriato su `0`. Spesso nel corso del tempo si accumulano grandi quantità di dati che possono usare una percentuale considerevole dello spazio di archiviazione assegnato. Amazon RDS non consente di troncare le tabelle di log, ma è possibile spostarne il contenuto. La rotazione delle tabelle ne salva il contenuto in una tabella di backup e crea una nuova tabella di log vuota. Puoi ruotare manualmente le tabelle di log con le seguenti procedure a riga di comando, nelle quali il prompt dei comandi è indicato da `PROMPT>`:

```
PROMPT> CALL mysql.rds_rotate_slow_log;
PROMPT> CALL mysql.rds_rotate_general_log;
```

Per rimuovere completamente i dati vecchi e recuperare lo spazio del disco, chiama la procedura adeguata due volte in successione.

Formato di registrazione binario

MariaDB in Amazon RDS supporta i formati di logging binario basati su riga, basati su istruzioni e quelli misti. Il formato di logging binario predefinito è quello misto. Per informazioni sui diversi formati di log binari di MariaDB, consulta la pagina relativa ai formati di log binari nella documentazione di MariaDB.

Se intendi utilizzare la replica, il formato dei log binari è importante in quanto determina il record delle modifiche dei dati che viene registrato nell'origine e inviato alle destinazioni della replica. Per ulteriori informazioni sui vantaggi e sugli svantaggi dei vari formati di logging binario per la replica, consulta la pagina relativa a vantaggi e svantaggi della replica basata su istruzioni e basata su riga nella documentazione di MySQL.

> **Important**
>
> L'impostazione del formato di registrazione binario su "basato su riga" può generare file di log binari molto grandi. I file di log binari di grandi dimensioni riducono lo spazio di archiviazione disponibile per un'istanza database e possono anche determinare un aumento della quantità di tempo necessaria per eseguire un'operazione di ripristino di un'istanza database.
> La replica basata sulle istruzioni può causare incoerenze tra l'istanza database di origine e una replica di lettura. Per ulteriori informazioni, consulta la pagina relativa a istruzioni non sicure per la replica basata su istruzioni nella documentazione di MariaDB.

Per impostare il formato di registrazione binaria MariaDB

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione scegliere Parameter groups (Gruppi di parametri).
3. Scegliere il gruppo di parametri usati dall'istanza database che si desidera modificare.

   Non è consentito modificare un gruppo di parametri predefinito. Se l'istanza database è usata da un gruppo di parametri predefinito, creare un nuovo gruppo di parametri e associarlo all'istanza database.

   Per ulteriori informazioni sui gruppi di parametri database, consulta Utilizzo di gruppi di parametri.
4. Per Parameter group actions (Operazioni del gruppo di parametri), scegliere Edit (Modifica).
5. Impostare il parametro `binlog_format` sul formato di logging binario scelto (ROW, STATEMENT o MIXED).
6. Scegliere Save Changes (Salva modifiche) per salvare gli aggiornamenti applicati al gruppo di parametri database.

Accesso ai log binari MariaDB

Puoi utilizzare la utility mysqlbinlog per scaricare log binari in formato di testo dalle istanze database MariaDB. Il log binario viene scaricato sul tuo computer locale. Per ulteriori informazioni sull'uso dell'utilità mysqlbinlog, consulta la pagina relativa all'uso di mysqlbinlog nella documentazione di MariaDB.

Per eseguire la utility mysqlbinlog su un'istanza Amazon RDS usa le seguenti opzioni:

- Specifica l'opzione `--read-from-remote-server`.
- `--host`: specifica il nome DNS dall'endpoint dell'istanza.
- `--port`: specifica la porta utilizzata dall'istanza.
- `--user`: specifica un utente di MariaDB al quale è stata concessa l'autorizzazione slave permission.
- `--password`: specifica la password per l'utente o ometti un valore di password affinché la utility ti chieda una password.
- `--result-file`: specifica il file locale che riceve l'output.
- Specifica il nome di uno o più file di log binari. Per ottenere un elenco dei log disponibili utilizza il comando SQL SHOW BINARY LOGS.

Per ulteriori informazioni sulle opzioni di mysqlbinlog, consulta la pagina relativa alle opzioni di mysqlbinlog nella documentazione di MariaDB.

Di seguito è riportato un esempio:

Per LinuxmacOS, oUnix:

```
mysqlbinlog \
    --read-from-remote-server \
    --host=mariadbinstance1.1234abcd.region.rds.amazonaws.com \
    --port=3306  \
    --user ReplUser \
    --password <password> \
    --result-file=/tmp/binlog.txt
```

Per Windows:

```
mysqlbinlog ^
    --read-from-remote-server ^
    --host=mariadbinstance1.1234abcd.region.rds.amazonaws.com ^
    --port=3306  ^
    --user ReplUser ^
    --password <password> ^
    --result-file=/tmp/binlog.txt
```

Amazon RDS in genere elimina un log binario appena possibile. Tuttavia, il log binario deve essere disponibile sull'istanza affinché mysqlbinlog possa accedervi. Per specificare il numero di ore per cui RDS deve mantenere i registri binari, usa la stored procedure `mysql.rds_set_configuration`. Specifica un periodo di tempo sufficiente per scaricare i log. Dopo l'impostazione del periodo di retention, monitora l'utilizzo dello storage per l'istanza database per assicurare che i log binari conservati non occupino troppo spazio di storage.

L'esempio seguente imposta il periodo di conservazione su 1 giorno.

```
call mysql.rds_set_configuration('binlog retention hours', 24);
```

Per visualizzare l'impostazione attuale, utilizza la procedura archiviata `mysql.rds_show_configuration`.

```
call mysql.rds_show_configuration;
```

Annotazione log binario

In una istanza database MariaDB puoi usare l'evento `Annotate_rows` per annotare un evento di riga con una copia della query SQL che ha causato l'evento. Questo approccio fornisce una funzionalità simile all'abilitazione del parametro `binlog_rows_query_log_events` su un'istanza database RDS per MySQL.

Puoi abilitare le annotazioni di log binarie a livello globale creando un gruppo di parametri personalizzati e impostando il parametro `binlog_annotate_row_events` su **1**.
Puoi anche abilitare le annotazioni a livello di sessione richiamando `SET SESSION binlog_annotate_row_events = 1`. Usa `replicate_annotate_row_events` per replicare le annotazioni di log binario all'istanza di replica se la registrazione binaria è abilitata. Non sono necessari privilegi speciali per usare queste impostazioni.

Di seguito è illustrato un esempio di transazione basata su riga in MariaDB. L'uso della registrazione basata su riga viene attivato impostando il livello di isolamento delle transazioni su read-committed.

```
CREATE DATABASE IF NOT EXISTS test;
USE test;
CREATE TABLE square(x INT PRIMARY KEY, y INT NOT NULL) ENGINE = InnoDB;
SET SESSION TRANSACTION ISOLATION LEVEL READ COMMITTED;
BEGIN
INSERT INTO square(x, y) VALUES(5, 5 * 5);
```

```
COMMIT;
```

Senza annotazioni, le voci del log binario per la transazione appaiono nel modo seguente:

```
BEGIN
/*!*/;
# at 1163
# at 1209
#150922  7:55:57 server id 1855786460  end_log_pos 1209         Table_map:
 `test`.`square` mapped to number 76
#150922  7:55:57 server id 1855786460  end_log_pos 1247         Write_rows: table id 76
 flags: STMT_END_F
### INSERT INTO `test`.`square`
### SET
###   @1=5
###   @2=25
# at 1247
#150922  7:56:01 server id 1855786460  end_log_pos 1274         Xid = 62
COMMIT/*!*/;
```

La seguente istruzione abilita le annotazioni a livello di sessione per questa stessa transazione e le disabilita dopo aver eseguito la transazione:

```
CREATE DATABASE IF NOT EXISTS test;
USE test;
CREATE TABLE square(x INT PRIMARY KEY, y INT NOT NULL) ENGINE = InnoDB;
SET SESSION TRANSACTION ISOLATION LEVEL READ COMMITTED;
SET SESSION binlog_annotate_row_events = 1;
BEGIN;
INSERT INTO square(x, y) VALUES(5, 5 * 5);
COMMIT;
SET SESSION binlog_annotate_row_events = 0;
```

Con le annotazioni, le voci del log binario per la transazione appaiono nel modo seguente:

```
BEGIN
/*!*/;
# at 423
# at 483
# at 529
#150922  8:04:24 server id 1855786460  end_log_pos 483  Annotate_rows:
#Q> INSERT INTO square(x, y) VALUES(5, 5 * 5)
```

```
#150922  8:04:24 server id 1855786460  end_log_pos 529  Table_map: `test`.`square`
 mapped to number 76
#150922  8:04:24 server id 1855786460  end_log_pos 567  Write_rows: table id 76 flags:
 STMT_END_F
### INSERT INTO `test`.`square`
### SET
###   @1=5
###   @2=25
# at 567
#150922  8:04:26 server id 1855786460  end_log_pos 594  Xid = 88
COMMIT/*!*/;
```

File di log di database Microsoft SQL Server

Puoi accedere a log di errore, log dell'agente, file di traccia e file dump di Microsoft SQL Server tramite la console Amazon RDS, la AWS CLI o l'API RDS. Per ulteriori informazioni sulla visualizzazione, il download e la visione di log di database basati su file, consulta Monitoraggio dei file di log di Amazon RDS.

Argomenti

Pianificazione della conservazione

I file di log vengono ruotati ogni giorno e ogni volta che viene riavviata un'istanza database. Di seguito è illustrata la pianificazione della conservazione per i log di Microsoft SQL Server in Amazon RDS.

Tipo di log	Pianificazione della conservazione
Log di errore	Viene conservato un massimo di 30 log di errori. Amazon RDS può eliminare i log degli errori più vecchi di 7 giorni.
Log dell'agente	Viene conservato un massimo di 10 log dell'agente. Amazon RDS può eliminare i log dell'agente più vecchi di 7 giorni.
File di traccia	I file di traccia vengono conservati in base al periodo di conservazione dei file di traccia dell'istanza database. Il periodo di conservazione dei file di traccia predefinito è 7 giorni. Per modificare il periodo di conservazione dei file di traccia per l'istanza database, consulta Impostazione del periodo di retention dei file di traccia e dei file dump.
File dump	I file dump vengono conservati in base al periodo di conservazione dei file dump dell'istanza database. Il periodo di conservazione dei file dump predefinito è 7 giorni. Per modificare il periodo di conservazione dei file dump per l'istanza database, consulta Impostazione del periodo di retention dei file di traccia e dei file dump.

Visualizzazione del log di errore di SQL Server tramite la procedura archiviata rds_read_error_log

Puoi usare la stored procedure `rds_read_error_log` in Amazon RDS per visualizzare i log degli errori e i log dell'agente. Per ulteriori informazioni, consulta Visualizzazione dei log dell'agente e degli errori.

Pubblicazione dei log di SQL Server su Amazon CloudWatch Logs

Con Amazon RDS for SQL Server, puoi pubblicare gli eventi dei log degli errori e degli agenti direttamente su CloudWatch Amazon Logs. Analizza i dati di log con CloudWatch Logs, quindi utilizzali CloudWatch per creare allarmi e visualizzare i parametri.

Con CloudWatch Logs, puoi fare quanto segue:

- Conservare i log in uno spazio di storage estremamente durevole con un periodo di retention che definisci tu.
- Ricerca e filtraggio dei dati di log.
- Condivisione dei dati di log tra account.
- Esportare log in Simple Storage Service (Amazon S3).
- Trasmetti dati ad Amazon OpenSearch Service.
- Elaborare dati di log in tempo reale con Amazon Kinesis Data Streams. Per ulteriori informazioni, consulta Working with Amazon CloudWatch Logs nella Amazon Managed Service for Apache Flink for SQL Applications Developer Guide.

Amazon RDS pubblica ogni log di database SQL Server come flusso di database separato nel gruppo di log. Ad esempio, se pubblichi i log degli agenti e i log degli errori, i dati di errore vengono archiviati in un flusso di log degli errori nel gruppo di log e i dati dei `/aws/rds/instance/`*`my_instance`*`/error` log degli agenti vengono archiviati nel gruppo di log. `/aws/rds/instance/`*`my_instance`*`/agent`

Per le istanze database multi-AZ, Amazon RDS pubblica il log del database come due flussi separati nel gruppo di log. Ad esempio, se pubblichi log di errori, i dati degli errori vengono conservati nei flussi di log `/aws/rds/instance/`*`my_instance.node1`*`/error` e `/aws/rds/instance/`*`my_instance.node2`*`/error` rispettivamente. I flussi di log non cambiano durante un failover e il flusso di log degli errori di ogni nodo può contenere i log degli errori provenienti da un'istanza primaria o secondaria. Con Multi-AZ, viene creato automaticamente un flusso di log per

`/aws/rds/instance/my_instance/rds-events` archiviare i dati degli eventi, come i failover delle istanze DB.

> **Note**
>
> La pubblicazione dei log di SQL Server su CloudWatch Logs non è abilitata per impostazione predefinita. Non è supportata la pubblicazione di trace e dump file. La pubblicazione dei log di SQL Server su CloudWatch Logs è supportata in tutte le aree geografiche, ad eccezione dell'Asia Pacifico (Hong Kong).

Console

Per pubblicare i log di SQL Server DB nei registri da CloudWatch AWS Management Console

1. Apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, scegliere Databases (Database) e selezionare l'istanza database da modificare.
3. Scegliere Modify (Modifica).
4. Nella sezione Esportazioni dei log, scegli i log che desideri iniziare a pubblicare su Logs. CloudWatch

   È possibile scegliere Log agente, Log errori o entrambi.
5. Scegliere Continue (Continua) e quindi Modify DB Instance (Modifica istanza database) nella pagina di riepilogo.

AWS CLI

Per pubblicare i log SQL Server, puoi utilizzare il comando `modify-db-instance` con i parametri seguenti:

- `--db-instance-identifier`
- `--cloudwatch-logs-export-configuration`

> **Note**
>
> Viene sempre applicata all'istanza database una modifica all'opzione `--cloudwatch-logs-export-configuration` immediatamente. Pertanto, le opzioni `--apply-immediately` e `--no-apply-immediately` non hanno alcun effetto.

Puoi pubblicare i log SQL Server anche utilizzando i seguenti comandi:

- `create-db-instance`
- `restore-db-instance-from-db-snapshot`
- `restore-db-instance-to-point-in-time`

Example

L'esempio seguente crea un'istanza DB di SQL Server con la pubblicazione dei CloudWatch log abilitata. Il valore `--enable-cloudwatch-logs-exports`è un array di stringhe JSON che comprende `error`, `agent`, o entrambi.

Per LinuxmacOS, oUnix:

```
aws rds create-db-instance \
    --db-instance-identifier mydbinstance \
    --enable-cloudwatch-logs-exports '["error","agent"]' \
    --db-instance-class db.m4.large \
    --engine sqlserver-se
```

Per Windows:

```
aws rds create-db-instance ^
    --db-instance-identifier mydbinstance ^
    --enable-cloudwatch-logs-exports "[\"error\",\"agent\"]" ^
    --db-instance-class db.m4.large ^
    --engine sqlserver-se
```

> **Note**
>
> Quando usi il prompt comandi di Windows, non devi inserire le doppie virgolette (") nel codice JSON precedendole con il backslash (\).

Example

L'esempio seguente modifica un'istanza DB di SQL Server esistente per pubblicare i file di registro in CloudWatch Logs. Il valore `--cloudwatch-logs-export-configuration` è un oggetto JSON. La chiave per questo oggetto è `EnableLogTypes`, e il suo valore è un array di stringhe che può include`reerror`, `agent`, o entrambi.

PerLinux, omacOS: Unix

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --cloudwatch-logs-export-configuration '{"EnableLogTypes":["error","agent"]}'
```

Per Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --cloudwatch-logs-export-configuration "{\"EnableLogTypes\":[\"error\",\"agent\"]}"
```

> **Note**
>
> Quando usi il prompt comandi di Windows, non devi inserire le doppie virgolette (") nel codice JSON precedendole con il backslash (\).

Example

L'esempio seguente modifica un'istanza DB di SQL Server esistente per disabilitare i file di registro dell'agente di pubblicazione in CloudWatch Logs. Il valore `--cloudwatch-logs-export-configuration` è un oggetto JSON. La chiave per questo oggetto è `DisableLogTypes`, e il suo valore è un array di stringhe che può include`reerror`, `agent`, o entrambi.

PerLinux, omacOS: Unix

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --cloudwatch-logs-export-configuration '{"DisableLogTypes":["agent"]}'
```

Per Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --cloudwatch-logs-export-configuration "{\"DisableLogTypes\":[\"agent\"]}"
```

> **Note**
>
> Quando usi il prompt comandi di Windows, non devi inserire le doppie virgolette (") nel codice JSON precedendole con il backslash (\).

File di log del database MySQL

Puoi monitorare i log MySQL direttamente tramite la console Amazon RDS, l'API di Amazon RDS, AWS CLI o gli SDK AWS. Puoi anche eseguire l'accesso ai log MySQL indirizzando i log a una tabella del database nel database principale e facendo una ricerca in tale tabella. Puoi utilizzare la utility mysqlbinlog per scaricare un log binario.

Per ulteriori informazioni sulla visualizzazione, il download e la visione di log di database basati su file, consulta Monitoraggio dei file di log di Amazon RDS.

Argomenti

Panoramica dei registri di database RDS per MySQL

Puoi monitorare i seguenti tipi di file di registro RDS per MySQL:

- Log di errori
- Log delle query lente
- Log generale
- Log di audit

Il registro degli errori RDS per MySQL viene generato per impostazione predefinita. È possibile generare la query lenta e i log generali impostando i parametri nel gruppo di parametri di database.

Argomenti

Registri degli errori RDS per MySQL

RDS per MySQL scrive errori nel file `mysql-error.log`. Ogni file di log ha l'ora di creazione (in UTC) accodata al nome. I file di log hanno anche un timestamp che ti aiuta a determinare quando le voci del log sono state scritte.

RDS per MySQL scrive nel registro degli errori solo durante l'avvio, l'arresto e quando si verificano errori. Un'istanza database può andare avanti ore senza che ci siano nuove voci scritte nel file di log degli errori. Se non vedi voci recenti, significa che il server non ha riscontrato errori che generano una voce di registro.

In base alla progettazione, i registri degli errori vengono filtrati in modo da visualizzare solo eventi imprevisti come errori. Tuttavia, i registri degli errori contengono anche altre informazioni sul database, ad esempio l'avanzamento della query, che non vengono visualizzate. Pertanto, anche senza errori effettivi, la dimensione dei registri degli errori potrebbe aumentare a causa delle attività del database in corso. E anche quando presentano una dimensione in byte o kilobyte nella AWS Management Console, i log degli errori potrebbero avere 0 byte quando li scarichi.

RDS per MySQL scrive `mysql-error.log` su disco ogni 5 minuti. Aggiunge il contenuto del registro a `mysql-error-running.log`.

RDS per MySQL ruota il file `mysql-error-running.log` ogni ora. Conserva i registri generati nelle ultime due settimane.

> **Note**
>
> Il periodo di conservazione dei log è diverso tra Amazon RDS e Aurora.

Registri generali e delle query lente di RDS per MySQL

Il registro delle query lente e il registro generale di RDS per MySQL possono essere scritti in un file o una tabella di database impostando i parametri nel gruppo parametri del database. Per informazioni sulla creazione e la modifica di un gruppo di parametri database, consulta Utilizzo di gruppi di parametri. Devi impostare questi parametri prima di poter visualizzare il log delle query lente o il log generale nella console Amazon RDS o tramite l'API di Amazon RDS, la CLI di Amazon RDS o gli SDK AWS.

Puoi controllare la registrazione di RDS per MySQL utilizzando i parametri in questo elenco:

- `slow_query_log`: per creare il log delle query lente, imposta su 1. Il valore predefinito è 0.

- `general_log`: per creare il log generale, imposta su 1. Il valore predefinito è 0.
- `long_query_time`: per evitare che le query a esecuzione rapida vengano registrate nel registro delle query lente, specifica in secondi un valore per il runtime di query più breve da registrare. Il valore predefinito è 10 secondi, il minimo è 0 secondi. Se log_output = FILE, puoi specificare un valore in virgola mobile con risoluzione al microsecondo. Se log_output = TABLE, devi specificare un valore intero con risoluzione al secondo. Vengono registrate solo le query con runtime che supera il valore `long_query_time`. Ad esempio, impostando `long_query_time` su 0,1 si impedisce a tutte le query con tempo di esecuzione inferiore a 100 millisecondi di essere registrate.
- `log_queries_not_using_indexes`: per registrare tutte le query che non usano un indice sul log delle query lente, imposta su 1. Le query che non utilizzano un indice vengono registrate anche se il runtime è inferiore al valore del parametro `long_query_time`. Il valore predefinito è 0.
- `log_output` *`option`*: puoi specificare una delle seguenti opzioni per il parametro `log_output`.
  - TABLE (predefinito) `mysql.general_log` Scrive le query generali nella tabella – e le query lente nella tabella `mysql.slow_log`.
  - FILE – Scrive sia i log generali sia i log delle query lente nel file system.
  - NONE – Disabilita il logging.

Per ulteriori informazioni sui log delle query lente e i log generali, consulta i seguenti argomenti nella documentazione di MySQL:

- Log delle query lente
- Log delle query generali

Registro di controllo di MySQL

Per accedere al log di audit, l'istanza database deve usare un gruppo di opzioni personalizzato con l'opzione `MARIADB_AUDIT_PLUGIN`. Per ulteriori informazioni, consulta Supporto per MySQL del plug-in per audit MariaDB.

Rotazione e conservazione dei registri per RDS per MySQL

Quando la registrazione è abilitata, Amazon RDS ruota i log delle tabelle o elimina i file di log a intervalli regolari. Questa è una misura preventiva per ridurre l'eventualità che un file di log molto grande comprometta l'uso del database o la performance. RDS per MySQL gestisce la rotazione e l'eliminazione come segue:

- Le dimensioni dei log delle query lente, degli errori e generale di MySQL sono limitate a un massimo del 2 per cento dello spazio di storage assegnato per un'istanza database. Per mantenere questa soglia, i log vengono ruotati automaticamente ogni ora. MySQL rimuove i file di registro più vecchi di due settimane. Se le dimensioni del file di log combinato superano tale soglia dopo la rimozione dei file di log più vecchi, i file di log più grandi vengono eliminati fino a che le dimensioni del file di log non rimangono inferiori alla soglia.
- Quando la registrazione `FILE` è abilitata, i file di registro vengono esaminati ogni ora e quelli più vecchi di due settimane vengono eliminati. In alcuni casi, la dimensione del file di log combinato restante dopo l'eliminazione supera la soglia del 2 per cento di spazio assegnato a un'istanza database. In questi casi, i file di log più vecchi vengono eliminati fino a che le dimensioni del file di log non rimangono inferiori alla soglia.
- Quando la registrazione `TABLE` è abilitata, in alcuni casi, le tabelle di log vengono ruotate ogni 24 ore. Questa rotazione avviene se lo spazio usato dai registri delle tabelle è più del 20% dello spazio di archiviazione assegnato oppure se la dimensione di tutti i registri combinati è maggiore di 10 GB. Se la quantità di spazio utilizzato per un'istanza database è maggiore del 90 per cento dello spazio di storage assegnato per l'istanza database, allora le soglie di rotazione del log vengono ridotte. Le tabelle dei registri vengono ruotate se lo spazio utilizzato dai registri delle tabelle supera il 10% dello spazio di archiviazione assegnato oppure se la dimensione di tutti i registri combinati è maggiore di 5 GB. Puoi iscriverti all'evento `low_free_storage` per ricevere notifica quando le tabelle di log vengono ruotate per liberare spazio. Per ulteriori informazioni, consulta Utilizzo della notifica degli eventi di Amazon RDS.

  Quando le tabelle dei registri vengono ruotate, la tabella dei registri corrente viene copiata prima in una tabella dei registri di backup. Quindi le voci della tabella dei registri corrente vengono rimosse. Se esiste già una tabella di log di backup, questa viene eliminata prima che la tabella di log corrente sia copiata nel backup. Puoi eseguire una query sulla tabella di log di backup, se necessario. La tabella di log di backup per la tabella `mysql.general_log` è denominata `mysql.general_log_backup`. La tabella di log di backup per la tabella `mysql.slow_log` è denominata `mysql.slow_log_backup`.

  Puoi ruotare la tabella `mysql.general_log` chiamando la procedura `mysql.rds_rotate_general_log`. Puoi ruotare la tabella `mysql.slow_log` chiamando la procedura `mysql.rds_rotate_slow_log`.

  I log della tabella vengono ruotati durante l'aggiornamento della versione del database.

Per usare i log dalla console Amazon RDS, dall'API di Amazon RDS, dalla CLI di Amazon RDS o dagli SDK AWS, imposta il parametro `log_output` su FILE. Come il log degli errori MySQL, questi file di log vengono ruotati ogni ora. I file di registro generati durante le due settime precedenti vengono conservati. Il periodo di conservazione è diverso tra Amazon RDS e Aurora.

Limiti di dimensioni nei registri di ripristino

Per RDS for MySQL versione 8.0.32 e precedenti, il valore predefinito di questo parametro è 256 MB. Questo importo viene derivato moltiplicando il valore predefinito del `innodb_log_file_size` parametro (128 MB) per il valore predefinito del parametro (2). `innodb_log_files_in_group` Per ulteriori informazioni, consulta Best practice per la configurazione dei parametri per Amazon RDS for MySQL, parte 1: Parametri relativi alle prestazioni.

A partire dalla versione 8.0.33 di RDS per MySQL, Amazon RDS utilizza il parametro anziché il parametro. `innodb_redo_log_capacity` `innodb_log_file_size` Il valore predefinito di Amazon RDS del `innodb_redo_log_capacity` parametro è 2 GB. Per ulteriori informazioni, consulta Changes in MySQL 8.0.30 (Modifiche in MySQL 8.0.30) nella documentazione di MySQL.

Pubblicazione dei log MySQL su Amazon Logs CloudWatch

Puoi configurare la tua istanza DB MySQL per pubblicare i dati di log in un gruppo di log in Amazon Logs. CloudWatch Con CloudWatch Logs, puoi eseguire analisi in tempo reale dei dati di log e utilizzarli CloudWatch per creare allarmi e visualizzare metriche. È possibile utilizzare CloudWatch Logs per archiviare i record di registro in un archivio altamente durevole.

Amazon RDS pubblica ogni log di database MySQL come flusso di database separato nel gruppo di log. Ad esempio, se configuri la funzione di esportazione affinché includa il log delle query lente, i dati relativi alle query lente vengono archiviati in un flusso delle log delle query lente nel gruppo di log `/aws/rds/instance/my_instance/slowquery`.

Il log degli errori è abilitato per impostazione predefinita. La tabella seguente fornisce un riepilogo dei requisiti per gli altri log MySQL.

Log	Requisito
Log di controllo	L'istanza database deve usare un gruppo di opzioni personalizzato con l'opzione `MARIADB_AUDIT_PLUGIN` .

Log	Requisito
Log generale	L'istanza database deve usare un gruppo di parametri personalizzato con l'impostazione `general_log = 1` per abilitare il log generale.
Log delle query lente	L'istanza database deve usare un gruppo di parametri personalizzato con l'impostazione `slow_query_log = 1` per abilitare il log delle query lente.
Output log	L'istanza DB deve utilizzare un gruppo di parametri personalizzato con l'impostazione dei parametri `log_output = FILE` per scrivere i log nel file system e pubblicarli CloudWatch nei registri.

Console

Per pubblicare i log MySQL su Logs utilizzando CloudWatch la console

1. Apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, scegliere Databases (Database) e selezionare l'istanza database da modificare.
3. Scegliere Modify (Modifica).
4. Nella sezione Esportazioni dei log, scegli i log che desideri iniziare a pubblicare su Logs. CloudWatch
5. Scegliere Continue (Continua) e quindi Modify DB Instance (Modifica istanza database) nella pagina di riepilogo.

AWS CLI

Puoi pubblicare i log MySQL con la AWS CLI. Puoi chiamare il comando `modify-db-instance` con i parametri seguenti:

- `--db-instance-identifier`

- `--cloudwatch-logs-export-configuration`

> **Note**
>
> Viene sempre applicata all'istanza database una modifica all'opzione `--cloudwatch-logs-export-configuration` immediatamente. Pertanto, le opzioni `--apply-immediately` e `--no-apply-immediately` non hanno alcun effetto.

Puoi pubblicare i log MySQL anche chiamando i seguenti comandi della AWS CLI:

- `create-db-instance`
- `restore-db-instance-from-db-snapshot`
- `restore-db-instance-from-s3`
- `restore-db-instance-to-point-in-time`

Esegui uno di questi comandi dell'AWS CLI con le opzioni seguenti:

- `--db-instance-identifier`
- `--enable-cloudwatch-logs-exports`
- `--db-instance-class`
- `--engine`

Altre opzioni potrebbero essere richieste a seconda del comando AWS CLI eseguito.

Example

L'esempio seguente modifica un'istanza database MySQL esistente per pubblicare i file di registro in Logs. CloudWatch Il valore `--cloudwatch-logs-export-configuration` è un oggetto JSON. La chiave per questo oggetto è `EnableLogTypes` e il suo valore è una matrice di stringhe con qualsiasi combinazione di `audit`, `error`, `general` e `slowquery`.

PerLinux, o: macOS Unix

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
```

```
    --cloudwatch-logs-export-configuration '{"EnableLogTypes":
["audit","error","general","slowquery"]}'
```

Per Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --cloudwatch-logs-export-configuration '{"EnableLogTypes":
["audit","error","general","slowquery"]}'
```

Example

L'esempio seguente crea un'istanza DB MySQL e pubblica i file di registro in Logs. CloudWatch Il valore `--enable-cloudwatch-logs-exports` è una matrice di stringhe JSON. Le stringhe possono essere una qualsiasi combinazione di `audit`, `error`, `general` e `slowquery`.

PerLinux, o: macOS Unix

```
aws rds create-db-instance \
    --db-instance-identifier mydbinstance \
    --enable-cloudwatch-logs-exports '["audit","error","general","slowquery"]' \
    --db-instance-class db.m4.large \
    --engine MySQL
```

Per Windows:

```
aws rds create-db-instance ^
    --db-instance-identifier mydbinstance ^
    --enable-cloudwatch-logs-exports '["audit","error","general","slowquery"]' ^
    --db-instance-class db.m4.large ^
    --engine MySQL
```

API RDS

Puoi pubblicare i log MySQL con RDS API. Puoi chiamare l'operazione `ModifyDBInstance` con i parametri seguenti:

- `DBInstanceIdentifier`
- `CloudwatchLogsExportConfiguration`

> **Note**
>
> Viene sempre applicata all'istanza database una modifica al parametro `CloudwatchLogsExportConfiguration` immediatamente. Pertanto, il parametro `ApplyImmediately` non ha alcun effetto.

Puoi pubblicare i log MySQL anche chiamando le seguenti operazioni API RDS:

- `CreateDBInstance`
- `RestoreDBInstanceFromDBSnapshot`
- `RestoreDBInstanceFromS3`
- `RestoreDBInstanceToPointInTime`

Esegui una di queste azioni API RDS con i seguenti parametri:

- `DBInstanceIdentifier`
- `EnableCloudwatchLogsExports`
- `Engine`
- `DBInstanceClass`

Altri parametri potrebbero essere richiesti a seconda del comando della AWS CLI che viene eseguito.

Gestione dei log MySQL basati su tabella

Puoi indirizzare il log generale e il log delle query lente alle tabelle sull'istanza database creando un gruppo di parametri del database e impostando il parametro server `log_output` su `TABLE`. Le query generali vengono quindi registrate sulla tabella `mysql.general_log`, mentre le query lente vengono registrate sulla tabella `mysql.slow_log`. Puoi eseguire query sulle tabelle per avere accesso alle informazioni di log. L'abilitazione di questa registrazione aumenta il numero di dati scritti sul database, il che potrebbe compromettere le performance.

Sia il log generale che quello delle query lente sono disattivati per impostazione predefinita. Per abilitare la registrazione sulle tabelle devi impostare anche i parametri server `general_log` e `slow_query_log` su `1`.

Le tabelle di log continuano a crescere fino a che le rispettive attività di registrazione non vengono disattivate eseguendo la reimpostazione del parametro appropriato su `0`. Spesso nel corso del tempo si accumulano grandi quantità di dati che possono usare una percentuale considerevole dello spazio di archiviazione assegnato. Amazon RDS non consente di troncare le tabelle dei registri, ma è possibile spostarne il contenuto. La rotazione delle tabelle ne salva il contenuto in una tabella di backup e crea una nuova tabella di log vuota. Puoi ruotare manualmente le tabelle di log con le seguenti procedure a riga di comando, nelle quali il prompt dei comandi è indicato da `PROMPT>`:

```
PROMPT> CALL mysql.rds_rotate_slow_log;
PROMPT> CALL mysql.rds_rotate_general_log;
```

Per rimuovere completamente i dati vecchi e recuperare lo spazio del disco, chiama la procedura adeguata due volte in successione.

Configurazione del log binario di MySQL

Il log binario è un insieme di file di log che contengono informazioni sulle modifiche apportate ai dati di un'istanza server MySQL. Il log binario contiene informazioni come le seguenti:

- Eventi che descrivono le modifiche al database come la creazione di tabelle o la modifica di righe
- Informazioni sulla durata di ogni istruzione che ha aggiornato i dati
- Eventi per istruzioni che avrebbero potuto aggiornare i dati ma non l'hanno fatto

Il log binario registra le istruzioni inviate durante la replica. È inoltre necessario per alcune operazioni di ripristino. Per ulteriori informazioni, consulta The Binary Log (Il log binario) e Binary Log Overview (Panoramica sul log binario) nella documentazione di MySQL.

La caratteristica di backup automatici determina se il log binario è attivato o disattivato per MySQL. Sono disponibili le seguenti opzioni:

Attivazione del log binario

Impostare il tempo di conservazione del backup su un valore positivo diverso da zero.

Disattivazione del log binario

Impostare il tempo di conservazione del backup su zero.

Per ulteriori informazioni, consulta Abilitazione dei backup automatici.

MySQL su Amazon RDS supporta i formati di logging binario basati su righe, basati su istruzioni e misti. Si consiglia il formato misto a meno che non sia necessario un formato binlog specifico. Per dettagli sui diversi formati di log binario MySQL, consulta Binary logging formats (Formati di log binari) nella documentazione MySQL.

Se pianifichi di utilizzare la replica, il formato di logging binario è importante in quanto determina il record delle modifiche dei dati che viene registrato nella sorgente e inviato ai target della replica. Per ulteriori informazioni sui vantaggi e sugli svantaggi dei vari formati di logging binario per la replica, consulta la pagina relativa a vantaggi e svantaggi della replica basata su istruzioni e basata su riga nella documentazione di MySQL.

> **Important**
>
> L'impostazione del formato di registrazione binario su "basato su riga" può generare file di log binari molto grandi. I file di log binari di grandi dimensioni riducono lo spazio di storage disponibile per un'istanza database e possono determinare un aumento della quantità di tempo necessaria per eseguire un'operazione di ripristino di un'istanza database.
> La replica basata sulle istruzioni può causare incoerenze tra l'istanza database di origine e una replica di lettura. Per ulteriori informazioni, consulta la pagina relativa alla determinazione delle istruzioni sicure e non sicure nel logging binario nella documentazione MySQL.
> Abilitando la registrazione binaria, aumenta il numero delle operazioni I/O di scrittura sul disco nell'istanza database. Puoi monitorare l'utilizzo degli IOPS con la `WriteIOPS` CloudWatch metrica.

Per impostare il formato di registrazione binaria MySQL

1. Apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel pannello di navigazione scegliere Parameter groups (Gruppi di parametri).
3. Scegliete il gruppo di parametri del DB, associato al di istanze DB, che desiderate modificare.

Non è consentito modificare un gruppo di parametri predefinito. Se l'istanza database è usata da un gruppo di parametri predefinito, creare un nuovo gruppo di parametri e associarlo all'istanza database.

Per ulteriori informazioni sui gruppi di parametri, consulta Utilizzo di gruppi di parametri.

4. Da Azioni, scegli Modifica.
5. Imposta il parametro `binlog_format` sul formato di registrazione binaria scelto (`ROW`, `STATEMENT` o `MIXED`).

   Puoi disattivare la registrazione binaria impostando il periodo di conservazione dei backup di un'istanza database su zero, ma ciò disabilita i backup automatici giornalieri. La disabilitazione dei backup automatici disattiva o disabilita la variabile di sessione. `log_bin` Ciò disabilita la registrazione binaria sull'istanza DB RDS for MySQL, che a sua volta reimposta la variabile di `binlog_format` sessione al valore predefinito di nel database. `ROW` Si consiglia di non disabilitare i backup. Per ulteriori informazioni sull'impostazione Periodo di conservazione dei backup, consulta Impostazioni per istanze database.
6. Scegliere Salva modifiche per salvare gli aggiornamenti applicati al gruppo di parametri database.

Poiché il parametro `binlog_format` è dinamico, non è necessario riavviare l'istanza database per applicare le modifiche.

> **Important**
>
> La modifica di un gruppo di parametri database influisce su tutte le istanze database che utilizzano tale gruppo di parametri. Se si desidera specificare diversi formati di logging binario per diverse istanze database MySQL in una regione AWS, le istanze database devono utilizzare gruppi di parametri database diversi. Questi gruppi di parametri identificano diversi formati di logging. Assegnare il gruppo di parametri database appropriato a ciascuna istanza database.

Accesso ai log binari MySQL

Puoi utilizzare la utility mysqlbinlog per il download o lo streaming di log binari dalle istanze database RDS for MySQL. Il log binario viene scaricato sul computer locale dove è possibile

eseguire operazioni come la riproduzione del log tramite utility mysql. Per ulteriori informazioni sull'uso dell'utilità mysqlbinlog, consulta Utilizzo di mysqlbinlog per il backup di file di log binari nella documentazione di MySQL.

Per eseguire la utility mysqlbinlog su un'istanza Amazon RDS usa le seguenti opzioni:

- `--read-from-remote-server` - Obbligatorio
- `--host`: il nome DNS dall'endpoint dell'istanza.
- `--port`: la porta utilizzata dall'istanza.
- `--user`: un utente MySQL al quale è stata concessa l'autorizzazione `REPLICATION SLAVE`.
- `--password`: la password dell'utente MySQL oppure ometti un valore di password affinché l'utilità richieda una password.
- `--raw`: scarica il file in formato binario.
- `--result-file`: il file locale per riceve l'output raw.
- `--stop-never`: trasmette in streaming i file di log binari.
- `--verbose`: quando utilizzi il formato binlog `ROW`, includi questa opzione per visualizzare gli eventi di riga come istruzioni pseudo-SQL. Per ulteriori informazioni sull'opzione `--verbose`, consulta Visualizzazione degli eventi di riga di mysqlbinlog nella documentazione di MySQL.
- Specifica il nome di uno o più file di log binari. Per ottenere l'elenco dei log disponibili, utilizza il comando SQL `SHOW BINARY LOGS`.

Per ulteriori informazioni sulle opzioni di mysqlbinlog, consulta Utilità mysqlbinlog per l'elaborazione di file di log binari nella documentazione di MySQL.

Gli esempi seguenti mostrano come utilizzare l'utilità mysqlbinlog.

Per LinuxmacOS, oUnix:

```
mysqlbinlog \
    --read-from-remote-server \
    --host=MySQLInstance1.cg034hpkmmjt.region.rds.amazonaws.com \
    --port=3306  \
    --user ReplUser \
    --password \
    --raw \
    --verbose \
    --result-file=/tmp/ \
```

```
    binlog.00098
```

Per Windows:

```
mysqlbinlog ^
    --read-from-remote-server ^
    --host=MySQLInstance1.cg034hpkmmjt.region.rds.amazonaws.com ^
    --port=3306  ^
    --user ReplUser ^
    --password ^
    --raw ^
    --verbose ^
    --result-file=/tmp/ ^
    binlog.00098
```

Amazon RDS in genere elimina un log binario appena possibile, ma il log binario deve essere ancora disponibile sull'istanza affinché mysqlbinlog possa accedervi. Per specificare il numero di ore che RDS deve rispettare per conservare i log binari usa la procedura archiviata mysql.rds_set_configuration e specifica un periodo abbastanza lungo che ti consenta di scaricare i log. Dopo l'impostazione del periodo di retention, monitora l'utilizzo dello storage per l'istanza database per assicurare che i log binari conservati non occupino troppo spazio di storage.

L'esempio seguente imposta il periodo di conservazione su 1 giorno.

```
call mysql.rds_set_configuration('binlog retention hours', 24);
```

Per visualizzare l'impostazione attuale, utilizza la procedura archiviata mysql.rds_show_configuration.

```
call mysql.rds_show_configuration;
```

File di log del database Oracle

Puoi accedere ai log di avviso, ai file di audit e ai file di traccia Oracle tramite console Amazon RDS o API. Per ulteriori informazioni sulla visualizzazione, il download e la visione di log di database basati su file, consulta Monitoraggio dei file di log di Amazon RDS.

I file di verifica Oracle forniti sono i file di verifica Oracle standard. Amazon RDS supporta la caratteristica FGA (Fine-Grained Auditing) di Oracle. Tuttavia, l'accesso ai log non fornisce accesso agli eventi FGA archiviati nella tabella `SYS.FGA_LOG$` e accessibili tramite la vista `DBA_FGA_AUDIT_TRAIL`.

L'operazione API `DescribeDBLogFiles` che elenca i file di log Oracle disponibili per una istanza database ignora il parametro `MaxRecords` e restituisce fino a 1.000 record. La chiamata restituisce `LastWritten` come data POSIX in millisecondi.

Argomenti

- Pianificazione della conservazione
- Utilizzo di file di traccia Oracle
- Pubblicazione dei log Oracle su Amazon CloudWatch Logs
- Metodi precedenti per l'accesso ai log di avvisi e ai log del listener

Pianificazione della conservazione

Il motore database Oracle potrebbe ruotare i file di log nel caso in cui diventino molto grandi. Per conservare i file di audit o di traccia è necessario scaricarli. Se archivi i file localmente riduci i costi di storage di Amazon RDS e rendi più spazio disponibile per i dati.

La seguente tabella illustra la pianificazione di conservazione per i log di avviso, i file di audit e i file di traccia Oracle su Amazon RDS.

Tipo di log	Pianificazione della conservazione
Log di avvisi	Il log degli avvisi di testo viene ruotato ogni giorno con conservazione per 30 giorni gestita da Amazon RDS. Il log di avviso XML viene conservat o per un minimo di sette giorni. Puoi accedere a questo log usando la visualizzazione `ALERTLOG`.

Tipo di log	Pianificazione della conservazione
File di audit	Il periodo di conservazione predefinito per i file di verifica è sette giorni. Amazon RDS può eliminare i file di verifica più vecchi di sette giorni.
File di traccia	Il periodo di conservazione dei file di traccia predefinito è 7 giorni. Amazon RDS può eliminare i file di traccia più vecchi di sette giorni.
Log del listener	Il periodo di conservazione predefinito per i log dei listener è sette giorni. Amazon RDS può eliminare i log del listener più vecchi di sette giorni.

> **Note**
>
> I file di audit e i file di traccia condividono la stessa configurazione di conservazione.

Utilizzo di file di traccia Oracle

Di seguito si riportano le descrizioni delle procedure Amazon RDS per creare, aggiornare ed eliminare file di traccia.

Argomenti

- Elenco di file
- Creazione di file di traccia e tracciamento di una sessione
- Recupero di file di traccia
- Eliminazione di file di traccia

Elenco di file

È possibile utilizzare una delle due procedure per consentire l'accesso a qualsiasi file nel percorso `background_dump_dest`. La prima procedura aggiorna una vista contenente un elenco di tutti i file attualmente in `background_dump_dest`.

```
EXEC rdsadmin.manage_tracefiles.refresh_tracefile_listing;
```

Dopo l'aggiornamento della vista, eseguire la query della vista seguente per accedere ai risultati.

```
SELECT * FROM rdsadmin.tracefile_listing;
```

Un'alternativa al processo precedente è quella di utilizzare `FROM table` per lo streaming di dati non relazionali in un formato di tipo tabella per elencare i contenuti della directory database.

```
SELECT * FROM TABLE(rdsadmin.rds_file_util.listdir('BDUMP'));
```

La query seguente mostra il testo di un file di log.

```
SELECT text FROM
 TABLE(rdsadmin.rds_file_util.read_text_file('BDUMP','alert_dbname.log.date'));
```

In una replica di lettura, ottenere il nome della directory BDUMP eseguendo una query `V$DATABASE.DB_UNIQUE_NAME`. Se il nome univoco è `DATABASE_B`, allora la directory BDUMP è `BDUMP_B`. Nell'esempio seguente viene eseguita una query sul nome BDUMP in una replica e viene quindi utilizzato questo nome per eseguire una query sul contenuto di `alert_DATABASE.log.2020-06-23`.

```
SELECT 'BDUMP' || (SELECT regexp_replace(DB_UNIQUE_NAME,'.*(_[A-Z])', '\1') FROM V
$DATABASE) AS BDUMP_VARIABLE FROM DUAL;

BDUMP_VARIABLE
--------------
BDUMP_B

SELECT TEXT FROM
 table(rdsadmin.rds_file_util.read_text_file('BDUMP_B','alert_DATABASE.log.2020-06-23'));
```

Creazione di file di traccia e tracciamento di una sessione

Poiché `ALTER SESSION` non è soggetto a restrizioni, restano disponibili molti metodi standard per generare file di traccia in Oracle per un'istanza database Amazon RDS. Le procedure seguenti sono fornite per i file di traccia che richiedono maggiore accesso.

Metodo Oracle	Metodo Amazon RDS
`oradebug hanganalyze 3`	

Metodo Oracle	Metodo Amazon RDS
	`EXEC rdsadmin.manage_tracefiles.hanganalyze;`
`oradebug dump systemstate 266`	`EXEC rdsadmin.manage_tracefiles.dump_systemstate;`

Puoi utilizzare molti metodi standard per tracciare singole sessioni collegate all'istanza database Oracle in Amazon RDS. Per abilitare la traccia di una sessione puoi eseguire i sottoprogrammi in pacchetti PL/SQL forniti da Oracle, ad esempio `DBMS_SESSION` e `DBMS_MONITOR`. Per ulteriori informazioni, consulta la pagina relativa all'abilitazione della traccia per una sessione nella documentazione di Oracle.

Recupero di file di traccia

Puoi recuperare qualsiasi file di traccia in `background_dump_dest` usando una query SQL standard su una tabella esterna gestita da Amazon RDS–. Per utilizzare questo metodo, devi eseguire la procedura per impostare la posizione di questa tabella sul file di traccia specifico.

Ad esempio, puoi utilizzare la vista `rdsadmin.tracefile_listing` indicata precedentemente per elencare tutti i file di traccia sul sistema. Successivamente, puoi impostare la vista `tracefile_table` per puntare al file di traccia desiderato utilizzando la procedura seguente.

```
EXEC
 rdsadmin.manage_tracefiles.set_tracefile_table_location('CUST01_ora_3260_SYSTEMSTATE.trc');
```

L'esempio seguente crea una tabella esterna nello schema corrente con la posizione impostata sul file fornito. Puoi recuperare il contenuto in un file locale utilizzando una query SQL.

```
SPOOL /tmp/tracefile.txt
SELECT * FROM tracefile_table;
SPOOL OFF;
```

Eliminazione di file di traccia

I file di traccia possono accumularsi e occupare spazio su disco. Per impostazione predefinita, Amazon RDS elimina i file di traccia e i file di log che risalgono a più di sette giorni prima. Puoi

visualizzare e impostare il periodo di conservazione dei file di traccia tramite la procedura `show_configuration`. Dovresti eseguire il comando `SET SERVEROUTPUT ON` per poter visualizzare i risultati della configurazione.

L'esempio seguente mostra il periodo di conservazione dei file di traccia e imposta quindi un nuovo periodo di conservazione dei file di traccia.

```
# Show the current tracefile retention
SQL> EXEC rdsadmin.rdsadmin_util.show_configuration;
NAME:tracefile retention
VALUE:10080
DESCRIPTION:tracefile expiration specifies the duration in minutes before tracefiles in
 bdump are automatically deleted.

# Set the tracefile retention to 24 hours:
SQL> EXEC rdsadmin.rdsadmin_util.set_configuration('tracefile retention',1440);
SQL> commit;

#show the new tracefile retention
SQL> EXEC rdsadmin.rdsadmin_util.show_configuration;
NAME:tracefile retention
VALUE:1440
DESCRIPTION:tracefile expiration specifies the duration in minutes before tracefiles in
 bdump are automatically deleted.
```

Oltre al processo di eliminazione periodica, puoi rimuovere manualmente i file da `background_dump_dest`. L'esempio seguente mostra come eliminare tutti i file che risalgono a cinque minuti prima.

```
EXEC rdsadmin.manage_tracefiles.purge_tracefiles(5);
```

Puoi inoltre eliminare tutti i file che corrispondono a un modello specifico (non includono l'estensione del file, come .trc). L'esempio seguente mostra come eliminare tutti i file che iniziano con `SCHPOC1_ora_5935`.

```
EXEC rdsadmin.manage_tracefiles.purge_tracefiles('SCHPOC1_ora_5935');
```

Pubblicazione dei log Oracle su Amazon CloudWatch Logs

Puoi configurare la tua istanza DB RDS for Oracle per pubblicare i dati di log in un gruppo di log in Amazon CloudWatch Logs. Con CloudWatch Logs, puoi analizzare i dati di log e utilizzarli

CloudWatch per creare allarmi e visualizzare metriche. È possibile utilizzare CloudWatch Logs per archiviare i record di registro in un archivio altamente durevole.

Amazon RDS pubblica ogni log di database Oracle come flusso di database separato nel gruppo di log. Ad esempio, se configuri la funzione di esportazione affinché includa il log di audit, i dati relativi all'audit vengono archiviati in un flusso di log di audit nel gruppo di log `/aws/rds/instance/my_instance/audit`. La tabella seguente riassume i requisiti per la pubblicazione dei log su Amazon Logs da parte di RDS for Oracle. CloudWatch CloudWatch

Nome log	Requisito	Predefinita
Log di avviso	Nessuna. Non puoi disabilitare questo registro.	Abilitato
Log di traccia	Imposta il `trace_enabled` parametro su TRUE o lascialo impostato sul valore predefinito.	TRUE
Log di audit	Imposta il `audit_trail` parametro su uno dei seguenti valori consentiti: `{ none | os | db [, extended] | xml [, extended] }`	none
Log del listener	Nessuna. Non puoi disabilitare questo registro.	Abilitato
Log di Oracle Management Agent	Nessuna. Non puoi disabilitare questo registro.	Abilitato

Questo registro di Oracle Management Agent è costituito dai gruppi di log riportati nella tabella di seguito.

Nome log	CloudWatch gruppo di log
emctl.log	oemagent-emctl
emdctlj.log	oemagent-emdctlj

Nome log	CloudWatch gruppo di log
gcagent.log	oemagent-gcagent
gcagent_errors.log	oemagent-gcagent-errors
emagent.nohup	oemagent-emagent-nohup
secure.log	oemagent-sicuro

Per maggiori informazioni, consulta Individuazione dei file di log e di traccia di Management Agent nella documentazione Oracle.

Console

Per pubblicare i log di Oracle DB su CloudWatch Logs da AWS Management Console

1. Apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, scegliere Databases (Database) e selezionare l'istanza database da modificare.
3. Scegliere Modify (Modifica).
4. Nella sezione Esportazioni dei log, scegli i log che desideri iniziare a pubblicare su Logs. CloudWatch
5. Scegliere Continue (Continua) e quindi Modify DB Instance (Modifica istanza database) nella pagina di riepilogo.

AWS CLI

Per pubblicare i log Oracle, puoi utilizzare il comando `modify-db-instance` con i parametri seguenti:

- `--db-instance-identifier`
- `--cloudwatch-logs-export-configuration`

> **Note**
>
> Viene sempre applicata all'istanza database una modifica all'opzione `--cloudwatch-logs-export-configuration` immediatamente. Pertanto, le opzioni `--apply-immediately` e `--no-apply-immediately` non hanno alcun effetto.

Puoi pubblicare i log Oracle anche utilizzando i seguenti comandi:

- `create-db-instance`
- `restore-db-instance-from-db-snapshot`
- `restore-db-instance-from-s3`
- `restore-db-instance-to-point-in-time`

Example

L'esempio seguente crea un'istanza DB Oracle con la pubblicazione dei CloudWatch log abilitata. Il valore `--cloudwatch-logs-export-configuration` è una matrice di stringhe JSON. Le stringhe possono essere una qualsiasi combinazione di `alert`, `audit`, `listener` e `trace`.

Per LinuxmacOS, oUnix:

```
aws rds create-db-instance \
    --db-instance-identifier mydbinstance \
    --cloudwatch-logs-export-configuration
 '["trace","audit","alert","listener","oemagent"]' \
    --db-instance-class db.m5.large \
    --allocated-storage 20 \
    --engine oracle-ee \
    --engine-version 12.1.0.2.v18 \
    --license-model bring-your-own-license \
    --master-username myadmin \
    --manage-master-user-password
```

Per Windows:

```
aws rds create-db-instance ^
    --db-instance-identifier mydbinstance ^
    --cloudwatch-logs-export-configuration trace alert audit listener oemagent ^
```

```
    --db-instance-class db.m5.large ^
    --allocated-storage 20 ^
    --engine oracle-ee ^
    --engine-version 12.1.0.2.v18 ^
    --license-model bring-your-own-license ^
    --master-username myadmin ^
    --manage-master-user-password
```

Example

L'esempio seguente modifica un'istanza Oracle DB esistente per pubblicare i file di log in CloudWatch Logs. Il valore `--cloudwatch-logs-export-configuration` è un oggetto JSON. La chiave per questo oggetto è `EnableLogTypes` e il suo valore è una matrice di stringhe con qualsiasi combinazione di `alert`, `audit`, `listener` e `trace`.

PerLinux, omacOS: Unix

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --cloudwatch-logs-export-configuration '{"EnableLogTypes":
["trace","alert","audit","listener","oemagent"]}'
```

Per Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --cloudwatch-logs-export-configuration EnableLogTypes=\"trace\",\"alert\",\"audit
\",\"listener\",\"oemagent\"
```

Example

L'esempio seguente modifica un'istanza di Oracle DB esistente per disabilitare la pubblicazione dei file di audit e di log del listener in Logs. CloudWatch Il valore `--cloudwatch-logs-export-configuration` è un oggetto JSON. La chiave per questo oggetto è `DisableLogTypes` e il suo valore è una matrice di stringhe con qualsiasi combinazione di `alert`, `audit`, `listener` e `trace`.

PerLinux, o: macOS Unix

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --cloudwatch-logs-export-configuration '{"DisableLogTypes":["audit","listener"]}'
```

Per Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --cloudwatch-logs-export-configuration DisableLogTypes=\"audit\",\"listener\"
```

API RDS

Puoi pubblicare i log database Oracle con l'API RDS. Puoi chiamare l'operazione `ModifyDBInstance` con i parametri seguenti:

- `DBInstanceIdentifier`
- `CloudwatchLogsExportConfiguration`

> **Note**
>
> Viene sempre applicata all'istanza database una modifica al parametro `CloudwatchLogsExportConfiguration` immediatamente. Pertanto, il parametro `ApplyImmediately` non ha alcun effetto.

Puoi pubblicare i log Oracle anche chiamando le seguenti operazioni API RDS:

- `CreateDBInstance`
- `RestoreDBInstanceFromDBSnapshot`
- `RestoreDBInstanceFromS3`
- `RestoreDBInstanceToPointInTime`

Esegui una di queste azioni API RDS con i seguenti parametri:

- `DBInstanceIdentifier`
- `EnableCloudwatchLogsExports`
- `Engine`
- `DBInstanceClass`

Altri parametri potrebbero essere richiesti a seconda dell'operazione RDS eseguita.

Metodi precedenti per l'accesso ai log di avvisi e ai log del listener

Puoi visualizzare il log di avvisi tramite console Amazon RDS. Puoi inoltre utilizzare l'istruzione SQL per accedere al log di avvisi.

```
SELECT message_text FROM alertlog;
```

La visualizzazione `listenerlog` contiene le voci per Oracle Database versione 12.1.0.2 e precedenti. Per accedere al log del listener per queste versioni del database utilizza la seguente query.

```
SELECT message_text FROM listenerlog;
```

Per le versioni 12.2.0.1 e successive di Oracle Database, accedi al log del listener utilizzando Amazon Logs. CloudWatch

> **Note**
>
> Oracle ruota i log di avvisi e del listener quando superano 10 MB, punto in cui non sono disponibili dalle viste Amazon RDS.

File di log del database RDS per PostgreSQL

RDS per PostgreSQL registra le attività del database nel file di log PostgreSQL predefinito. Per un'istanza database PostgreSQL on-premise, questi messaggi vengono archiviati localmente in `log/postgresql.log`. Per un'istanza database RDS per PostgreSQL, il file di log è disponibile nell'istanza Amazon RDS. Inoltre, devi utilizzare la console Amazon RDS per visualizzarne o scaricarne il contenuto. Il livello di registrazione predefinito rileva gli errori di accesso, gli errori irreversibili del server, i deadlock e gli errori delle query.

Per ulteriori informazioni su come visualizzare, scaricare e guardare i registri di database basati su file, consulta Monitoraggio dei file di log di Amazon RDS. Per ulteriori informazioni sui registri PostgreSQL, consulta Working with Amazon RDS and Aurora PostgreSQL logs: Part 1 (Utilizzo dei registri RDS e Aurora PostgreSQL: parte 1) e Working with Amazon RDS and Aurora PostgreSQL logs: Part 2 (Utilizzo dei registri RDS e Aurora PostgreSQL: parte 2).

Oltre ai log PostgreSQL standard trattati in questo argomento, RDS per PostgreSQL supporta anche l'estensione di audit PostgreSQL (`pgAudit`). La maggior parte dei settori regolamentati e degli enti governativi deve mantenere un log di audit o un audit trail delle modifiche apportate ai dati per conformità ai requisiti legali. Per informazioni sull'installazione e sull'utilizzo di pgAudit, consulta Utilizzo di pgAudit per registrare l'attività del database.

Argomenti

- Parametri che influiscono sul comportamento della registrazione
- Attivazione della registrazione delle query per l'istanza database RDS per PostgreSQL
- Pubblicazione dei log PostgreSQL su Amazon Logs CloudWatch

Parametri che influiscono sul comportamento della registrazione

È possibile personalizzare il comportamento di registrazione per l'istanza database RDS per PostgreSQL modificando vari parametri. Nella tabella seguente sono riportati, tra le altre impostazioni, i parametri che stabiliscono la durata di archiviazione dei log, quando ruotarli e se l'output del log è in formato CSV (valori separati da virgole). Puoi anche trovare l'output di testo inviato a STDERR, tra le altre impostazioni. Per modificare le impostazioni per i parametri modificabili, utilizza un gruppo di parametri del·clusterdatabase personalizzato per l'istanza database RDS per PostgreSQL. Per ulteriori informazioni, consulta Utilizzo di gruppi di parametri DB in un'istanza DB. Come indicato nella tabella, `log_line_prefix` non può essere modificato.

Parametro	Predefinito	Descrizione
log_destination	stderr	Imposta il formato di output per il registro. L'impostazione predefinita è `stderr`, ma puoi anche specificare il formato CSV aggiungendo `csvlog` all'impostazione. Per ulteriori informazioni, consulta Impostazione della destinazione del registro (`stderr`, `csvlog`).
log_filename	postgresql.log.%Y-%m-%d-%H	Specifica il modello per il nome del file di log. Oltre al valore predefinito, questo parametro supporta `postgresql.log.%Y-%m-%d` per il modello del nome del file.
log_line_prefix	%t:%r:%u@%d:[%p]:	Definisce il prefisso per ogni riga di log che viene scritta in `stderr`, per annotare l'ora (%t), l'host remoto (%r), l'utente (%u), il database (%d) e l'ID del processo (%p). Non puoi modificare questo parametro.
log_rotation_age	60	I minuti dopo i quali il file di log viene ruotato automaticamente. Puoi modificare questo valore entro un intervallo compreso tra 1 e 1440 minuti. Per ulteriori informazioni, consulta Impostazione della rotazione dei file di log.
log_rotation_size	–	La dimensione (KB) che stabilisce la rotazione automatica del log. Per impostazione predefinita, questo parametro non viene utilizzato perché i log vengono ruotati in base al parametro. `log_rotation_age` Per ulteriori informazioni, consulta Impostazione della rotazione dei file di log.
rds.log_retention_period	4320	I registri PostgreSQL più vecchi del numero di minuti specificato vengono eliminati. Il valore di default di 4.320 minuti elimina i file di log dopo

Parametro	Predefinito	Descrizione
		3 giorni. Per ulteriori informazioni, consulta Impostazione del periodo di retention dei log.

Per identificare i problemi dell'applicazione, puoi cercare fallimenti di query, errori di accesso, deadlock ed errori irreversibili del server nel registro. Ad esempio, supponi di convertire un'applicazione legacy da Oracle ad Amazon RDS PostgreSQL, ma non tutte le query sono state convertite correttamente. Queste query formattate in modo errato generano messaggi di errore nei registri che puoi utilizzare per identificare i problemi. Per ulteriori informazioni sulla registrazione delle query, consulta Attivazione della registrazione delle query per l'istanza database RDS per PostgreSQL.

Negli argomenti seguenti sono disponibili informazioni su come impostare vari parametri che controllano i dettagli di base dei log PostgreSQL.

Argomenti

- Impostazione del periodo di retention dei log
- Impostazione della rotazione dei file di log
- Impostazione della destinazione del registro (stderr, csvlog)
- Informazioni sul parametro log_line_prefix

Impostazione del periodo di retention dei log

Il parametro `rds.log_retention_period` specifica per quanto tempo l'istanza database RDS per PostgreSQL conserva i file di log. L'impostazione predefinita è 3 giorni (4.320 minuti), ma è possibile impostare qualsiasi valore compreso tra 1 giorno (1.440 minuti) e 7 giorni (10.080 minuti). Assicurati che l'istanza database RDS per PostgreSQL abbia spazio di archiviazione sufficiente per contenere i file di log per il periodo di tempo specificato.

l'istanza database RDS per PostgreSQL. Per ulteriori informazioni, consulta Pubblicazione dei log PostgreSQL su Amazon Logs CloudWatch . Logs. CloudWatch

Impostazione della rotazione dei file di log

Per impostazione predefinita, nuovi file di log vengono creati da Amazon RDS ogni ora. La tempistica è controllata dal parametro `log_rotation_age`. Questo parametro ha un valore predefinito di

60 (minuti), ma è possibile impostarlo su qualsiasi valore tra 1 minuto e 24 ore (1.440 minuti). Al momento della rotazione, viene creato un nuovo file di log distinto. Il file è denominato in base al modello specificato dal parametro `log_filename`.

I file di log possono anche essere ruotati in base alle loro dimensioni, come specificato dal parametro `log_rotation_size`. Questo parametro specifica che il log deve essere ruotato quando raggiunge la dimensione specificata (in kilobyte). Per un'istanza database RDS for PostgreSQL, `log_rotation_size` non è impostato, cioè non è specificato alcun valore. Tuttavia, puoi impostare il parametro su un valore compreso tra 0 e 2.097.151 kB (kilobyte).

I nomi dei file di registro si basano sul modello di nome di file specificato nel parametro `log_filename`. Le impostazioni disponibili per questo parametro sono le seguenti:

- `postgresql.log.%Y-%m-%d` : formato predefinito per il nome del file di registro. Include l'anno, il mese e la data nel nome del file di log.
- `postgresql.log.%Y-%m-%d-%H`: include l'ora nel formato del nome del file di registro.

Per ulteriori informazioni, consulta `log_rotation_age` e `log_rotation_size` nella documentazione di PostgreSQL.

Impostazione della destinazione del registro (**`stderr`**, **`csvlog`**)

Per impostazione predefinita, PostgreSQL genera i log in formato errore standard (stderr). Questo formato è l'impostazione predefinita per il parametro `log_destination`. Ogni messaggio ha un prefisso che utilizza il modello specificato nel parametro `log_line_prefix`. Per ulteriori informazioni, consulta Informazioni sul parametro log_line_prefix.

RDS per PostgreSQL può anche generare log in formato `csvlog`. Il formato `csvlog` è utile per analizzare i dati dei registri in formato CSV. Ad esempio, supponi di utilizzare l'estensione `log_fdw` per lavorare con i log come tabelle esterne. La tabella esterna creata sui file di log di `stderr` contiene una singola colonna con i dati degli eventi di log. Aggiungendo `csvlog` al parametro `log_destination`, ottieni il file di log in formato CSV con le demarcazioni per le diverse colonne della tabella esterna. In tal modo puoi ordinare e analizzare i log più facilmente. Per informazioni su come usare `log_fdw` con `csvlog`, consulta Utilizzo dell'estensione log_fdw per accedere al registro di database utilizzando SQL.

Se specifichi `csvlog` per questo parametro, tieni presente che vengono generati entrambi i file `stderr` e `csvlog`. Ti consigliamo di monitorare lo spazio di archiviazione consumato dai registri tenendo conto di `rds.log_retention_period` e delle altre impostazioni che

influiscono sull'archiviazione e sulla rotazione dei registri. Utilizzando `stderr` e `csvlog` lo spazio di archiviazione consumato dai registri aumenta più del doppio.

Se aggiungi `csvlog` a `log_destination` e vuoi ripristinare solo `stderr`, devi reimpostare il parametro. Per farlo, nella console Amazon RDS apri il gruppo di parametri del·clusterdatabase personalizzato per la tua istanza. Scegli il parametro `log_destination`, seleziona Edit parameter (Modifica parametro), quindi Reset (Reimposta).

Per ulteriori informazioni sulla configurazione dei registri, consulta Utilizzo dei log Amazon RDS e Aurora PostgreSQL: Parte 1.

Informazioni sul parametro log_line_prefix

Il formato di log `stderr` applica il prefisso a ogni messaggio di log con i dettagli specificati dal parametro `log_line_prefix`, come indicato di seguito.

```
%t:%r:%u@%d:[%p]:t
```

Non puoi modificare questa impostazione. Ogni voce del log inviata a `stderr` include le seguenti informazioni.

- `%t` - Ora della voce di log
- `%r` - Indirizzo dell'host remoto
- `%u@%d` - Nome utente @ nome del database
- `[%p]` - ID del processo, se disponibile

Attivazione della registrazione delle query per l'istanza database RDS per PostgreSQL

È possibile raccogliere informazioni più approfondite sulle attività dei database, tra cui query, query in attesa di blocchi, checkpoint e molti altri dettagli impostando alcuni parametri elencati nella tabella seguente. Questo argomento illustra la registrazione delle query.

Parametro	Predefinito	Descrizione
log_connections	–	Registra ogni connessione riuscita.
log_disconnections	–	Registra il momento in cui termina ciascuna sessione e la relativa durata.

Parametro	Predefinito	Descrizione
log_checkpoints	1	Registra ogni checkpoint.
log_lock_waits	–	Registra lunghe attese di lock. Per impostazi one predefinita, questo parametro non è impostato.
log_min_duration_s ample	–	Imposta il tempo (ms) minimo di esecuzione oltre il quale viene registrato un campione di istruzioni. La dimensione del campione viene impostata utilizzando il parametro `log_state ment_sample_rate` .
log_min_duration_s tatement	–	Viene registrata qualsiasi istruzione SQL che viene eseguita per il periodo specificato o per più tempo. Per impostazione predefinita, questo parametro non è impostato. L'attivazione di questo parametro può aiutarti a trovare query non ottimizzate.
log_statement	–	Imposta il tipo di istruzioni registrate. Per impostazione predefinita, questo parametro non è impostato, ma puoi modificarlo in `all`, `ddl` o `mod` per specificare i tipi di istruzioni SQL che vuoi registrare. Se specifichi un valore diverso da `none` per questo parametro, dovrai adottare ulteriori misure per evitare l'esposizione delle password nei file di log. Per ulteriori informazi oni, consulta Riduzione del rischio di esposizio ne delle password quando si utilizza la registraz ione delle query.
log_statement_samp le_rate	–	La percentuale di istruzioni che superano il tempo specificato in `log_min_duration_s ample` da registrare, espressa come valore in virgola mobile compreso tra 0,0 e 1,0.

Parametro	Predefinito	Descrizione
log_statement_stats	–	Scrive le statistiche cumulative sulla prestazione nel registro del server.

Utilizzo della registrazione per trovare query lente

È possibile registrare istruzioni e query SQL per trovare le query con prestazioni lente. Puoi attivare questa funzionalità modificando le impostazioni nei parametri `log_statement` e `log_min_duration` come descritto in questa sezione. Prima di attivare la registrazione delle query per l'istanza database RDS per PostgreSQL, è necessario essere consapevoli della possibile esposizione delle password nei registri e di come mitigare i rischi. Per ulteriori informazioni, consulta Riduzione del rischio di esposizione delle password quando si utilizza la registrazione delle query.

Di seguito sono disponibili informazioni di riferimento sui parametri `log_statement` e `log_min_duration`.

log_statement

Questo parametro specifica il tipo di istruzioni SQL che devono essere inviate al registro. Il valore predefinito è `none`. Se modifichi questo parametro in `all`, `ddl` o `mod`, esegui le azioni consigliate per ridurre il rischio di esporre le password nei log. Per ulteriori informazioni, consulta Riduzione del rischio di esposizione delle password quando si utilizza la registrazione delle query.

tutto

Registra tutte le istruzioni. Questa impostazione è consigliata per il debug.

ddl

Registra tutte le istruzioni DDL (Data Definition Language), come CREATE, ALTER, DROP e così via.

mod

Registra tutte le istruzioni DDL e DML (Data Manipulation Language), come INSERT, UPDATE e DELETE, che modificano i dati.

nessuno

Nessuna istruzione SQL viene registrata. Consigliamo questa impostazione per evitare il rischio di esporre le password nei registri.

log_min_duration_statement

Viene registrata qualsiasi istruzione SQL che viene eseguita per il periodo specificato o per più tempo. Per impostazione predefinita, questo parametro non è impostato. L'attivazione di questo parametro può aiutarti a trovare query non ottimizzate.

–1–2147483647

Il numero di millisecondi (ms) di runtime durante il quale un'istruzione viene registrata.

Per configurare la registrazione delle query

Questi passaggi presuppongono che l'istanza database RDS per PostgreSQL utilizzi un gruppo di parametri database personalizzato.

1. Imposta il parametro `log_statement` su `all`. L'esempio seguente mostra le informazioni scritte nel file `postgresql.log` con questa impostazione del parametro.

```
2022-10-05 22:05:52 UTC:52.95.4.1(11335):postgres@labdb:[3639]:LOG: statement:
 SELECT feedback, s.sentiment,s.confidence
FROM support,aws_comprehend.detect_sentiment(feedback, 'en') s
ORDER BY s.confidence DESC;
2022-10-05 22:05:52 UTC:52.95.4.1(11335):postgres@labdb:[3639]:LOG: QUERY
 STATISTICS
2022-10-05 22:05:52 UTC:52.95.4.1(11335):postgres@labdb:[3639]:DETAIL: ! system
 usage stats:
! 0.017355 s user, 0.000000 s system, 0.168593 s elapsed
! [0.025146 s user, 0.000000 s system total]
! 36644 kB max resident size
! 0/8 [0/8] filesystem blocks in/out
! 0/733 [0/1364] page faults/reclaims, 0 [0] swaps
! 0 [0] signals rcvd, 0/0 [0/0] messages rcvd/sent
! 19/0 [27/0] voluntary/involuntary context switches
2022-10-05 22:05:52 UTC:52.95.4.1(11335):postgres@labdb:[3639]:STATEMENT: SELECT
 feedback, s.sentiment,s.confidence
FROM support,aws_comprehend.detect_sentiment(feedback, 'en') s
ORDER BY s.confidence DESC;
2022-10-05 22:05:56 UTC:52.95.4.1(11335):postgres@labdb:[3639]:ERROR: syntax error
 at or near "ORDER" at character 1
2022-10-05 22:05:56 UTC:52.95.4.1(11335):postgres@labdb:[3639]:STATEMENT: ORDER BY
 s.confidence DESC;
---------------------- END OF LOG ----------------------
```

2. Impostare il parametro `log_min_duration_statement`. L'esempio seguente mostra le informazioni scritte nel file `postgresql.log` quando il parametro è impostato su 1.

   Le query che superano la durata specificata nel parametro `log_min_duration_statement` vengono registrate. Di seguito viene riportato un esempio. Puoi visualizzare il file di log per l'istanza database RDS per PostgreSQL nella console Amazon RDS.

```
2022-10-05 19:05:19 UTC:52.95.4.1(6461):postgres@labdb:[6144]:LOG: statement: DROP
 table comments;
2022-10-05 19:05:19 UTC:52.95.4.1(6461):postgres@labdb:[6144]:LOG: duration:
 167.754 ms
2022-10-05 19:08:07 UTC::@:[355]:LOG: checkpoint starting: time
2022-10-05 19:08:08 UTC::@:[355]:LOG: checkpoint complete: wrote 11 buffers
 (0.0%); 0 WAL file(s) added, 0 removed, 0 recycled; write=1.013 s, sync=0.006 s,
 total=1.033 s; sync files=8, longest=0.004 s, average=0.001 s; distance=131028 kB,
 estimate=131028 kB
---------------------- END OF LOG ----------------------
```

Riduzione del rischio di esposizione delle password quando si utilizza la registrazione delle query

Ti consigliamo di mantenere `log_statement` impostato su `none` per evitare di esporre le password. Se imposti `log_statement` su `all`, `ddl` o `mod`, ti consigliamo di eseguire una o più delle seguenti operazioni.

- Per il client, applica la crittografia delle informazioni sensibili. Per ulteriori informazioni, consulta Encryption Options (Opzioni di crittografia) nella documentazione di PostgreSQL. Usa le opzioni `ENCRYPTED` (e `UNENCRYPTED`) delle istruzioni `CREATE` e `ALTER`. Per ulteriori informazioni, consulta CREATE USER nella documentazione di PostgreSQL.
- Per l'istanza database RDS per PostgreSQL, configura e usa l'estensione di audit PostgreSQL (pgAudit). Questa estensione oscura le informazioni sensibili nelle istruzioni CREATE e ALTER inviate al registro. Per ulteriori informazioni, consulta Utilizzo di pgAudit per registrare l'attività del database.
- Limita l'accesso ai log. CloudWatch
- Utilizza meccanismi di autenticazione più efficaci come IAM.

Pubblicazione dei log PostgreSQL su Amazon Logs CloudWatch

Per archiviare i record di log PostgreSQL in uno storage altamente durevole, puoi utilizzare Amazon Logs. CloudWatch Con CloudWatch Logs, puoi anche eseguire analisi in tempo reale dei dati di log e utilizzarli CloudWatch per visualizzare metriche e creare allarmi. Ad esempio, se imposti `log_statement` su `ddl`, puoi impostare un avviso per notificare ogni volta che viene eseguita un'istruzione DDL. Puoi scegliere di caricare i log di PostgreSQL in Logs durante il processo di creazione dell'istanza DB RDS CloudWatch per PostgreSQL. Se hai scelto di non caricare i registri, puoi successivamente modificare l'istanza per iniziare a caricare i registri da quel momento in poi. In altre parole, i log esistenti non vengono caricati. Solo i nuovi log vengono caricati quando vengono creati sull'istanza database RDS per PostgreSQL modificata.

Tutte le versioni di RDS per PostgreSQL attualmente disponibili supportano la pubblicazione di file di registro in Logs. CloudWatch Per informazioni dettagliate, consulta Amazon RDS for PostgreSQL updates (Aggiornamenti di Amazon RDS per PostgreSQL) in Amazon RDS for PostgreSQL Release Notes (Note di rilascio di Amazon RDS per PostgreSQL).

Per utilizzare CloudWatch Logs, configura l'istanza DB RDS for PostgreSQL per pubblicare i dati di log in un gruppo di log.

È possibile pubblicare i seguenti tipi di log in CloudWatch Logs for RDS for PostgreSQL:

- Log di PostgreSQL
- Registro di aggiornamento

Dopo aver completato la configurazione, Amazon RDS pubblica gli eventi di log per registrare i flussi all'interno di un CloudWatch gruppo di log. Ad esempio, i dati di log di PostgreSQL sono archiviati in un gruppo di log `/aws/rds/instance/my_instance/postgresql`. Per visualizzare i log, apri la console all' CloudWatch indirizzo https://console.aws.amazon.com/cloudwatch/.

Console

Per pubblicare i log di PostgreSQL su Logs utilizzando la console CloudWatch

1. Apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel pannello di navigazione, scegliere Databases (Database).
3. Scegliere l'istanza database che si desidera modificare e selezionare Modify (Modifica).

4. Nella sezione Esportazioni dei log, scegli i log che desideri iniziare a pubblicare su Logs. CloudWatch

   La sezione Esportazioni dei log è disponibile solo per le versioni di PostgreSQL che supportano la pubblicazione nei registri. CloudWatch

5. Scegliere Continue (Continua) e quindi Modify DB Instance (Modifica istanza database) nella pagina di riepilogo.

AWS CLI

È possibile pubblicare i log di PostgreSQL con. AWS CLI Puoi chiamare il comando `modify-db-instance` con i parametri seguenti.

- `--db-instance-identifier`
- `--cloudwatch-logs-export-configuration`

> **Note**
>
> Viene sempre applicata all'istanza database una modifica all'opzione `--cloudwatch-logs-export-configuration` immediatamente. Pertanto, le opzioni `--apply-immediately` e `--no-apply-immediately` non hanno alcun effetto.

Puoi anche pubblicare i log PostgreSQL chiamando i seguenti comandi CLI:

- `create-db-instance`
- `restore-db-instance-from-db-snapshot`
- `restore-db-instance-to-point-in-time`

Esegui uno di questi comandi CLI con le opzioni seguenti:

- `--db-instance-identifier`
- `--enable-cloudwatch-logs-exports`
- `--db-instance-class`
- `--engine`

Altre opzioni potrebbero essere richieste a seconda del comando CLI eseguito.

Example Modifica un'istanza per pubblicare i log in Logs CloudWatch

L'esempio seguente modifica un'istanza DB PostgreSQL esistente per pubblicare file di registro in Logs. CloudWatch Il valore `--cloudwatch-logs-export-configuration` è un oggetto JSON. La chiave per questo oggetto è `EnableLogTypes` e il suo valore è una matrice di stringhe con qualsiasi combinazione di `postgresql` e `upgrade`.

Per, o: Linux macOS Unix

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --cloudwatch-logs-export-configuration '{"EnableLogTypes":["postgresql",
 "upgrade"]}'
```

Per Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --cloudwatch-logs-export-configuration '{"EnableLogTypes":
["postgresql","upgrade"]}'
```

Example Crea un'istanza per pubblicare i log in Logs CloudWatch

L'esempio seguente crea un'istanza DB PostgreSQL e pubblica i file di registro in Logs. CloudWatch Il valore `--enable-cloudwatch-logs-exports` è una matrice di stringhe JSON. Le stringhe possono essere una qualsiasi combinazione di `postgresql` e `upgrade`.

Per, o: Linux macOS Unix

```
aws rds create-db-instance \
    --db-instance-identifier mydbinstance \
    --enable-cloudwatch-logs-exports '["postgresql","upgrade"]' \
    --db-instance-class db.m4.large \
    --engine postgres
```

Per Windows:

```
aws rds create-db-instance ^
    --db-instance-identifier mydbinstance ^
    --enable-cloudwatch-logs-exports '["postgresql","upgrade"]' ^
    --db-instance-class db.m4.large ^
    --engine postgres
```

API RDS

È possibile pubblicare i log PostgreSQL con RDS API. Puoi chiamare l'operazione ModifyDBInstance con i parametri seguenti:

- `DBInstanceIdentifier`
- `CloudwatchLogsExportConfiguration`

> **Note**
>
> Viene sempre applicata all'istanza database una modifica al parametro `CloudwatchLogsExportConfiguration` immediatamente. Pertanto, il parametro `ApplyImmediately` non ha alcun effetto.

È possibile anche pubblicare i log PostgreSQL eseguendo una delle seguenti azioni API RDS:

- CreateDBInstance
- RestoreDBInstanceFromDBSnapshot
- RestoreDBInstanceToPointInTime

Esegui una di queste azioni API RDS con i seguenti parametri:

- `DBInstanceIdentifier`
- `EnableCloudwatchLogsExports`
- `Engine`
- `DBInstanceClass`

Altri parametri potrebbero essere richiesti a seconda dell'operazione eseguita.

Monitoraggio delle chiamate API di Amazon RDS in AWS CloudTrail

AWS CloudTrail è un servizio AWS che ti aiuta a controllare il tuo account AWS. AWS CloudTrail è attivato sul tuo account AWS quando lo crei. Per ulteriori informazioni su CloudTrail, consulta la AWS CloudTrailGuida per l'utente di .

Argomenti

Integrazione di CloudTrail con Amazon RDS

Tutte le operazioni Amazon RDS sono registrate da CloudTrail. CloudTrail fornisce un record delle operazioni eseguite da un utente, un ruolo o un servizio AWS in Amazon RDS.

Eventi CloudTrail

CloudTrail acquisisce le chiamate API per Amazon RDS come eventi. Un evento rappresenta una singola richiesta da un'origine e include informazioni sull'operazione richiesta, data e ora dell'operazione, parametri della richiesta e così via. Gli eventi includono le chiamate della console Amazon RDS e le chiamate del codice alle operazioni API Amazon RDS.

L'attività Amazon RDS viene registrata in un evento CloudTrail nella cronologia eventi. Puoi utilizzare la console CloudTrail per visualizzare gli ultimi 90 giorni di attività API ed eventi registrati in una regione AWS. Per ulteriori informazioni, consulta Visualizzazione di eventi mediante la cronologia eventi di CloudTrail.

Trail CloudTrail

Per una registrazione continua degli eventi nell'account AWS che includa gli eventi per Amazon RDS, crea un percorso. Un percorso è una configurazione che consente la consegna di eventi a un bucket Simple Storage Service (Amazon S3) specificato. CloudTrail in genere consegna i file di log entro 15 minuti dall'attività dell'account.

> **Note**
>
> Se non configuri un trail, è comunque possibile visualizzare gli eventi più recenti nella console di CloudTrail in Event history (Cronologia eventi).

È possibile creare due tipi di trail per un account AWS: un trail che si applica a tutte le regioni o un trail che si applica a una regione. Per impostazione predefinita, quando si crea un trail nella console, il trail sarà valido in tutte le regioni .

Inoltre, è possibile configurare altri servizi AWS per analizzare con maggiore dettaglio e usare i dati evento raccolti nei registri CloudTrail. Per ulteriori informazioni, consulta:

- Panoramica della creazione di un percorso
- Servizi e integrazioni CloudTrail supportati
- Configurazione delle notifiche Amazon SNS per CloudTrail
- Ricezione di file di log CloudTrail da più regioni e Ricezione di file di log CloudTrail da più account

Voci del file di log Amazon RDS

I file di log di CloudTrail possono contenere una o più voci di log. I file di log di CloudTrail non sono una traccia stack ordinata delle chiamate pubbliche dell'API, quindi non vengono visualizzati in un ordine specifico.

L'esempio seguente mostra una voce di log di CloudTrail che illustra l'operazione `CreateDBInstance`.


```
{
    "eventVersion": "1.04",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "AKIAIOSFODNN7EXAMPLE",
        "arn": "arn:aws:iam::123456789012:user/johndoe",
        "accountId": "123456789012",
        "accessKeyId": "AKIAI44QH8DHBEXAMPLE",
        "userName": "johndoe"
    },
    "eventTime": "2018-07-30T22:14:06Z",
    "eventSource": "rds.amazonaws.com",
    "eventName": "CreateDBInstance",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "aws-cli/1.15.42 Python/3.6.1 Darwin/17.7.0 botocore/1.10.42",
    "requestParameters": {
        "enableCloudwatchLogsExports": [
```

```
            "audit",
            "error",
            "general",
            "slowquery"
        ],
        "dBInstanceIdentifier": "test-instance",
        "engine": "mysql",
        "masterUsername": "myawsuser",
        "allocatedStorage": 20,
        "dBInstanceClass": "db.m1.small",
        "masterUserPassword": "****"
    },
    "responseElements": {
        "dBInstanceArn": "arn:aws:rds:us-east-1:123456789012:db:test-instance",
        "storageEncrypted": false,
        "preferredBackupWindow": "10:27-10:57",
        "preferredMaintenanceWindow": "sat:05:47-sat:06:17",
        "backupRetentionPeriod": 1,
        "allocatedStorage": 20,
        "storageType": "standard",
        "engineVersion": "8.0.28",
        "dbInstancePort": 0,
        "optionGroupMemberships": [
            {
                "status": "in-sync",
                "optionGroupName": "default:mysql-8-0"
            }
        ],
        "dBParameterGroups": [
            {
                "dBParameterGroupName": "default.mysql8.0",
                "parameterApplyStatus": "in-sync"
            }
        ],
        "monitoringInterval": 0,
        "dBInstanceClass": "db.m1.small",
        "readReplicaDBInstanceIdentifiers": [],
        "dBSubnetGroup": {
            "dBSubnetGroupName": "default",
            "dBSubnetGroupDescription": "default",
            "subnets": [
                {
                    "subnetAvailabilityZone": {"name": "us-east-1b"},
                    "subnetIdentifier": "subnet-cbfff283",
```

```
                "subnetStatus": "Active"
            },
            {
                "subnetAvailabilityZone": {"name": "us-east-1e"},
                "subnetIdentifier": "subnet-d7c825e8",
                "subnetStatus": "Active"
            },
            {
                "subnetAvailabilityZone": {"name": "us-east-1f"},
                "subnetIdentifier": "subnet-6746046b",
                "subnetStatus": "Active"
            },
            {
                "subnetAvailabilityZone": {"name": "us-east-1c"},
                "subnetIdentifier": "subnet-bac383e0",
                "subnetStatus": "Active"
            },
            {
                "subnetAvailabilityZone": {"name": "us-east-1d"},
                "subnetIdentifier": "subnet-42599426",
                "subnetStatus": "Active"
            },
            {
                "subnetAvailabilityZone": {"name": "us-east-1a"},
                "subnetIdentifier": "subnet-da327bf6",
                "subnetStatus": "Active"
            }
        ],
        "vpcId": "vpc-136a4c6a",
        "subnetGroupStatus": "Complete"
    },
    "masterUsername": "myawsuser",
    "multiAZ": false,
    "autoMinorVersionUpgrade": true,
    "engine": "mysql",
    "cACertificateIdentifier": "rds-ca-2015",
    "dbiResourceId": "db-ETDZIIXHEWY5N7GXVC4SH7H5IA",
    "dBSecurityGroups": [],
    "pendingModifiedValues": {
        "masterUserPassword": "****",
        "pendingCloudwatchLogsExports": {
            "logTypesToEnable": [
                "audit",
                "error",
```

```
                    "general",
                    "slowquery"
                ]
            }
        },
        "dBInstanceStatus": "creating",
        "publiclyAccessible": true,
        "domainMemberships": [],
        "copyTagsToSnapshot": false,
        "dBInstanceIdentifier": "test-instance",
        "licenseModel": "general-public-license",
        "iAMDatabaseAuthenticationEnabled": false,
        "performanceInsightsEnabled": false,
        "vpcSecurityGroups": [
            {
                "status": "active",
                "vpcSecurityGroupId": "sg-f839b688"
            }
        ]
    },
    "requestID": "daf2e3f5-96a3-4df7-a026-863f96db793e",
    "eventID": "797163d3-5726-441d-80a7-6eeb7464acd4",
    "eventType": "AwsApiCall",
    "recipientAccountId": "123456789012"
}
```

Come illustrato nell'elemento `userIdentity` nell'esempio precedente, ogni voce di evento o di registro contiene informazioni su chi ha generato la richiesta. Le informazioni di identità consentono di determinare quanto segue:

- Se la richiesta è stata effettuata con le credenziali utente IAM o root.
- Se la richiesta è stata effettuata con le credenziali di sicurezza temporanee per un ruolo o un utente federato.
- Se la richiesta è stata effettuata da un altro servizio AWS.

Per ulteriori informazioni su `userIdentity`, consultare Elemento userIdentity CloudTrail. Per ulteriori informazioni su `CreateDBInstance` e altre operazioni di Amazon RDS, consulta la Documentazione di riferimento delle API di Amazon RDS.

Monitoraggio di Amazon RDS tramite i flussi di attività del database

Con Flussi di attività del database puoi monitorare pressoché in tempo reale i flussi di attività del database.

Argomenti

Panoramica dei flussi di attività di database

Come amministratore del database Amazon RDS, devi proteggere il database e soddisfare i requisiti normativi e di conformità. Una strategia consiste nell'integrare i flussi di attività del database con gli strumenti di monitoraggio. In questo modo, puoi monitorare e impostare gli allarmi per l'attività di verifica nel database.

Le minacce alla sicurezza sono sia esterne che interne. Per proteggersi dalle minacce interne, è possibile controllare l'accesso degli amministratori ai flussi di dati configurando la funzionalità flussi di attività del database. I DBA Amazon RDS non hanno accesso alla raccolta, alla trasmissione, all'archiviazione e all'elaborazione dei flussi.

Argomenti

- Classi di istanza database supportate per i flussi di attività di database

Come funzionano i flussi di attività del database

Amazon RDS inserisce le attività in un flusso di dati Amazon Kinesis pressoché in tempo reale. Il flusso Kinesis viene creato automaticamente. Da Kinesis, puoi configurare AWS servizi come Amazon Data Firehose e consumare lo stream e AWS Lambda archiviare i dati.

> **Important**
>
> L'utilizzo della funzionalità Flussi di attività del database in Amazon RDS è gratuito, ma Amazon Kinesis addebita i costi del flusso di dati. Per ulteriori informazioni, consulta Prezzi di Amazon Kinesis Data Streams.

Puoi configurare le applicazioni per la gestione della conformità affinché attingano dai flussi di attività del database. Tali applicazioni possono utilizzare il flusso per generare avvisi e attività di verifica per il database.

Amazon RDS supporta i flussi di attività del database nelle implementazioni multi-AZ. In questo caso, i flussi di attività del database controllano sia le istanze primarie che quelle di stand-by.

Verifica nel database Oracle Database e Microsoft SQL Server

La verifica è il monitoraggio e la registrazione delle azioni del database configurate. Amazon RDS non acquisisce le attività del database per impostazione predefinita. Puoi creare e gestire autonomamente le policy di verifica nel database.

Argomenti

- Verifica unificata in Oracle Database
- Verifica in Microsoft SQL Server
- Campi di verifica non nativi per Oracle Database e SQL Server
- Sovrascrittura di un gruppo parametri del database

Verifica unificata in Oracle Database

In un database Oracle, una policy di verifica unificata è un gruppo denominato di impostazioni di verifica che è possibile utilizzare per controllare un aspetto del comportamento dell'utente. Una policy

può essere semplice come controllare le attività di un singolo utente. È inoltre possibile creare policy di verifica complesse che utilizzano condizioni.

Un database Oracle scrive record di verifica, inclusi i record di verifica SYS, sul percorso di verifica unificato. Ad esempio, se si verifica un errore durante un'istruzione INSERT, la verifica standard indica il numero di errore e l'istruzione SQL eseguita. Il percorso di verifica si trova in una tabella di sola lettura nello schema AUDSYS. Per accedere a questi record, esegui una query nella vista del dizionario dei dati UNIFIED_AUDIT_TRAIL.

In genere, è possibile configurare i flussi di attività del database come segue:

1. Crea una policy di verifica Oracle Database utilizzando il comando CREATE AUDIT POLICY.

   Oracle Database genera record di verifica.
2. Attiva la policy di verifica utilizzando il comando AUDIT POLICY.
3. Configurazione dei flussi di attività di database

   Solo le attività che corrispondono alle policy di verifica Oracle Database vengono acquisite e inviate a Amazon Kinesis Data Stream. Quando i flussi di attività del database sono abilitati, un amministratore di database Oracle non può modificare le policy di verifica o rimuovere i log di verifica.

Per ulteriori informazioni sulle policy di verifica unificate, consulta Informazioni sulle attività di verifica con policy di verifica unificate e AUDIT nella Guida alla sicurezza di Oracle Database.

Verifica in Microsoft SQL Server

Il flusso di attività del database utilizza la funzionalità SQLAudit per verificare il database SQL Server.

L'istanza RDS per SQL Server contiene:

- Verifica del server: la verifica di SQL Server raccoglie una singola istanza di azioni a livello di server o database e un gruppo di azioni da monitorare. Le verifiche RDS_DAS_AUDIT e RDS_DAS_AUDIT_CHANGES a livello di server sono gestite da RDS.
- Specifica di verifica del server: la specifica di verifica del server registra gli eventi a livello di server. È possibile modificare la specifica RDS_DAS_SERVER_AUDIT_SPEC. Questa specifica è collegata alla verifica del server RDS_DAS_AUDIT. La specifica RDS_DAS_CHANGES_AUDIT_SPEC è gestita da RDS.

- Specifica di verifica del database: la specifica di verifica del database registra gli eventi a livello di database. È possibile creare una specifica di verifica del database `RDS_DAS_DB_<name>` e collegarla alla verifica del server `RDS_DAS_AUDIT`.

È possibile configurare i flussi di attività del database utilizzando la console o la CLI. In genere, è possibile configurare i flussi di attività del database come segue:

1. (Facoltativo) Crea una specifica di verifica del database con il comando `CREATE DATABASE AUDIT SPECIFICATION` e collegala alla verifica del server `RDS_DAS_AUDIT`.
2. (Facoltativo) Modifica la specifica di verifica del server con il comando `ALTER SERVER AUDIT SPECIFICATION` e definisci le policy.
3. Attiva le policy di verifica del database e del server. Per esempio:

   ```
   ALTER DATABASE AUDIT SPECIFICATION [<Your database specification>] WITH (STATE=ON)

   ALTER SERVER AUDIT SPECIFICATION [RDS_DAS_SERVER_AUDIT_SPEC] WITH (STATE=ON)
   ```
4. Configurazione dei flussi di attività di database

   Solo le attività che corrispondono alle policy di verifica del server e del database vengono acquisite e inviate al flusso di dati Amazon Kinesis. Quando i flussi di attività del database sono abilitati e le policy sono bloccate, un amministratore di database non può modificare le policy di verifica o rimuovere i log di verifica.

   > **Important**
   >
   > Se la specifica di verifica di un database specifico è abilitata e la policy è bloccata, il database non può essere eliminato.

Per ulteriori informazioni sulla verifica di SQL Server, consulta Componenti di verifica di SQL Server nella documentazione di Microsoft SQL Server.

Campi di verifica non nativi per Oracle Database e SQL Server

Quando si avvia un flusso di attività del database, ogni evento del database genera un evento del flusso di attività corrispondente. Ad esempio, un utente di database potrebbe eseguire le istruzioni `SELECT` e `INSERT`. Il database controlla questi eventi e li invia a un Amazon Kinesis Data Stream.

Gli eventi sono rappresentati nel flusso come oggetti JSON. Un oggetto JSON contiene un `DatabaseActivityMonitoringRecord`, che contiene una matrice `databaseActivityEventList`. I campi predefiniti nella matrice includono `class`, `clientApplication` e `command`.

Per impostazione predefinita, un flusso di attività non include campi di verifica nativi del motore. È possibile configurare Amazon RDS per Oracle e SQL Server in modo che includano questi campi aggiuntivi nell'oggetto JSON `engineNativeAuditFields`.

In Oracle Database, la maggior parte degli eventi nel percorso di verifica unificato esegue la mappatura ai campi nel flusso di attività dei dati RDS. Ad esempio, il campo `UNIFIED_AUDIT_TRAIL.SQL_TEXT` nelle mappe di verifica unificate al campo `commandText` in un flusso di attività di database. Tuttavia, i campi di verifica di Oracle Database come `OS_USERNAME` non mappano a campi predefiniti in un flusso di attività di database.

In SQL Server, la maggior parte dei campi dell'evento registrati da SQLAudit esegue la mappatura ai campi nel flusso di attività del database RDS. Ad esempio, il campo `code` in `sys.fn_get_audit_file` nella verifica viene mappato al campo `commandText` in un flusso di attività del database. Tuttavia, i campi di verifica del database SQL Server come `permission_bitmask` non vengono mappati ai campi predefiniti in un flusso di attività del database.

Per ulteriori informazioni su databaseActivityEvent List, consulta. databaseActivityEventElenca l'array JSON

Sovrascrittura di un gruppo parametri del database

In genere, è possibile attivare la verifica unificata in RDS per Oracle allegando un gruppo di parametri. Tuttavia, i flussi di attività del database richiedono una configurazione aggiuntiva. Per migliorare l'esperienza del cliente, Amazon RDS effettua le seguenti operazioni:

- Se attivi un flusso di attività, RDS per Oracle ignora i parametri di verifica nel gruppo di parametri.
- Se disattivi un flusso di attività, RDS per Oracle smette di ignorare i parametri di verifica.

Il flusso di attività del database per SQL Server è indipendente da qualsiasi parametro impostato nell'opzione di verifica di SQL.

Modalità asincrona per flussi di attività di database

I flussi di attività in Amazon RDS sono sempre asincroni. Quando una sessione di database genera un evento del flusso di attività, vengono ripristinate immediatamente le normali attività della sessione. In background, Amazon RDS rende l'evento di flusso di attività un record persistente.

Se si verifica un errore nell'attività in background, Amazon RDS genera un evento. Questo indica l'inizio e la fine di qualsiasi finestra temporale in cui i record dell'evento del flusso di attività potrebbero essere stati persi. La modalità asincrona favorisce le prestazioni del database rispetto alla precisione del flusso di attività.

Requisiti e limitazioni per flussi di attività del database

In RDS, i flussi di attività del database hanno i requisiti e le limitazioni riportati di seguito:

- I flussi di attività del database richiedono l'utilizzo di Amazon Kinesis.
- AWS Key Management Service (AWS KMS) è necessario per i flussi di attività del database perché sono sempre crittografati.
- L'applicazione di una crittografia aggiuntiva al flusso di dati di Amazon Kinesis è incompatibile con i flussi di attività del database, che sono già crittografati con la tua chiave. AWS KMS
- Puoi creare e gestire autonomamente le policy di verifica. A differenza di Amazon Aurora, RDS per Oracle non acquisisce le attività del database per impostazione predefinita.
- Puoi creare e gestire autonomamente le policy o le specifiche di verifica. A differenza di Amazon Aurora, Amazon RDS non acquisisce le attività del database per impostazione predefinita.
- In un'implementazione multi-AZ, avvia il flusso di attività del database solo sull'istanza database primaria. Il flusso di attività controlla automaticamente sia l'istanza DB principale che quelle in stand-by. Durante un failover non sono richiesti passaggi aggiuntivi.
- La ridenominazione di un'istanza database non crea un nuovo flusso Kinesis.
- I CDB non sono supportati per RDS per Oracle.
- Le repliche di lettura non sono supportate.

Disponibilità di regioni e versioni

Il supporto varia a seconda delle versioni specifiche di ciascun motore di database e a seconda delle Regioni AWS. Per ulteriori informazioni sulla disponibilità di versioni e regioni con flussi di attività del database, consulta Regioni e motori DB supportati per i flussi di attività del database in Amazon RDS.

Classi di istanza database supportate per i flussi di attività di database

In RDS per Oracle è possibile utilizzare i flussi di attività del database con le seguenti classi di istanza database:

- db.m4.*large
- db.m5.*large
- db.m5d.*large
- db.m6i.*large
- db.r4.*large
- db.r5.*large
- db.r5.*large.tpc*.mem*x
- db.r5b.*large
- db.r5b.*large.tpc*.mem*x
- db.r5d.*large
- db.r6i.*large
- db.x2idn.*large
- db.x2iedn.*large
- db.x2iezn.*large
- db.z1d.*large

In RDS per SQL Server è possibile utilizzare i flussi di attività del database con le seguenti classi di istanza database:

- db.m4.*large
- db.m5.*large
- db.m5d.*large
- db.m6i.*large

- db.r4.*large
- db.r5.*large
- db.r5b.*large
- db.r5d.*large
- db.r6i.*large
- db.x1e.*large
- db.z1d.*large

Per ulteriori informazioni sulle classi di istanza, consulta Classi di istanze database.

Configurazione della verifica unificata per Oracle Database

Quando si configura la verifica unificata per l'utilizzo con i flussi di attività del database, sono possibili le seguenti situazioni:

- La verifica unificata non è configurata per il database Oracle.

  In questo caso, crea nuove policy con il comando `CREATE AUDIT POLICY` e quindi attivale con il comando `AUDIT POLICY`. L'esempio seguente crea e attiva una policy per monitorare gli utenti con privilegi e ruoli specifici.

  ```
  CREATE AUDIT POLICY table_pol
  PRIVILEGES CREATE ANY TABLE, DROP ANY TABLE
  ROLES emp_admin, sales_admin;

  AUDIT POLICY table_pol;
  ```

  Per istruzioni complete, consulta Configurazione delle policy di verifica nella documentazione di Oracle Database.
- La verifica unificata è configurata per il database Oracle.

  Quando attivi un flusso di attività del database, RDS per Oracle cancella automaticamente i dati di audit esistenti. Inoltre revoca i privilegi del percorso di verifica. RDS per Oracle non può più eseguire le seguenti operazioni:

  - Rimozione dei record del percorso di verifica unificata.
  - Aggiunta, eliminazione o modifica delle policy di verifica unificata.

- Aggiornamento dell'ultimo timestamp archiviato.

> **Important**
>
> Consigliamo vivamente di eseguire il backup dei dati di verifica prima di attivare un flusso di attività di database.

Per una descrizione della vista `UNIFIED_AUDIT_TRAIL`, consulta UNIFIED_AUDIT_TRAIL. Se si dispone di un account con Oracle Support, consulta Come rimuovere il PERCORSO DI VERIFICA UNIFICATA.

Configurazione della policy di audit per Microsoft SQL Server

Un'istanza database SQL Server dispone dell'audit del server `RDS_DAS_AUDIT`, gestito da Amazon RDS. È possibile definire le policy per registrare gli eventi del server nelle specifiche di audit del server `RDS_DAS_SERVER_AUDIT_SPEC`. È possibile creare una specifica di audit del database, ad esempio `RDS_DAS_DB_<name>`, e definire le policy per registrare gli eventi del database. Per l'elenco dei gruppi di azioni di audit a livello di server e database, consulta Azioni e gruppi di azioni di audit di SQL Server nella documentazione di Microsoft SQL Server.

La policy del server predefinita monitora solo gli accessi non riusciti e le modifiche a qualsiasi specifica di audit del database o del server per i flussi di attività del database.

Le limitazioni relative all'audit e alle specifiche di audit sono le seguenti:

- Non è possibile modificare le specifiche di audit del server o del database quando il flusso di attività del database è bloccato.
- Non è possibile modificare la specifica di audit del server `RDS_DAS_AUDIT`.
- Non è possibile modificare la specifica di audit di SQL Server `RDS_DAS_CHANGES` o le relative specifiche di audit del server `RDS_DAS_CHANGES_AUDIT_SPEC`.
- Quando si crea una specifica di audit del database, è necessario utilizzare il formato `RDS_DAS_DB_<name>`, ad esempio `RDS_DAS_DB_databaseActions`.

> **Important**
>
> Per le classi di istanze più piccole, consigliamo di non eseguire l'audit di tutti i dati, ma solo di quelli necessari. In tal modo si riduce l'impatto sulle prestazioni dei flussi di attività del database su queste classi di istanze.

Il seguente codice di esempio modifica la specifica di audit del server `RDS_DAS_SERVER_AUDIT_SPEC` e verifica qualsiasi azione di disconnessione e di accesso completato:

```
ALTER SERVER AUDIT SPECIFICATION [RDS_DAS_SERVER_AUDIT_SPEC]
      WITH (STATE=OFF);
ALTER SERVER AUDIT SPECIFICATION [RDS_DAS_SERVER_AUDIT_SPEC]
      ADD (LOGOUT_GROUP),
      ADD (SUCCESSFUL_LOGIN_GROUP)
      WITH (STATE = ON );
```

Il codice di esempio seguente crea una specifica di audit del database `RDS_DAS_DB_database_spec` e la collega alla specifica di audit del server `RDS_DAS_AUDIT`:

```
USE testDB;
CREATE DATABASE AUDIT SPECIFICATION [RDS_DAS_DB_database_spec]
     FOR SERVER AUDIT [RDS_DAS_AUDIT]
     ADD ( INSERT, UPDATE, DELETE
          ON testTable BY testUser )
     WITH (STATE = ON);
```

Dopo aver configurato le specifiche di audit, assicurati che le specifiche `RDS_DAS_SERVER_AUDIT_SPEC` e `RDS_DAS_DB_<name>` siano impostate sullo stato `ON`. A questo punto i dati di audit possono essere inviati al flusso di attività del database.

Avvio di un flusso di attività di database

Quando avvii un flusso di attività per l'istanza database, ogni evento di attività del database configurato nella policy di audit, genera un evento di flusso di attività. Gli eventi di accesso vengono generati da comandi SQL quali `CONNECT` e `SELECT`. Gli eventi di modifica vengono generati da comandi SQL quali `CREATE` e `INSERT`.

> **Important**
>
> Attivazione di un flusso di attività per un'istanza database Oracle cancella i dati di verifica esistenti. Inoltre revoca i privilegi del percorso di verifica. Quando il flusso è abilitato, RDS per Oracle non può più eseguire le seguenti operazioni:
>
> - Rimozione dei record del percorso di verifica unificata.
> - Aggiunta, eliminazione o modifica delle policy di verifica unificata.
> - Aggiornamento dell'ultimo timestamp archiviato.

Console

Come avviare un flusso di attività di database

1. Apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel pannello di navigazione, scegliere Databases (Database).
3. Seleziona l'istanza database Amazon RDS per cui desideri abilitare un flusso di attività. In un'implementazione Multi-AZ, avvia il flusso solo sull'istanza database primaria. Il flusso di attività controlla automaticamente sia l'istanza DB principale che quelle in stand-by.
4. In Actions (Operazioni), scegliere Start activity stream (Avvia flusso di attività).

   Viene visualizzata la finestra Avvia flusso di attività di database: *nome*, dove *nome* è la tua istanza RDS.
5. Specificare le seguenti impostazioni:

   - In AWS KMS key, scegliere una chiave dall'elenco di AWS KMS keys.

     Amazon RDS utilizza la chiave KMS per crittografare la chiave che a sua volta esegue la crittografia dell'attività del database. Scegliere una chiave KMS diversa dalla chiave di default. Per ulteriori informazioni sulle chiavi di crittografia e AWS KMS, consulta Che cos'è AWS Key Management Service? nella Guida per gli sviluppatori di AWS Key Management Service.
   - Per Eventi di attività del database seleziona Abilita i campi di controllo nativi del motore per includere nel flusso i campi di controllo specifici del motore.
   - Scegliere Immediatamente.

Selezionando Subito, l'istanza RDS viene riavviata immediatamente. Se si sceglie Durante la finestra di manutenzione successiva, l'istanza RDS non si riavvia subito. In questo caso, il flusso di attività del database non viene avviato fino alla finestra di manutenzione successiva.

6. Scegli Start database activity stream (Avvia flusso di attività di database).

Lo stato del database mostra che il flusso di attività è in fase di avvio.

> **Note**
>
> Se ricevi l'errore `You can't start a database activity stream in this configuration`, controlla Classi di istanza database supportate per i flussi di attività di database per vedere se l'istanza RDS utilizza una classe di istanza supportata.

AWS CLI

Per avviare i flussi di attività del database per (un'istanza DB), configura di database utilizzando il start-activity-streamAWS CLIcomando.

- `--resource-arn` *`arn`*: specifica l'Amazon Resource Name (ARN) dell'istanza database.
- `--kms-key-id` *`key`*: specifica l'identificatore della chiave KMS per la crittografia dei messaggi nel flusso di attività del database. L'identificatore di chiave AWS KMS è l'ARN della chiave, l'ID chiave, l'ARN dell'alias o il nome alias per la AWS KMS key.
- `--engine-native-audit-fields-included`: include campi di controllo specifici del motore nel flusso di dati. Per escludere questi campi, specificare `--no-engine-native-audit-fields-included` (predefinito).

L'esempio seguente avvia un flusso di attività del database per un'istanza database in modalità asincrona.

Per LinuxmacOS, oUnix:

```
aws rds start-activity-stream \
    --mode async \
    --kms-key-id my-kms-key-arn \
    --resource-arn my-instance-arn \
    --engine-native-audit-fields-included \
```

```
    --apply-immediately
```

Per Windows:

```
aws rds start-activity-stream ^
    --mode async ^
    --kms-key-id my-kms-key-arn ^
    --resource-arn my-instance-arn ^
    --engine-native-audit-fields-included ^
    --apply-immediately
```

API RDS

Per avviare i flussi di attività del database per di database (un'istanza DB), configura utilizzando l'StartActivityStreamoperazione.

Richiamare l'operazione con i parametri seguenti:

- `Region`
- `KmsKeyId`
- `ResourceArn`
- `Mode`
- `EngineNativeAuditFieldsIncluded`

Modifica di un flusso di attività del database

Quando si avvia il flusso di attività, potrebbe essere necessario personalizzare la policy di audit di Amazon RDS. Per non perdere tempo e dati a causa dell'interruzione del flusso di attività, puoi modificare lo stato della policy di audit in una delle seguenti impostazioni:

Locked (default) (Bloccato (impostazione predefinita))

Le policy di audit nel database sono di sola lettura.

Unlocked (Sbloccato)

Le policy di audit nel database sono di lettura/scrittura.

I passaggi di base sono i seguenti:

1. Modifica lo stato della policy di audit in sbloccato.
2. Personalizza la policy di audit.
3. Modifica lo stato della policy di audit in bloccato.

Console

Per modificare lo stato della policy di audit del flusso di attività

1. Apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel pannello di navigazione, scegliere Databases (Database).
3. Per Actions (Operazioni), scegli Modify database activity stream (Modifica flusso di attività del database).

   Viene visualizzata la finestra Modify database activity stream: ***name*** (Modifica flusso di attività del database), dove *name* (nome) è l'istanza RDS.
4. Seleziona una delle seguenti opzioni:

   Locked (Bloccato)

   Quando si blocca la policy di audit, questa diventa di sola lettura. Non puoi modificare la policy di audit a meno che non sblocchi la policy o interrompi il flusso di attività.

   Unlocked (Sbloccato)

   Quando si sblocca la policy di audit, questa diventa di lettura/scrittura. Puoi modificare la policy di audit all'avvio del flusso di attività.
5. Scegli Modify DB activity stream (Modifica flusso di attività del database).

   Lo stato del database Amazon RDS mostra Configurazione del flusso di attività in corso.
6. (Facoltativo) Scegli il collegamento all'istanza database. Quindi seleziona la scheda Configurazione.

   Il campo Audit policy status (Stato della policy di audit) mostra uno dei seguenti valori:

   - Locked (Bloccato)
   - Unlocked (Sbloccato)
   - Locking policy (Policy di blocco)
   - Unlocking policy (Policy di sblocco)

AWS CLI

Per modificare lo stato del flusso di attività per l'istanza del database, utilizzare il modify-activity-streamAWS CLIcomando.

Opzione	Obbligatorio?	Description
`--resource-arn` *my-instance-ARN*	Sì	Il nome della risorsa Amazon (ARN) dell'istanza database RDS.
`--audit-policy-state`	No	Il nuovo stato della policy di audit per il flusso di attività del database sull'istanza: `locked` o `unlocked`.

Nell'esempio seguente la policy di audit viene sbloccata per il flusso di attività avviato su *my-instance-ARN*.

Per LinuxmacOS, oUnix:

```
aws rds modify-activity-stream \
    --resource-arn my-instance-ARN \
    --audit-policy-state unlocked
```

Per Windows:

```
aws rds modify-activity-stream ^
    --resource-arn my-instance-ARN ^
    --audit-policy-state unlocked
```

Nell'esempio seguente viene descritta l'istanza *my-instance*. L'output di esempio parziale mostra che la policy di audit è sbloccata.

```
aws rds describe-db-instances --db-instance-identifier my-instance

{
    "DBInstances": [
        {
            ...
            "Engine": "oracle-ee",
```

```
            ...
            "ActivityStreamStatus": "started",
            "ActivityStreamKmsKeyId": "ab12345e-1111-2bc3-12a3-ab1cd12345e",
            "ActivityStreamKinesisStreamName": "aws-rds-das-db-
AB1CDEFG23GHIJK4LMNOPQRST",
            "ActivityStreamMode": "async",
            "ActivityStreamEngineNativeAuditFieldsIncluded": true,
            "ActivityStreamPolicyStatus": "unlocked",
            ...
        }
    ]
}
```

API RDS

Per modificare lo stato delle politiche del flusso di attività del database, utilizzate l'ModifyActivityStreamoperazione.

Richiamare l'operazione con i parametri seguenti:

- `AuditPolicyState`
- `ResourceArn`

Recupero dello stato di un flusso di attività del database

Puoi recuperare lo stato di un flusso di attività per l'istanza database Amazon RDS tramite la console o la AWS CLI.

Console

Come recuperare lo stato di un flusso di attività del database

1. Apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel pannello di navigazione, scegliere Database e selezionare il link dell'istanza database.
3. Scegliere la scheda Configurazione e selezionare Flusso di attività di database per lo stato.

AWS CLI

È possibile ottenere la configurazione del flusso di attività per un'istanza database come risposta a una richiesta della CLI describe-db-instances.

Nell'esempio seguente viene illustrato *my-instance*.

```
aws rds --region my-region describe-db-instances --db-instance-identifier my-db
```

Il seguente esempio mostra una risposta in formato JSON: Sono visualizzati i seguenti campi:

- `ActivityStreamKinesisStreamName`
- `ActivityStreamKmsKeyId`
- `ActivityStreamStatus`
- `ActivityStreamMode`
- `ActivityStreamPolicyStatus`

```
{
    "DBInstances": [
        {
            ...
            "Engine": "oracle-ee",
            ...
            "ActivityStreamStatus": "starting",
            "ActivityStreamKmsKeyId": "ab12345e-1111-2bc3-12a3-ab1cd12345e",
            "ActivityStreamKinesisStreamName": "aws-rds-das-db-
AB1CDEFG23GHIJK4LMNOPQRST",
            "ActivityStreamMode": "async",
            "ActivityStreamEngineNativeAuditFieldsIncluded": true,
            "ActivityStreamPolicyStatus": locked",
            ...
        }
    ]
}
```

API RDS

È possibile ottenere la configurazione del flusso di attività per un database come risposta a un'operazione DescribeDBInstances.

Arresto di un flusso di attività di database

Puoi interrompere un flusso di attività utilizzando la console o AWS CLI.

Se elimini l'istanza database Amazon RDS, il flusso di attività viene arrestato e il flusso Amazon Kinesis sottostante viene eliminato automaticamente.

Console

Per disattivare un flusso di attività

1. Apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel pannello di navigazione, scegliere Databases (Database).
3. Scegliere un database per il quale si desidera interrompere il flusso di attività di database.
4. In Actions (Operazioni), scegliere Stop activity stream (Interrompi flusso di attività). Viene visualizzata la finestra Database Activity Stream (Flusso di attività di database).

    a. Scegliere Immediatamente.

        Selezionando Subito, l'istanza RDS viene riavviata immediatamente. Se si sceglie Durante la finestra di manutenzione successiva, l'istanza RDS non si riavvia subito. In questo caso, il flusso di attività del database non viene arrestato fino alla finestra di manutenzione successiva.

    b. Scegli Continue (Continua).

AWS CLI

Per interrompere i flussi di attività del database per database, configura l'istanza DB utilizzando il AWS CLI comando stop-activity-stream. Identifica la regione AWS per l'istanza database mediante il parametro `--region`. Il parametro `--apply-immediately` è facoltativo.

Per LinuxmacOS, oUnix:

```
aws rds --region MY_REGION \
    stop-activity-stream \
    --resource-arn MY_DB_ARN \
    --apply-immediately
```

Per Windows:

```
aws rds --region MY_REGION ^
    stop-activity-stream ^
    --resource-arn MY_DB_ARN ^
```

```
    --apply-immediately
```

API RDS

Per interrompere i flussi di attività del database per di database (il database), configura l'istanza DB del utilizzando l'StopActivityStreamoperazione. Identifica la regione AWS per l'istanza database mediante il parametro `Region`. Il parametro `ApplyImmediately` è facoltativo.

Monitoraggio di flussi di attività di database

I flussi di attività di database monitorano e segnalano le attività. Il flusso di attività viene raccolto e trasmesso a Amazon Kinesis. Da Kinesis, è possibile monitorare il flusso di attività oppure altri servizi e applicazioni possono utilizzare il flusso di attività per ulteriori analisi. Puoi trovare il nome dello stream Kinesis sottostante utilizzando il AWS CLI comando o l'operazione API RDS.

Amazon RDS gestisce il flusso Kinesis per tuo conto come segue:

- Amazon RDS crea automaticamente il flusso Kinesis con un periodo di conservazione di 24 ore.
- Amazon RDS dimensiona il flusso Kinesis, se necessario.
- Se si interrompe il flusso di attività del database o si elimina l'istanza database, Amazon RDS elimina il flusso Kinesis.

Le seguenti categorie di attività vengono monitorate e inserite nel log di controllo del flusso di attività:

- Comandi SQL: tutti i comandi SQL sono controllati e anche le istruzioni preparate, le funzioni integrate e le funzioni in PL/SQL Le chiamate alle procedure archiviate vengono controllate. Vengono inoltre controllate tutte le istruzioni SQL rilasciate all'interno di procedure o funzioni archiviate.
- Altre informazioni di database – L'attività monitorata include l'istruzione SQL completa, il conteggio righe delle righe interessate da comandi DML, gli oggetti ai quali si accede e il nome del database univoco. I flussi di attività del database monitorano anche le variabili di bind e i parametri della stored procedure.

> **Important**
>
> Il testo SQL completo di ogni istruzione è visibile nel registro di controllo del flusso di attività, inclusi eventuali dati sensibili. Tuttavia, le password degli utenti del database vengono omesse se Oracle può stabilirle dal contesto, come nell'istruzione SQL seguente.

```
ALTER ROLE role-name WITH password
```

- Informazioni di connessione – L'attività monitorata include informazioni di sessione e di rete, l'ID di processo del server e i codici di uscita.

Se un flusso di attività restituisce un errore durante il monitoraggio dell'istanza database, riceverai una notifica mediante eventi RDS.

Argomenti

- Accesso a un flusso di attività da Kinesis
- Contenuti ed esempi del registro di controllo
- databaseActivityEventElenca l'array JSON
- Elaborazione di un flusso di attività del database utilizzando l'SDK AWS

Accesso a un flusso di attività da Kinesis

Quando abiliti un flusso di attività per un database, viene creato automaticamente un flusso Kinesis. Da Kinesis, puoi monitorare l'attività del database in tempo reale. Per analizzare ulteriormente l'attività del database, puoi connettere il flusso Kinesis ad applicazioni consumer. Puoi anche connettere lo stream ad applicazioni di gestione della conformità come Security Guardium di di Imperva. SecureSphere

Puoi accedere al tuo flusso Kinesis dalla console RDS o dalla console Kinesis.

Come accedere a un flusso di attività da Kinesis utilizzando la console RDS

1. Apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel pannello di navigazione, scegliere Databases (Database).
3. Seleziona l'istanza database Amazon RDS in cui hai avviato un flusso di attività.
4. Scegliere Configuration (Configurazione).
5. In Database activity stream (Flusso di attività del database), scegli il collegamento sotto Kinesis stream (Flusso Kinesis).
6. Nella console Kinesis, scegli Monitoring (Monitoraggio) per iniziare l'osservazione dell'attività del database.

Per accedere a un flusso di attività da Kinesis utilizzando la console Kinesis

1. Aprire la console Kinesis all'indirizzo https://console.aws.amazon.com/kinesis.
2. Scegliere il flusso di attività dall'elenco di flussi Kinesis.

   Il nome di un flusso di attività include il prefisso `aws-rds-das-db-` seguito dall'ID risorsa del database. Di seguito è riportato un esempio.

   ```
   aws-rds-das-db-NHVOV4PCLWHGF52NP
   ```

   Per utilizzare la console Amazon RDS per trovare l'ID risorsa per il database, scegli l'istanza database dall'elenco di database, quindi seleziona la scheda Configuration (Configurazione).

   Per utilizzare AWS CLI per trovare il nome completo dello stream Kinesis per un flusso di attività, usa una richiesta CLI e annota il valore di `ActivityStreamKinesisStreamName` nella risposta.
3. Scegliere Monitoring (Monitoraggio) per iniziare l'osservazione dell'attività di database.

Per ulteriori informazioni sull'utilizzo di Amazon Kinesis, consulta Che cos'è Amazon Kinesis Data Streams?.

Contenuti ed esempi del registro di controllo

Gli eventi monitorati sono rappresentati nel flusso di attività del database come stringhe JSON. La struttura è costituita da un oggetto JSON contenente un `DatabaseActivityMonitoringRecord`, che a sua volta contiene un array `databaseActivityEventList` di eventi attività.

Argomenti

- Esempi di log di verifica per un flusso di attività
- DatabaseActivityMonitoringRecords Oggetto JSON
- databaseActivityEvents Oggetto JSON

Esempi di log di verifica per un flusso di attività

Di seguito sono riportati registri di controllo JSON decrittografati di esempio di record di eventi attività.

Example Record di evento di attività di un'istruzione CONNECT SQL

Il seguente record di evento di attività mostra un accesso con l'utilizzo di un'istruzione SQL `CONNECT` (`command`) mediante un client JDBC Thin (`clientApplication`) per il database Oracle.


```
{
    "class": "Standard",
    "clientApplication": "JDBC Thin Client",
    "command": "LOGON",
    "commandText": null,
    "dbid": "0123456789",
    "databaseName": "ORCL",
    "dbProtocol": "oracle",
    "dbUserName": "TEST",
    "endTime": null,
    "errorMessage": null,
    "exitCode": 0,
    "logTime": "2021-01-15 00:15:36.233787",
    "netProtocol": "tcp",
    "objectName": null,
    "objectType": null,
    "paramList": [],
    "pid": 17904,
    "remoteHost": "123.456.789.012",
    "remotePort": "25440",
    "rowCount": null,
    "serverHost": "987.654.321.098",
    "serverType": "oracle",
    "serverVersion": "19.0.0.0.ru-2020-01.rur-2020-01.r1.EE.3",
    "serviceName": "oracle-ee",
    "sessionId": 987654321,
    "startTime": null,
    "statementId": 1,
    "substatementId": null,
    "transactionId": "0000000000000000",
    "engineNativeAuditFields": {
        "UNIFIED_AUDIT_POLICIES": "TEST_POL_EVERYTHING",
        "FGA_POLICY_NAME": null,
        "DV_OBJECT_STATUS": null,
        "SYSTEM_PRIVILEGE_USED": "CREATE SESSION",
        "OLS_LABEL_COMPONENT_TYPE": null,
        "XS_SESSIONID": null,
        "ADDITIONAL_INFO": null,
        "INSTANCE_ID": 1,
```

```
        "DBID": 123456789
        "DV_COMMENT": null,
        "RMAN_SESSION_STAMP": null,
        "NEW_NAME": null,
        "DV_ACTION_NAME": null,
        "OLS_PROGRAM_UNIT_NAME": null,
        "OLS_STRING_LABEL": null,
        "RMAN_SESSION_RECID": null,
        "OBJECT_PRIVILEGES": null,
        "OLS_OLD_VALUE": null,
        "XS_TARGET_PRINCIPAL_NAME": null,
        "XS_NS_ATTRIBUTE": null,
        "XS_NS_NAME": null,
        "DBLINK_INFO": null,
        "AUTHENTICATION_TYPE": "(TYPE\u003d(DATABASE));(CLIENT ADDRESS\u003d((ADDRESS
\u003d(PROTOCOL\u003dtcp)(HOST\u003d205.251.233.183)(PORT\u003d25440))));",
        "OBJECT_EDITION": null,
        "OLS_PRIVILEGES_GRANTED": null,
        "EXCLUDED_USER": null,
        "DV_ACTION_OBJECT_NAME": null,
        "OLS_LABEL_COMPONENT_NAME": null,
        "EXCLUDED_SCHEMA": null,
        "DP_TEXT_PARAMETERS1": null,
        "XS_USER_NAME": null,
        "XS_ENABLED_ROLE": null,
        "XS_NS_ATTRIBUTE_NEW_VAL": null,
        "DIRECT_PATH_NUM_COLUMNS_LOADED": null,
        "AUDIT_OPTION": null,
        "DV_EXTENDED_ACTION_CODE": null,
        "XS_PACKAGE_NAME": null,
        "OLS_NEW_VALUE": null,
        "DV_RETURN_CODE": null,
        "XS_CALLBACK_EVENT_TYPE": null,
        "USERHOST": "a1b2c3d4e5f6.amazon.com",
        "GLOBAL_USERID": null,
        "CLIENT_IDENTIFIER": null,
        "RMAN_OPERATION": null,
        "TERMINAL": "unknown",
        "OS_USERNAME": "sumepate",
        "OLS_MAX_READ_LABEL": null,
        "XS_PROXY_USER_NAME": null,
        "XS_DATASEC_POLICY_NAME": null,
        "DV_FACTOR_CONTEXT": null,
        "OLS_MAX_WRITE_LABEL": null,
```

```
        "OLS_PARENT_GROUP_NAME": null,
        "EXCLUDED_OBJECT": null,
        "DV_RULE_SET_NAME": null,
        "EXTERNAL_USERID": null,
        "EXECUTION_ID": null,
        "ROLE": null,
        "PROXY_SESSIONID": 0,
        "DP_BOOLEAN_PARAMETERS1": null,
        "OLS_POLICY_NAME": null,
        "OLS_GRANTEE": null,
        "OLS_MIN_WRITE_LABEL": null,
        "APPLICATION_CONTEXTS": null,
        "XS_SCHEMA_NAME": null,
        "DV_GRANTEE": null,
        "XS_COOKIE": null,
        "DBPROXY_USERNAME": null,
        "DV_ACTION_CODE": null,
        "OLS_PRIVILEGES_USED": null,
        "RMAN_DEVICE_TYPE": null,
        "XS_NS_ATTRIBUTE_OLD_VAL": null,
        "TARGET_USER": null,
        "XS_ENTITY_TYPE": null,
        "ENTRY_ID": 1,
        "XS_PROCEDURE_NAME": null,
        "XS_INACTIVITY_TIMEOUT": null,
        "RMAN_OBJECT_TYPE": null,
        "SYSTEM_PRIVILEGE": null,
        "NEW_SCHEMA": null,
        "SCN": 5124715
    }
}
```

Il seguente record di evento di attività mostra un errore di accesso per il database SQL Server.

```
{
    "type": "DatabaseActivityMonitoringRecord",
    "clusterId": "",
    "instanceId": "db-4JCWQLUZVFYP7DIWP6JVQ77O3Q",
    "databaseActivityEventList": [
        {
            "class": "LOGIN",
            "clientApplication": "Microsoft SQL Server Management Studio",
            "command": "LOGIN FAILED",
```

```
            "commandText": "Login failed for user 'test'. Reason: Password did not
 match that for the login provided. [CLIENT: local-machine]",
            "databaseName": "",
            "dbProtocol": "SQLSERVER",
            "dbUserName": "test",
            "endTime": null,
            "errorMessage": null,
            "exitCode": 0,
            "logTime": "2022-10-06 21:34:42.7113072+00",
            "netProtocol": null,
            "objectName": "",
            "objectType": "LOGIN",
            "paramList": null,
            "pid": null,
            "remoteHost": "local machine",
            "remotePort": null,
            "rowCount": 0,
            "serverHost": "172.31.30.159",
            "serverType": "SQLSERVER",
            "serverVersion": "15.00.4073.23.v1.R1",
            "serviceName": "sqlserver-ee",
            "sessionId": 0,
            "startTime": null,
            "statementId": "0x1eb0d1808d34a94b9d3dcf5432750f02",
            "substatementId": 1,
            "transactionId": "0",
            "type": "record",
            "engineNativeAuditFields": {
                "target_database_principal_id": 0,
                "target_server_principal_id": 0,
                "target_database_principal_name": "",
                "server_principal_id": 0,
                "user_defined_information": "",
                "response_rows": 0,
                "database_principal_name": "",
                "target_server_principal_name": "",
                "schema_name": "",
                "is_column_permission": false,
                "object_id": 0,
                "server_instance_name": "EC2AMAZ-NFUJJNO",
                "target_server_principal_sid": null,
                "additional_information": "<action_info "xmlns=\"http://
schemas.microsoft.com/sqlserver/2008/sqlaudit_data\"><pooled_connection>0</
```

```
pooled_connection><error>0x00004818</error><state>8</state><address>local machine</
address><PasswordFirstNibbleHash>B</PasswordFirstNibbleHash></action_info>"-->,
                "duration_milliseconds": 0,
                "permission_bitmask": "0x00000000000000000000000000000000",
                "data_sensitivity_information": "",
                "session_server_principal_name": "",
                "connection_id": "98B4F537-0F82-49E3-AB08-B9D33B5893EF",
                "audit_schema_version": 1,
                "database_principal_id": 0,
                "server_principal_sid": null,
                "user_defined_event_id": 0,
                "host_name": "EC2AMAZ-NFUJJNO"
            }
        }
    ]
}
```

> **Note**
>
> Se un flusso di attività del database non è abilitato, l'ultimo campo nel documento JSON è `"engineNativeAuditFields": { }`.

Example Record di evento attività di un'istruzione CREATE TABLE

Il seguente esempio mostra un evento `CREATE TABLE` per il database Oracle.

```
{
    "class": "Standard",
    "clientApplication": "sqlplus@ip-12-34-5-678 (TNS V1-V3)",
    "command": "CREATE TABLE",
    "commandText": "CREATE TABLE persons(\n    person_id NUMBER GENERATED BY DEFAULT AS
 IDENTITY,\n    first_name VARCHAR2(50) NOT NULL,\n    last_name VARCHAR2(50) NOT NULL,
\n    PRIMARY KEY(person_id)\n)",
    "dbid": "0123456789",
    "databaseName": "ORCL",
    "dbProtocol": "oracle",
    "dbUserName": "TEST",
    "endTime": null,
    "errorMessage": null,
    "exitCode": 0,
    "logTime": "2021-01-15 00:22:49.535239",
```

```
    "netProtocol": "beq",
    "objectName": "PERSONS",
    "objectType": "TEST",
    "paramList": [],
    "pid": 17687,
    "remoteHost": "123.456.789.0",
    "remotePort": null,
    "rowCount": null,
    "serverHost": "987.654.321.01",
    "serverType": "oracle",
    "serverVersion": "19.0.0.0.ru-2020-01.rur-2020-01.r1.EE.3",
    "serviceName": "oracle-ee",
    "sessionId": 1234567890,
    "startTime": null,
    "statementId": 43,
    "substatementId": null,
    "transactionId": "090011007F0D0000",
    "engineNativeAuditFields": {
        "UNIFIED_AUDIT_POLICIES": "TEST_POL_EVERYTHING",
        "FGA_POLICY_NAME": null,
        "DV_OBJECT_STATUS": null,
        "SYSTEM_PRIVILEGE_USED": "CREATE SEQUENCE, CREATE TABLE",
        "OLS_LABEL_COMPONENT_TYPE": null,
        "XS_SESSIONID": null,
        "ADDITIONAL_INFO": null,
        "INSTANCE_ID": 1,
        "DV_COMMENT": null,
        "RMAN_SESSION_STAMP": null,
        "NEW_NAME": null,
        "DV_ACTION_NAME": null,
        "OLS_PROGRAM_UNIT_NAME": null,
        "OLS_STRING_LABEL": null,
        "RMAN_SESSION_RECID": null,
        "OBJECT_PRIVILEGES": null,
        "OLS_OLD_VALUE": null,
        "XS_TARGET_PRINCIPAL_NAME": null,
        "XS_NS_ATTRIBUTE": null,
        "XS_NS_NAME": null,
        "DBLINK_INFO": null,
        "AUTHENTICATION_TYPE": "(TYPE\u003d(DATABASE));(CLIENT ADDRESS\u003d((PROTOCOL
\u003dbeq)(HOST\u003d123.456.789.0)));",
        "OBJECT_EDITION": null,
        "OLS_PRIVILEGES_GRANTED": null,
        "EXCLUDED_USER": null,
```

```
        "DV_ACTION_OBJECT_NAME": null,
        "OLS_LABEL_COMPONENT_NAME": null,
        "EXCLUDED_SCHEMA": null,
        "DP_TEXT_PARAMETERS1": null,
        "XS_USER_NAME": null,
        "XS_ENABLED_ROLE": null,
        "XS_NS_ATTRIBUTE_NEW_VAL": null,
        "DIRECT_PATH_NUM_COLUMNS_LOADED": null,
        "AUDIT_OPTION": null,
        "DV_EXTENDED_ACTION_CODE": null,
        "XS_PACKAGE_NAME": null,
        "OLS_NEW_VALUE": null,
        "DV_RETURN_CODE": null,
        "XS_CALLBACK_EVENT_TYPE": null,
        "USERHOST": "ip-10-13-0-122",
        "GLOBAL_USERID": null,
        "CLIENT_IDENTIFIER": null,
        "RMAN_OPERATION": null,
        "TERMINAL": "pts/1",
        "OS_USERNAME": "rdsdb",
        "OLS_MAX_READ_LABEL": null,
        "XS_PROXY_USER_NAME": null,
        "XS_DATASEC_POLICY_NAME": null,
        "DV_FACTOR_CONTEXT": null,
        "OLS_MAX_WRITE_LABEL": null,
        "OLS_PARENT_GROUP_NAME": null,
        "EXCLUDED_OBJECT": null,
        "DV_RULE_SET_NAME": null,
        "EXTERNAL_USERID": null,
        "EXECUTION_ID": null,
        "ROLE": null,
        "PROXY_SESSIONID": 0,
        "DP_BOOLEAN_PARAMETERS1": null,
        "OLS_POLICY_NAME": null,
        "OLS_GRANTEE": null,
        "OLS_MIN_WRITE_LABEL": null,
        "APPLICATION_CONTEXTS": null,
        "XS_SCHEMA_NAME": null,
        "DV_GRANTEE": null,
        "XS_COOKIE": null,
        "DBPROXY_USERNAME": null,
        "DV_ACTION_CODE": null,
        "OLS_PRIVILEGES_USED": null,
        "RMAN_DEVICE_TYPE": null,
```

```
        "XS_NS_ATTRIBUTE_OLD_VAL": null,
        "TARGET_USER": null,
        "XS_ENTITY_TYPE": null,
        "ENTRY_ID": 12,
        "XS_PROCEDURE_NAME": null,
        "XS_INACTIVITY_TIMEOUT": null,
        "RMAN_OBJECT_TYPE": null,
        "SYSTEM_PRIVILEGE": null,
        "NEW_SCHEMA": null,
        "SCN": 5133083
    }
}
```

Il seguente esempio mostra un evento CREATE TABLE per il database SQL Server.

```
{
    "type": "DatabaseActivityMonitoringRecord",
    "clusterId": "",
    "instanceId": "db-4JCWQLUZVFYP7DIWP6JVQ77O3Q",
    "databaseActivityEventList": [
        {
            "class": "SCHEMA",
            "clientApplication": "Microsoft SQL Server Management Studio - Query",
            "command": "ALTER",
            "commandText": "Create table [testDB].[dbo].[TestTable2](\r\ntextA
 varchar(6000),\r\n    textB varchar(6000)\r\n)",
            "databaseName": "testDB",
            "dbProtocol": "SQLSERVER",
            "dbUserName": "test",
            "endTime": null,
            "errorMessage": null,
            "exitCode": 1,
            "logTime": "2022-10-06 21:44:38.4120677+00",
            "netProtocol": null,
            "objectName": "dbo",
            "objectType": "SCHEMA",
            "paramList": null,
            "pid": null,
            "remoteHost": "local machine",
            "remotePort": null,
            "rowCount": 0,
            "serverHost": "172.31.30.159",
            "serverType": "SQLSERVER",
```

```
            "serverVersion": "15.00.4073.23.v1.R1",
            "serviceName": "sqlserver-ee",
            "sessionId": 84,
            "startTime": null,
            "statementId": "0x5178d33d56e95e419558b9607158a5bd",
            "substatementId": 1,
            "transactionId": "4561864",
            "type": "record",
            "engineNativeAuditFields": {
                "target_database_principal_id": 0,
                "target_server_principal_id": 0,
                "target_database_principal_name": "",
                "server_principal_id": 2,
                "user_defined_information": "",
                "response_rows": 0,
                "database_principal_name": "dbo",
                "target_server_principal_name": "",
                "schema_name": "",
                "is_column_permission": false,
                "object_id": 1,
                "server_instance_name": "EC2AMAZ-NFUJJNO",
                "target_server_principal_sid": null,
                "additional_information": "",
                "duration_milliseconds": 0,
                "permission_bitmask": "0x00000000000000000000000000000000",
                "data_sensitivity_information": "",
                "session_server_principal_name": "test",
                "connection_id": "EE1FE3FD-EF2C-41FD-AF45-9051E0CD983A",
                "audit_schema_version": 1,
                "database_principal_id": 1,
                "server_principal_sid":
 "0x0105000000000005150000000bdc2795e2d0717901ba6998cf4010000",
                "user_defined_event_id": 0,
                "host_name": "EC2AMAZ-NFUJJNO"
            }
        }
    ]
}
```


Example Record di evento attività di un'istruzione SELECT

Il seguente esempio mostra un evento SELECT per il database Oracle.


```
{
```

```
    "class": "Standard",
    "clientApplication": "sqlplus@ip-12-34-5-678 (TNS V1-V3)",
    "command": "SELECT",
    "commandText": "select count(*) from persons",
    "databaseName": "1234567890",
    "dbProtocol": "oracle",
    "dbUserName": "TEST",
    "endTime": null,
    "errorMessage": null,
    "exitCode": 0,
    "logTime": "2021-01-15 00:25:18.850375",
    "netProtocol": "beq",
    "objectName": "PERSONS",
    "objectType": "TEST",
    "paramList": [],
    "pid": 17687,
    "remoteHost": "123.456.789.0",
    "remotePort": null,
    "rowCount": null,
    "serverHost": "987.654.321.09",
    "serverType": "oracle",
    "serverVersion": "19.0.0.0.ru-2020-01.rur-2020-01.r1.EE.3",
    "serviceName": "oracle-ee",
    "sessionId": 1080639707,
    "startTime": null,
    "statementId": 44,
    "substatementId": null,
    "transactionId": null,
    "engineNativeAuditFields": {
        "UNIFIED_AUDIT_POLICIES": "TEST_POL_EVERYTHING",
        "FGA_POLICY_NAME": null,
        "DV_OBJECT_STATUS": null,
        "SYSTEM_PRIVILEGE_USED": null,
        "OLS_LABEL_COMPONENT_TYPE": null,
        "XS_SESSIONID": null,
        "ADDITIONAL_INFO": null,
        "INSTANCE_ID": 1,
        "DV_COMMENT": null,
        "RMAN_SESSION_STAMP": null,
        "NEW_NAME": null,
        "DV_ACTION_NAME": null,
        "OLS_PROGRAM_UNIT_NAME": null,
        "OLS_STRING_LABEL": null,
        "RMAN_SESSION_RECID": null,
```

```
        "OBJECT_PRIVILEGES": null,
        "OLS_OLD_VALUE": null,
        "XS_TARGET_PRINCIPAL_NAME": null,
        "XS_NS_ATTRIBUTE": null,
        "XS_NS_NAME": null,
        "DBLINK_INFO": null,
        "AUTHENTICATION_TYPE": "(TYPE\u003d(DATABASE));(CLIENT ADDRESS\u003d((PROTOCOL
\u003dbeq)(HOST\u003d123.456.789.0)));",
        "OBJECT_EDITION": null,
        "OLS_PRIVILEGES_GRANTED": null,
        "EXCLUDED_USER": null,
        "DV_ACTION_OBJECT_NAME": null,
        "OLS_LABEL_COMPONENT_NAME": null,
        "EXCLUDED_SCHEMA": null,
        "DP_TEXT_PARAMETERS1": null,
        "XS_USER_NAME": null,
        "XS_ENABLED_ROLE": null,
        "XS_NS_ATTRIBUTE_NEW_VAL": null,
        "DIRECT_PATH_NUM_COLUMNS_LOADED": null,
        "AUDIT_OPTION": null,
        "DV_EXTENDED_ACTION_CODE": null,
        "XS_PACKAGE_NAME": null,
        "OLS_NEW_VALUE": null,
        "DV_RETURN_CODE": null,
        "XS_CALLBACK_EVENT_TYPE": null,
        "USERHOST": "ip-12-34-5-678",
        "GLOBAL_USERID": null,
        "CLIENT_IDENTIFIER": null,
        "RMAN_OPERATION": null,
        "TERMINAL": "pts/1",
        "OS_USERNAME": "rdsdb",
        "OLS_MAX_READ_LABEL": null,
        "XS_PROXY_USER_NAME": null,
        "XS_DATASEC_POLICY_NAME": null,
        "DV_FACTOR_CONTEXT": null,
        "OLS_MAX_WRITE_LABEL": null,
        "OLS_PARENT_GROUP_NAME": null,
        "EXCLUDED_OBJECT": null,
        "DV_RULE_SET_NAME": null,
        "EXTERNAL_USERID": null,
        "EXECUTION_ID": null,
        "ROLE": null,
        "PROXY_SESSIONID": 0,
        "DP_BOOLEAN_PARAMETERS1": null,
```

```
        "OLS_POLICY_NAME": null,
        "OLS_GRANTEE": null,
        "OLS_MIN_WRITE_LABEL": null,
        "APPLICATION_CONTEXTS": null,
        "XS_SCHEMA_NAME": null,
        "DV_GRANTEE": null,
        "XS_COOKIE": null,
        "DBPROXY_USERNAME": null,
        "DV_ACTION_CODE": null,
        "OLS_PRIVILEGES_USED": null,
        "RMAN_DEVICE_TYPE": null,
        "XS_NS_ATTRIBUTE_OLD_VAL": null,
        "TARGET_USER": null,
        "XS_ENTITY_TYPE": null,
        "ENTRY_ID": 13,
        "XS_PROCEDURE_NAME": null,
        "XS_INACTIVITY_TIMEOUT": null,
        "RMAN_OBJECT_TYPE": null,
        "SYSTEM_PRIVILEGE": null,
        "NEW_SCHEMA": null,
        "SCN": 5136972
    }
}
```

Il seguente esempio mostra un evento SELECT per il database SQL Server.

```
{
    "type": "DatabaseActivityMonitoringRecord",
    "clusterId": "",
    "instanceId": "db-4JCWQLUZVFYP7DIWP6JVQ77O3Q",
    "databaseActivityEventList": [
        {
            "class": "TABLE",
            "clientApplication": "Microsoft SQL Server Management Studio - Query",
            "command": "SELECT",
            "commandText": "select * from [testDB].[dbo].[TestTable]",
            "databaseName": "testDB",
            "dbProtocol": "SQLSERVER",
            "dbUserName": "test",
            "endTime": null,
            "errorMessage": null,
            "exitCode": 1,
            "logTime": "2022-10-06 21:24:59.9422268+00",
```

```
            "netProtocol": null,
            "objectName": "TestTable",
            "objectType": "TABLE",
            "paramList": null,
            "pid": null,
            "remoteHost": "local machine",
            "remotePort": null,
            "rowCount": 0,
            "serverHost": "172.31.30.159",
            "serverType": "SQLSERVER",
            "serverVersion": "15.00.4073.23.v1.R1",
            "serviceName": "sqlserver-ee",
            "sessionId": 62,
            "startTime": null,
            "statementId": "0x03baed90412f564fad640ebe51f89b99",
            "substatementId": 1,
            "transactionId": "4532935",
            "type": "record",
            "engineNativeAuditFields": {
                "target_database_principal_id": 0,
                "target_server_principal_id": 0,
                "target_database_principal_name": "",
                "server_principal_id": 2,
                "user_defined_information": "",
                "response_rows": 0,
                "database_principal_name": "dbo",
                "target_server_principal_name": "",
                "schema_name": "dbo",
                "is_column_permission": true,
                "object_id": 581577110,
                "server_instance_name": "EC2AMAZ-NFUJJNO",
                "target_server_principal_sid": null,
                "additional_information": "",
                "duration_milliseconds": 0,
                "permission_bitmask": "0x00000000000000000000000000000001",
                "data_sensitivity_information": "",
                "session_server_principal_name": "test",
                "connection_id": "AD3A5084-FB83-45C1-8334-E923459A8109",
                "audit_schema_version": 1,
                "database_principal_id": 1,
                "server_principal_sid":
 "0x0105000000000005150000000bdc2795e2d0717901ba6998cf4010000",
                "user_defined_event_id": 0,
                "host_name": "EC2AMAZ-NFUJJNO"
```

```
            }
        }
    ]
}
```

DatabaseActivityMonitoringRecords Oggetto JSON

I record di eventi dell'attività del database si trovano in un oggetto JSON che contiene le seguenti informazioni.

Campo JSON	Tipo di dati	Descrizione
`type`	stringa	Il tipo di record JSON. Il valore è `DatabaseActivityMonitoringRecords` .
`version`	stringa	La versione dei record di monitoraggio delle attività del database. Oracle DB utilizza la versione 1.3 e SQL Server utilizza la versione 1.4. Queste versioni del motore introducono l'oggetto JSON `engineNativeAuditFields` .
databaseActivityEvents	string	Un oggetto JSON contenente gli eventi di attività.
key	string	Una chiave di crittografia utilizzata per decrittare databaseActivityEventElenco

databaseActivityEvents Oggetto JSON

L'oggetto JSON `databaseActivityEvents` contiene le seguenti informazioni.

Campi di primo livello nel record JSON

Ogni evento nel registro di controllo viene racchiuso in un record in formato JSON. Questo record contiene i seguenti campi.

type

Questo campo ha sempre il valore `DatabaseActivityMonitoringRecords`.

versione

Questo campo rappresenta la versione del protocollo o del contratto di dati del flusso di attività del database. Definisce quali campi sono disponibili.

databaseActivityEvents

Stringa crittografata che rappresenta uno o più eventi di attività. È rappresentato come un array di byte base64. Quando si decrittografa la stringa, il risultato è un record in formato JSON con campi come illustrato negli esempi di questa sezione.

key

Chiave dati crittografata utilizzata per crittografare la stringa `databaseActivityEvents`. È lo stesso AWS KMS key che hai fornito quando hai avviato il flusso di attività del database.

Nell'esempio seguente viene illustrato il formato di questo record.

```
{
  "type":"DatabaseActivityMonitoringRecords",
  "version":"1.3",
  "databaseActivityEvents":"encrypted audit records",
  "key":"encrypted key"
}
```

```
          "type":"DatabaseActivityMonitoringRecords",
          "version":"1.4",
          "databaseActivityEvents":"encrypted audit records",
          "key":"encrypted key"
```

Per decrittografare il contenuto del campo `databaseActivityEvents`, procedere come segue:

1. Decrittare il valore nel campo `key` JSON utilizzando la chiave KMS fornita all'avvio del flusso di attività del database. In questo modo viene restituita la chiave di crittografia dei dati in testo non crittografato.

2. Decodificare il valore nel campo `databaseActivityEvents` JSON con base64 per ottenere il testo cifrato, in formato binario, del payload di controllo.
3. Decifrare il testo cifrato binario con la chiave di crittografia dei dati decodificata nel primo passaggio.
4. Decomprimere il payload decrittografato.
   - Il payload crittografato è nel campo `databaseActivityEvents`.
   - Il campo `databaseActivityEventList` contiene una matrice di record di controllo. I campi `type` nella matrice possono essere `record` o `heartbeat`.

Il record dell'evento attività registro di controllo è un oggetto JSON che contiene le seguenti informazioni.

Campo JSON	Tipo di dati	Descrizione
`type`	stringa	Il tipo di record JSON. Il valore è `DatabaseActivityMonitoringRecord` .
`instanceId`	stringa	Identificatore della risorsa istanza database. Corrisponde all'attributo di istanza database `DbiResourceId` .
databaseActivityEventElenco	stringa	Matrice di record di controllo delle attività o messaggi heartbeat.

databaseActivityEventElenca l'array JSON

Il payload del registro di controllo è un array JSON `databaseActivityEventList` crittografato. La tabella riporta in ordine alfabetico i campi per ogni evento di attività nella matrice `DatabaseActivityEventList` decrittata di un log di verifica.

Quando la verifica unificata è abilitata in Oracle Database, i record di verifica vengono popolati in questo nuovo percorso di verifica. La visualizzazione `UNIFIED_AUDIT_TRAIL` mostra i record di verifica in formato tabulare recuperando i record di verifica dal percorso di verifica. Quando si avvia un flusso di attività del database, una colonna in `UNIFIED_AUDIT_TRAIL` viene mappata a un campo nella matrice `databaseActivityEventList`.

> **Important**
>
> Tale struttura di eventi è soggetta a modifiche. Amazon RDS potrebbe aggiungere nuovi campi agli eventi di attività in futuro. Nelle applicazioni che analizzano i dati JSON, assicurarsi che il codice possa ignorare o eseguire le azioni appropriate per i nomi di campo sconosciuti.

databaseActivityEventElenca i campi per Amazon RDS for Oracle

Campo	Tipo di dati	Origine	Descrizione
`class`	string	Colonna `AUDIT_TYPE` in `UNIFIED_AUDIT_TRAIL`	La classe dell'evento attività. Questo corrisponde alla colonna `AUDIT_TYPE` nella visualizzazione `UNIFIED_AUDIT_TRAIL`. I valori validi per Amazon RDS for Oracle sono i seguenti: • `Standard` • `FineGrainedAudit` • `XS` • `Database Vault` • `Label Security` • `RMAN_AUDIT` • `Datapump` • `Direct path API` Per ulteriori informazioni, consulta UNIFIED_AUDIT_TRAIL nella

Campo	Tipo di dati	Origine	Descrizione
			documentazione di Oracle.
clientApplication	string	CLIENT_PROGRAM_NAME in UNIFIED_AUDIT_TRAIL	L'applicazione utilizzata dal client per eseguire la connessione come segnalato dal client. Il client non deve fornire queste informazioni, pertanto il valore può essere nullo. Un valore di esempio è JDBC Thin Client.
command	string	Colonna ACTION_NAME in UNIFIED_AUDIT_TRAIL	Nome dell'azione eseguita dall'utente. Per comprendere l'azione completa, leggere sia il nome del comando che il valore AUDIT_TYPE . Un valore di esempio è ALTER DATABASE.
commandText	string	Colonna SQL_TEXT in UNIFIED_AUDIT_TRAIL	L'istruzione SQL associata all'evento. Un valore di esempio è ALTER DATABASE BEGIN BACKUP.
databaseName	string	Colonna NAME in V$DATABASE	Nome del database.

Campo	Tipo di dati	Origine	Descrizione
dbid	nume	Colonna DBID in UNIFIED_AUDIT_TRAIL	Identificatore numerico per il database. Un valore di esempio è 1559204751 .
dbProtocol	string	N/A	Il protocollo di database. In questa beta, il valore è oracle.
dbUserName	string	Colonna DBUSERNAME in UNIFIED_AUDIT_TRAIL	Il nome dell'utente del database le cui azioni sono state verificate. Un valore di esempio è RDSADMIN.
endTime	string	N/A	Questo campo non viene utilizzato per RDS for Oracle ed è sempre null.

Campo	Tipo di dati	Origine	Descrizione
`engineNativeAuditFields`	oggetto	`UNIFIED_AUDIT_TRAIL`	Per impostazione predefinita, è vuoto. Quando si avvia il flusso di attività con l'opzione `--engine-native-audit-fields-included` , questo oggetto include le seguenti colonne e i relativi valori: `ADDITIONAL_INFO` `APPLICATION_CONTEXTS` `AUDIT_OPTION` `AUTHENTICATION_TYPE` `CLIENT_IDENTIFIER` `CURRENT_USER` `DBLINK_INFO` `DBPROXY_USERNAME` `DIRECT_PATH_NUM_COLUMNS_LOADED` `DP_BOOLEAN_PARAMETERS1` `DP_TEXT_PARAMETERS1` `DV_ACTION_CODE` `DV_ACTION_NAME` `DV_ACTION_OBJECT_NAME` `DV_COMMENT` `DV_EXTENDED_ACTION_CODE` `DV_FACTOR_CONTEXT` `DV_GRANTEE` `DV_OBJECT_STATUS`

Campo	Tipo di dati	Origine	Descrizione
			DV_RETURN_CODE DV_RULE_SET_NAME ENTRY_ID EXCLUDED_OBJECT EXCLUDED_SCHEMA EXCLUDED_USER EXECUTION_ID EXTERNAL_USERID FGA_POLICY_NAME GLOBAL_USERID INSTANCE_ID KSACL_SER VICE_NAME KSACL_SOURCE_LOCA TION KSACL_USER_NAME NEW_NAME NEW_SCHEMA OBJECT_EDITION OBJECT_PRIVILEGES OLS_GRANTEE OLS_LABEL_COM PONENT_NAME OLS_LABEL_COMPO NENT_TYPE OLS_MAX_READ_LABEL OLS_MAX_WRITE_LA BEL OLS_MIN_WRITE_ LABEL OLS_NEW_VALUE OLS_OLD_VALUE OLS_PARENT_ GROUP_NAME OLS_POLICY_NAME OLS_PRIVILEGES_GRA NTED OLS_PRIVILEGE S_USED

Campo	Tipo di dati	Origine	Descrizione
			OLS_PROGRAM _UNIT_NAME OLS_STRING_LABEL OS_USERNAME PROTOCOL_ACTIO N_NAME PROTOCOL_MESSAGE PROTOCOL_RET URN_CODE PROTOCOL_SESSION_I D PROTOCOL_USERHOST PROXY_SESSIONID RLS_INFO RMAN_DEVICE_TYPE RMAN_OBJECT_TYPE RMAN_OPERATION RMAN_SESSION_RECID RMAN_SESSION_STAMP ROLE SCN SYSTEM_PRIVILEGE SYSTEM_PRIVIL EGE_USED TARGET_USER TERMINAL UNIFIED_AUDIT_P OLICIES USERHOST XS_CALLBAC K_EVENT_TYPE XS_COOKIE XS_DATASEC_PO LICY_NAME XS_ENABLED_ROLE XS_ENTITY_TYPE XS_INACTIVITY _TIMEOUT XS_NS_ATTRIBUTE

Campo	Tipo di dati	Origine	Descrizione
			`XS_NS_ATTRI` `BUTE_NEW_VAL` `XS_NS_ATTRIBUT` `E_OLD_VAL` `XS_NS_NAME` `XS_PACKAGE_NAME` `XS_PROCEDURE_NAME` `XS_PROXY_USER_NAME` `XS_SCHEMA_NAME` `XS_SESSIONID` `XS_TARGET_PRINC` `IPAL_NAME` `XS_USER_NAME` Per ulteriori informazioni, consulta UNIFIED_AUDIT_TRAIL nella documentazione di Oracle Database.
`errorMessage`	string	N/A	Questo campo non viene utilizzato per RDS for Oracle ed è sempre null.
`exitCode`	nume	Colonna `RETURN_CODE` in `UNIFIED_AUDIT_TRAIL`	Codice di errore di Oracle Database generato dall'operazione. Se l'azione ha avuto esito positivo, il valore è `0`.

Campo	Tipo di dati	Origine	Descrizione
logTime	string	Colonna EVENT_TIMESTAMP_UTC in UNIFIED_AUDIT_TRAIL	Timestamp della creazione della voce del percorso di verifica. Un valore di esempio è 2020-11-27 06:56:14.981404 .
netProtocol	string	Colonna AUTHENTICATION_TYPE in UNIFIED_AUDIT_TRAIL	Il protocollo di comunicazione di rete. Un valore di esempio è TCP.
objectName	string	Colonna OBJECT_NAME in UNIFIED_AUDIT_TRAIL	Il nome dell'oggetto interessato dall'operazione. Un valore di esempio è employees .
objectType	string	Colonna OBJECT_SCHEMA in UNIFIED_AUDIT_TRAIL	Il nome dello schema dell'oggetto interessato dall'operazione. Un valore di esempio è hr.
paramList	elenc	Colonna SQL_BINDS in UNIFIED_AUDIT_TRAIL	L'elenco delle variabili di bind, se presenti, associate a SQL_TEXT. Un valore di esempio è parameter_1,parameter_2 .
pid	nume	Colonna OS_PROCESS in UNIFIED_AUDIT_TRAIL	L'identificatore del processo del sistema operativo del processo di database Oracle. Un valore di esempio è 22396.

Campo	Tipo di dati	Origine	Descrizione
remoteHost	string	Colonna AUTHENTICATION_TYPE in UNIFIED_AUDIT_TRAIL	L'indirizzo IP o il nome dell'host da cui è stata generata la sessione. Un valore di esempio è 123.456.789.123 .
remotePort	string	Colonna AUTHENTICATION_TYPE in UNIFIED_AUDIT_TRAIL	Il numero di porta del client. Un valore tipico negli ambienti Oracle Database è 1521.
rowCount	nume	N/A	Questo campo non viene utilizzato per RDS for Oracle ed è sempre null.
serverHost	string	Host database	L'indirizzo IP dell'host del server di database. Un valore di esempio è 123.456.789.123 .
serverType	string	N/A	Il tipo di server di database. Il valore è sempre ORACLE.
serverVersion	string	Host database	La versione, il Release Update (RU) e la Release Update Revision (RUR) di Amazon RDS for Oracle. Un valore di esempio è 19.0.0.0.ru-2020-01.rur-2020-01.r1.EE.3 .

Campo	Tipo di dati	Origine	Descrizione
`serviceName`	string	Host database	Il nome del servizio Un valore di esempio è `oracle-ee` .
`sessionId`	nume	Colonna `SESSIONID` in `UNIFIED_AUDIT_TRAIL`	L'identificatore di sessione della verifica. Un esempio è `1894327130` .
`startTime`	string	N/A	Questo campo non viene utilizzato per RDS for Oracle ed è sempre null.
`statementId`	nume	Colonna `STATEMENT_ID` in `UNIFIED_AUDIT_TRAIL`	ID numerico per ogni esecuzione di istruzione. Un'istruzione può causare molte azioni. Un valore di esempio è `142197`.
`substatementId`	N/D	N/D	Questo campo non viene utilizzato per RDS for Oracle ed è sempre null.
`transactionId`	string	Colonna `TRANSACTION_ID` in `UNIFIED_AUDIT_TRAIL`	L'identificatore della transazione in cui l'oggetto viene modificato. Un valore di esempio è `02000800D5030000` .

databaseActivityEventElenca i campi per Amazon RDS for SQL Server

Campo	Tipo di dati	Origine	Descrizione
`class`	stringa	`sys.fn_get_audit_file.class_type` mappato a `sys.dm_audit_class_type_map.class_type_desc`	La classe dell'evento attività. Per ulteriori informazioni, consulta Controllo in SQL Server (motore del database) nella documentazione di Microsoft.
`clientApplication`	string	`sys.fn_get_audit_file.application_name`	L'applicazione a cui il client si connette come indicato dal client (SQL Server versione 14 e successive). Questo campo è nullo in SQL Server versione 13.
`command`	string	`sys.fn_get_audit_file.action_id` mappato a `sys.dm_audit_actions.name`	Categoria generale dell'istruzione SQL. I valori di questo campo dipendono dal valore della classe.
`commandText`	string	`sys.fn_get_audit_file.statement`	Questo campo indica l'istruzione SQL.
`databaseName`	string	`sys.fn_get_audit_file.database_name`	Nome del database.
`dbProtocol`	string	N/A	Il protocollo di database. Il valore è `SQLSERVER` .
`dbUserName`	string	`sys.fn_get_audit_file.server_principal_name`	L'utente del database per l'autenticazione del client.
`endTime`	string	N/A	Questo campo non è utilizzato da Amazon RDS per SQL Server e il valore è nullo.

Campo	Tipo di dati	Origine	Descrizione
`engineNativeAuditFields`	oggetto	Ogni campo presente in `sys.fn_get_audit_file` che non è elencato in questa colonna.	Per impostazione predefinita, è vuoto. Quando avvii il flusso di attività con l'opzione `--engine-native-audit-fields-included`, questo oggetto include altri campi di controllo nativi del motore, che non vengono restituiti da questa mappa JSON.
`errorMessage`	string	N/A	Questo campo non è utilizzato da Amazon RDS per SQL Server e il valore è nullo.
`exitCode`	integer	`sys.fn_get_audit_file.succeeded`	Indica se l'azione che ha avviato l'evento è stata completata. Questo campo non può essere nullo. Per tutti gli eventi, tranne quelli di accesso, questo campo indica se il controllo delle autorizzazioni è riuscito o meno, ma non se l'operazione è riuscita o meno. I valori includono: • `0` - Non riuscito • `1` - Riuscito
`logTime`	string	`sys.fn_get_audit_file.event_time`	Il timestamp dell'evento registrato da SQL Server.
`netProtocol`	string	N/A	Questo campo non è utilizzato da Amazon RDS per SQL Server e il valore è nullo.

Campo	Tipo di dati	Origine	Descrizione
objectName	string	sys.fn_get_audit_file.object_name	Il nome dell'oggetto di database se l'istruzione SQL viene eseguita su un oggetto.
objectType	string	sys.fn_get_audit_file.class_type mappato a sys.dm_audit_class_type_map.class_type_desc	Il tipo di oggetto di database se l'istruzione SQL viene eseguita su un oggetto.
paramList	string	N/A	Questo campo non è utilizzato da Amazon RDS per SQL Server e il valore è nullo.
pid	integer	N/D	Questo campo non è utilizzato da Amazon RDS per SQL Server e il valore è nullo.
remoteHost	string	sys.fn_get_audit_file.client_ip	L'indirizzo IP o il nome host del client che ha emesso l'istruzione SQL (SQL Server versione 14 e successive). Questo campo è nullo in SQL Server versione 13.
remotePort	integer	N/D	Questo campo non è utilizzato da Amazon RDS per SQL Server e il valore è nullo.
rowCount	integer	sys.fn_get_audit_file.affected_rows	Il numero di righe della tabella interessate dall'istruzione SQL (SQL Server versione 14 e successive). Questo campo è in SQL Server versione 13.
serverHost	string	Host database	L'indirizzo IP del server di database di host.

Campo	Tipo di dati	Origine	Descrizione
`serverType`	string	N/A	Il tipo di server di database. Il valore è `SQLSERVER` .
`serverVersion`	string	Host database	La versione di server di database, ad esempio 15.00.4073.23.v1.R1 per SQL Server 2017.
`serviceName`	string	Host database	Il nome del servizio Un valore di esempio è `sqlserver-ee` .
`sessionId`	integer	`sys.fn_get_audit_file.session_id`	Identificatore univoco della sessione.
`startTime`	string	N/A	Questo campo non è utilizzato da Amazon RDS per SQL Server e il valore è nullo.
`statementId`	string	`sys.fn_get_audit_file.sequence_group_id`	Identificatore univoco per l'istruzione SQL del client. L'identificatore è diverso per ogni evento generato. Un valore di esempio è `0x38eaf41562671840 94bb82071aaab644` .
`substatementId`	integer	`sys.fn_get_audit_file.sequence_number`	Identificatore per determinare il numero di sequenza di una dichiarazione. Questo identificatore è utile quando i record di grandi dimensioni vengono suddivisi in più record.
`transactionId`	integer	`sys.fn_get_audit_file.transaction_id`	Identificatore di una transazione. Se non ci sono transazioni attive, il valore è zero.

Campo	Tipo di dati	Origine	Descrizione
`type`	string	Flussi di attività di database generati	Tipo di evento. I valori sono `record` o `heartbeat` .

Elaborazione di un flusso di attività del database utilizzando l'SDK AWS

Puoi elaborare a livello di codice un flusso di attività utilizzando l'SDK. AWS Di seguito sono riportati esempi Java e Python completamente funzionanti dell'utilizzo dei record dei flussi di attività del database per l'abilitazione basata sull'istanza.

Java

```
import java.io.ByteArrayInputStream;
import java.io.ByteArrayOutputStream;
import java.io.IOException;
import java.net.InetAddress;
import java.nio.ByteBuffer;
import java.nio.charset.StandardCharsets;
import java.security.NoSuchAlgorithmException;
import java.security.NoSuchProviderException;
import java.security.Security;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
import java.util.UUID;
import java.util.zip.GZIPInputStream;

import javax.crypto.Cipher;
import javax.crypto.NoSuchPaddingException;
import javax.crypto.spec.SecretKeySpec;

import com.amazonaws.auth.AWSStaticCredentialsProvider;
import com.amazonaws.auth.BasicAWSCredentials;
import com.amazonaws.encryptionsdk.AwsCrypto;
import com.amazonaws.encryptionsdk.CryptoInputStream;
import com.amazonaws.encryptionsdk.jce.JceMasterKey;
import
 com.amazonaws.services.kinesis.clientlibrary.exceptions.InvalidStateException;
import com.amazonaws.services.kinesis.clientlibrary.exceptions.ShutdownException;
```

```
import com.amazonaws.services.kinesis.clientlibrary.exceptions.ThrottlingException;
import com.amazonaws.services.kinesis.clientlibrary.interfaces.IRecordProcessor;
import
 com.amazonaws.services.kinesis.clientlibrary.interfaces.IRecordProcessorCheckpointer;
import
 com.amazonaws.services.kinesis.clientlibrary.interfaces.IRecordProcessorFactory;
import
 com.amazonaws.services.kinesis.clientlibrary.lib.worker.InitialPositionInStream;
import
 com.amazonaws.services.kinesis.clientlibrary.lib.worker.KinesisClientLibConfiguration;
import com.amazonaws.services.kinesis.clientlibrary.lib.worker.ShutdownReason;
import com.amazonaws.services.kinesis.clientlibrary.lib.worker.Worker;
import com.amazonaws.services.kinesis.clientlibrary.lib.worker.Worker.Builder;
import com.amazonaws.services.kinesis.model.Record;
import com.amazonaws.services.kms.AWSKMS;
import com.amazonaws.services.kms.AWSKMSClientBuilder;
import com.amazonaws.services.kms.model.DecryptRequest;
import com.amazonaws.services.kms.model.DecryptResult;
import com.amazonaws.util.Base64;
import com.amazonaws.util.IOUtils;
import com.google.gson.Gson;
import com.google.gson.GsonBuilder;
import com.google.gson.annotations.SerializedName;
import org.bouncycastle.jce.provider.BouncyCastleProvider;

public class DemoConsumer {

    private static final String STREAM_NAME = "aws-rds-das-[instance-external-
resource-id]"; // aws-rds-das-db-ABCD123456
    private static final String APPLICATION_NAME = "AnyApplication"; //unique
 application name for dynamo table generation that holds kinesis shard tracking
    private static final String AWS_ACCESS_KEY =
 "[AWS_ACCESS_KEY_TO_ACCESS_KINESIS]";
    private static final String AWS_SECRET_KEY =
 "[AWS_SECRET_KEY_TO_ACCESS_KINESIS]";
    private static final String RESOURCE_ID = "[external-resource-id]"; // db-
ABCD123456
    private static final String REGION_NAME = "[region-name]"; //us-east-1, us-
east-2...
    private static final BasicAWSCredentials CREDENTIALS = new
 BasicAWSCredentials(AWS_ACCESS_KEY, AWS_SECRET_KEY);
    private static final AWSStaticCredentialsProvider CREDENTIALS_PROVIDER = new
 AWSStaticCredentialsProvider(CREDENTIALS);
```

```
    private static final AwsCrypto CRYPTO = new AwsCrypto();
    private static final AWSKMS KMS = AWSKMSClientBuilder.standard()
            .withRegion(REGION_NAME)
            .withCredentials(CREDENTIALS_PROVIDER).build();

    class Activity {
        String type;
        String version;
        String databaseActivityEvents;
        String key;
    }

    class ActivityEvent {
        @SerializedName("class") String _class;
        String clientApplication;
        String command;
        String commandText;
        String databaseName;
        String dbProtocol;
        String dbUserName;
        String endTime;
        String errorMessage;
        String exitCode;
        String logTime;
        String netProtocol;
        String objectName;
        String objectType;
        List<String> paramList;
        String pid;
        String remoteHost;
        String remotePort;
        String rowCount;
        String serverHost;
        String serverType;
        String serverVersion;
        String serviceName;
        String sessionId;
        String startTime;
        String statementId;
        String substatementId;
        String transactionId;
        String type;
    }
```

```
   class ActivityRecords {
       String type;
       String clusterId; // note that clusterId will contain an empty string on RDS
Oracle and RDS SQL Server
       String instanceId;
       List<ActivityEvent> databaseActivityEventList;
   }

   static class RecordProcessorFactory implements IRecordProcessorFactory {
       @Override
       public IRecordProcessor createProcessor() {
           return new RecordProcessor();
       }
   }

   static class RecordProcessor implements IRecordProcessor {

       private static final long BACKOFF_TIME_IN_MILLIS = 3000L;
       private static final int PROCESSING_RETRIES_MAX = 10;
       private static final long CHECKPOINT_INTERVAL_MILLIS = 60000L;
       private static final Gson GSON = new
GsonBuilder().serializeNulls().create();

       private static final Cipher CIPHER;
       static {
           Security.insertProviderAt(new BouncyCastleProvider(), 1);
           try {
               CIPHER = Cipher.getInstance("AES/GCM/NoPadding", "BC");
           } catch (NoSuchAlgorithmException | NoSuchPaddingException |
NoSuchProviderException e) {
               throw new ExceptionInInitializerError(e);
           }
       }

       private long nextCheckpointTimeInMillis;

       @Override
       public void initialize(String shardId) {
       }

       @Override
       public void processRecords(final List<Record> records, final
IRecordProcessorCheckpointer checkpointer) {
           for (final Record record : records) {
```

```
                processSingleBlob(record.getData());
            }

            if (System.currentTimeMillis() > nextCheckpointTimeInMillis) {
                checkpoint(checkpointer);
                nextCheckpointTimeInMillis = System.currentTimeMillis() +
CHECKPOINT_INTERVAL_MILLIS;
            }
        }

        @Override
        public void shutdown(IRecordProcessorCheckpointer checkpointer,
ShutdownReason reason) {
            if (reason == ShutdownReason.TERMINATE) {
                checkpoint(checkpointer);
            }
        }

        private void processSingleBlob(final ByteBuffer bytes) {
            try {
                // JSON $Activity
                final Activity activity = GSON.fromJson(new String(bytes.array(),
StandardCharsets.UTF_8), Activity.class);

                // Base64.Decode
                final byte[] decoded =
Base64.decode(activity.databaseActivityEvents);
                final byte[] decodedDataKey = Base64.decode(activity.key);

                Map<String, String> context = new HashMap<>();
                context.put("aws:rds:db-id", RESOURCE_ID);

                // Decrypt
                final DecryptRequest decryptRequest = new DecryptRequest()

.withCiphertextBlob(ByteBuffer.wrap(decodedDataKey)).withEncryptionContext(context);
                final DecryptResult decryptResult = KMS.decrypt(decryptRequest);
                final byte[] decrypted = decrypt(decoded,
getByteArray(decryptResult.getPlaintext()));

                // GZip Decompress
                final byte[] decompressed = decompress(decrypted);
                // JSON $ActivityRecords
```

```
                final ActivityRecords activityRecords = GSON.fromJson(new
String(decompressed, StandardCharsets.UTF_8), ActivityRecords.class);

                // Iterate throught $ActivityEvents
                for (final ActivityEvent event :
activityRecords.databaseActivityEventList) {
                    System.out.println(GSON.toJson(event));
                }
            } catch (Exception e) {
                // Handle error.
                e.printStackTrace();
            }
        }

        private static byte[] decompress(final byte[] src) throws IOException {
            ByteArrayInputStream byteArrayInputStream = new
ByteArrayInputStream(src);
            GZIPInputStream gzipInputStream = new
GZIPInputStream(byteArrayInputStream);
            return IOUtils.toByteArray(gzipInputStream);
        }

        private void checkpoint(IRecordProcessorCheckpointer checkpointer) {
            for (int i = 0; i < PROCESSING_RETRIES_MAX; i++) {
                try {
                    checkpointer.checkpoint();
                    break;
                } catch (ShutdownException se) {
                    // Ignore checkpoint if the processor instance has been shutdown
(fail over).
                    System.out.println("Caught shutdown exception, skipping
checkpoint." + se);
                    break;
                } catch (ThrottlingException e) {
                    // Backoff and re-attempt checkpoint upon transient failures
                    if (i >= (PROCESSING_RETRIES_MAX - 1)) {
                        System.out.println("Checkpoint failed after " + (i + 1) +
"attempts." + e);
                        break;
                    } else {
                        System.out.println("Transient issue when checkpointing -
attempt " + (i + 1) + " of " + PROCESSING_RETRIES_MAX + e);
                    }
                } catch (InvalidStateException e) {
```

```
                    // This indicates an issue with the DynamoDB table (check for
table, provisioned IOPS).
                    System.out.println("Cannot save checkpoint to the DynamoDB table
used by the Amazon Kinesis Client Library." + e);
                    break;
                }
                try {
                    Thread.sleep(BACKOFF_TIME_IN_MILLIS);
                } catch (InterruptedException e) {
                    System.out.println("Interrupted sleep" + e);
                }
            }
        }
    }

    private static byte[] decrypt(final byte[] decoded, final byte[] decodedDataKey)
throws IOException {
        // Create a JCE master key provider using the random key and an AES-GCM
encryption algorithm
        final JceMasterKey masterKey = JceMasterKey.getInstance(new
SecretKeySpec(decodedDataKey, "AES"),
                "BC", "DataKey", "AES/GCM/NoPadding");
        try (final CryptoInputStream<JceMasterKey> decryptingStream =
CRYPTO.createDecryptingStream(masterKey, new ByteArrayInputStream(decoded));
             final ByteArrayOutputStream out = new ByteArrayOutputStream()) {
            IOUtils.copy(decryptingStream, out);
            return out.toByteArray();
        }
    }

    public static void main(String[] args) throws Exception {
        final String workerId = InetAddress.getLocalHost().getCanonicalHostName() +
":" + UUID.randomUUID();
        final KinesisClientLibConfiguration kinesisClientLibConfiguration =
                new KinesisClientLibConfiguration(APPLICATION_NAME, STREAM_NAME,
CREDENTIALS_PROVIDER, workerId);

kinesisClientLibConfiguration.withInitialPositionInStream(InitialPositionInStream.LATEST);
        kinesisClientLibConfiguration.withRegionName(REGION_NAME);
        final Worker worker = new Builder()
                .recordProcessorFactory(new RecordProcessorFactory())
                .config(kinesisClientLibConfiguration)
                .build();
```

```
        System.out.printf("Running %s to process stream %s as worker %s...\n",
 APPLICATION_NAME, STREAM_NAME, workerId);

        try {
            worker.run();
        } catch (Throwable t) {
            System.err.println("Caught throwable while processing data.");
            t.printStackTrace();
            System.exit(1);
        }
        System.exit(0);
    }

    private static byte[] getByteArray(final ByteBuffer b) {
        byte[] byteArray = new byte[b.remaining()];
        b.get(byteArray);
        return byteArray;
    }
}
```

Python

```
import base64
import json
import zlib
import aws_encryption_sdk
from aws_encryption_sdk import CommitmentPolicy
from aws_encryption_sdk.internal.crypto import WrappingKey
from aws_encryption_sdk.key_providers.raw import RawMasterKeyProvider
from aws_encryption_sdk.identifiers import WrappingAlgorithm, EncryptionKeyType
import boto3

REGION_NAME = '<region>'                    # us-east-1
RESOURCE_ID = '<external-resource-id>'      # db-ABCD123456
STREAM_NAME = 'aws-rds-das-' + RESOURCE_ID  # aws-rds-das-db-ABCD123456

enc_client =
 aws_encryption_sdk.EncryptionSDKClient(commitment_policy=CommitmentPolicy.FORBID_ENCRYPT_AL

class MyRawMasterKeyProvider(RawMasterKeyProvider):
    provider_id = "BC"

    def __new__(cls, *args, **kwargs):
```

```
        obj = super(RawMasterKeyProvider, cls).__new__(cls)
        return obj

    def __init__(self, plain_key):
        RawMasterKeyProvider.__init__(self)
        self.wrapping_key =
 WrappingKey(wrapping_algorithm=WrappingAlgorithm.AES_256_GCM_IV12_TAG16_NO_PADDING,
                                        wrapping_key=plain_key,
 wrapping_key_type=EncryptionKeyType.SYMMETRIC)

    def _get_raw_key(self, key_id):
        return self.wrapping_key


def decrypt_payload(payload, data_key):
    my_key_provider = MyRawMasterKeyProvider(data_key)
    my_key_provider.add_master_key("DataKey")
    decrypted_plaintext, header = enc_client.decrypt(
        source=payload,

 materials_manager=aws_encryption_sdk.materials_managers.default.DefaultCryptoMaterialsManag
    return decrypted_plaintext


def decrypt_decompress(payload, key):
    decrypted = decrypt_payload(payload, key)
    return zlib.decompress(decrypted, zlib.MAX_WBITS + 16)


def main():
    session = boto3.session.Session()
    kms = session.client('kms', region_name=REGION_NAME)
    kinesis = session.client('kinesis', region_name=REGION_NAME)

    response = kinesis.describe_stream(StreamName=STREAM_NAME)
    shard_iters = []
    for shard in response['StreamDescription']['Shards']:
        shard_iter_response = kinesis.get_shard_iterator(StreamName=STREAM_NAME,
 ShardId=shard['ShardId'],

 ShardIteratorType='LATEST')
        shard_iters.append(shard_iter_response['ShardIterator'])

    while len(shard_iters) > 0:
```

```
        next_shard_iters = []
        for shard_iter in shard_iters:
            response = kinesis.get_records(ShardIterator=shard_iter, Limit=10000)
            for record in response['Records']:
                record_data = record['Data']
                record_data = json.loads(record_data)
                payload_decoded =
 base64.b64decode(record_data['databaseActivityEvents'])
                data_key_decoded = base64.b64decode(record_data['key'])
                data_key_decrypt_result =
 kms.decrypt(CiphertextBlob=data_key_decoded,

 EncryptionContext={'aws:rds:db-id': RESOURCE_ID})
                print (decrypt_decompress(payload_decoded,
 data_key_decrypt_result['Plaintext']))
            if 'NextShardIterator' in response:
                next_shard_iters.append(response['NextShardIterator'])
        shard_iters = next_shard_iters


if __name__ == '__main__':
    main()
```

Gestione dell'accesso ai flussi di attività di database

Qualsiasi utente con privilegi del ruolo AWS Identity and Access Management (IAM) appropriati per i flussi di attività di database può creare, avviare, interrompere e modificare le impostazioni del flusso di attività per un'istanza database. Queste operazioni sono incluse nel registro di controllo del flusso. Per le best practice di conformità, consigliamo di non fornire questi privilegi ai DBA.

Imposta l'accesso ai flussi di attività di database utilizzando policy IAM. Per ulteriori informazioni sull'autenticazione di Amazon RDS, consulta Gestione accessi e identità per Amazon RDS. Per ulteriori informazioni sulla creazione di policy IAM, consulta Creazione e utilizzo di una policy IAM per l'accesso al database IAM.

Example Policy per consentire la configurazione dei flussi di attività di database

Per fornire agli utenti l'accesso fine-grained per modificare i flussi di attività, utilizza la chiave di contesto dell'operazione specifica del servizio `rds:StartActivityStream` e `rds:StopActivityStream` in una policy IAM. L'esempio di policy IAM seguente consente a un utente o ruolo di configurare i flussi di attività.

```
{
   "Version":"2012-10-17",
   "Statement":[
        {
            "Sid":"ConfigureActivityStreams",
            "Effect":"Allow",
            "Action": [
                "rds:StartActivityStream",
                "rds:StopActivityStream"
            ],
            "Resource":"*",
          }
     ]
}
```

Example Policy per consentire l'avvio di flussi di attività di database

L'esempio di policy IAM seguente consente a un utente o ruolo di avviare i flussi di attività.

```
{
    "Version":"2012-10-17",
    "Statement":[
        {
            "Sid":"AllowStartActivityStreams",
            "Effect":"Allow",
            "Action":"rds:StartActivityStream",
            "Resource":"*"
        }
    ]
}
```

Example Policy per consentire l'interruzione di flussi di attività di database

L'esempio di policy IAM seguente consente a un utente o ruolo di interrompere i flussi di attività.

```
{
    "Version":"2012-10-17",
    "Statement":[
        {
            "Sid":"AllowStopActivityStreams",
            "Effect":"Allow",
            "Action":"rds:StopActivityStream",
```

```
            "Resource":"*"
        }
     ]
}
```

Example Policy per rifiutare l'avvio di flussi di attività di database

L'esempio di policy IAM seguente consente a un utente o ruolo di rifiutare l'avvio di flussi di attività.

```
{
    "Version":"2012-10-17",
    "Statement":[
        {
            "Sid":"DenyStartActivityStreams",
            "Effect":"Deny",
            "Action":"rds:StartActivityStream",
            "Resource":"*"
        }
     ]
}
```

Example Policy per rifiutare l'interruzione di flussi di attività di database

L'esempio di policy IAM seguente consente a un utente o ruolo di rifiutare l'interruzione di flussi di attività.

```
{
    "Version":"2012-10-17",
    "Statement":[
        {
            "Sid":"DenyStopActivityStreams",
            "Effect":"Deny",
            "Action":"rds:StopActivityStream",
            "Resource":"*"
        }
    ]
}
```

Utilizzo di Amazon RDS Custom

Amazon RDS Custom automatizza le attività e le operazioni di amministrazione del database. RDS Custom consente all'amministratore del database di accedere e personalizzare l'ambiente di database e il sistema operativo. Con RDS Custom, è possibile personalizzare per soddisfare i requisiti delle applicazioni legacy, personalizzate e in pacchetti.

Per i webinar e i blog più recenti su RDS Custom, consulta Amazon RDS Custom resources (Risorse Amazon RDS Custom).

Argomenti

Affrontare la sfida della personalizzazione del database

Amazon RDS Custom porta i vantaggi di Amazon RDS in un mercato che non può facilmente passare a un servizio completamente gestito a causa delle personalizzazioni richieste con applicazioni di terze parti. Amazon RDS Custom consente di risparmiare tempo amministrativo, è duraturo e scalabile con la tua azienda.

Se hai bisogno di gestire completamente l'intero database e il sistema operativo AWS, ti consigliamo Amazon RDS. Se hai bisogno di diritti amministrativi sul database e sul sistema operativo sottostante per rendere disponibili le applicazioni dipendenti, Amazon RDS Custom è la scelta migliore. Se desideri una piena responsabilità di gestione e hai semplicemente bisogno di un servizio di elaborazione gestito, l'opzione migliore è la gestione automatica dei database commerciali su Amazon EC2.

Per offrire un'esperienza di servizio gestito, Amazon RDS non ti consente di accedere all'host sottostante. Amazon RDS limita anche l'accesso ad alcune procedure di sistema e tabelle che richiedono privilegi avanzati. Tuttavia, per alcune applicazioni, potrebbe essere necessario eseguire operazioni come utente del sistema operativo (OS) privilegiato.

Ad esempio, potresti aver bisogno di eseguire alcune delle operazioni seguenti:

- Installa patch e pacchetti personalizzati del database e del sistema operativo.
- Configurare impostazioni specifiche del database.
- Configurare i file system per condividere i file direttamente con le loro applicazioni.

In precedenza, se dovevi personalizzare l'applicazione, dovevi distribuire il database in locale o su Amazon EC2. In questo caso, si assume la maggior parte o tutta la responsabilità per la gestione del database, come riassunto nella tabella seguente.

Funzionalità	Responsabilità locale	Responsabilità di Amazon EC2	Responsabilità di Amazon RDS
Ottimizzazione delle applicazioni	Customer	Customer	Customer
Dimensionamento	Customer	Customer	AWS
Elevata disponibilità	Customer	Customer	AWS
Backup del database	Customer	Customer	AWS
Patching del software del database	Customer	Customer	AWS
Installazione del software del database	Customer	Customer	AWS
Patching del sistema operativo	Customer	Customer	AWS
Installazione del sistema operativo	Customer	Customer	AWS
Manutenzione del server	Customer	AWS	AWS
Ciclo di vita hardware	Customer	AWS	AWS

Funzionalità	Responsabilità locale	Responsabilità di Amazon EC2	Responsabilità di Amazon RDS
Alimentazione, rete e raffreddamento	Customer	AWS	AWS

Quando gestisci autonomamente il software del database, ottieni un maggiore controllo, ma sei anche più incline agli errori dell'utente. Ad esempio, quando si apportano modifiche manualmente, è possibile causare accidentalmente tempi di inattività dell'applicazione. Potresti passare ore a controllare ogni modifica per identificare e risolvere un problema. Idealmente, si desidera un servizio di database gestito che automatizza le attività DBA comuni, ma supporta anche l'accesso privilegiato al database e al sistema operativo sottostante.

Modello di gestione e vantaggi per Amazon RDS Custom

Amazon RDS Custom è un servizio di database gestito per applicazioni legacy, personalizzate e in pacchetti che richiedono l'accesso al sistema operativo e all'ambiente di database sottostanti. RDS Custom automatizza la configurazione, il funzionamento e la scalabilità dei database, garantendo al Cloud AWS contempo l'accesso al database e al sistema operativo sottostante. Con questo accesso, è possibile configurare le impostazioni, installare patch e abilitare le funzionalità native per soddisfare i requisiti dell'applicazione dipendente. Con RDS Custom, è possibile eseguire il carico di lavoro del database utilizzando o il. AWS Management Console AWS CLI

RDS Custom supporta solo i motori di database Oracle Database e Microsoft SQL Server.

Argomenti

- Modello di responsabilità condivisa in RDS Custom
- Perimetro di supporto e configurazioni non supportate in RDS Custom
- Vantaggi principali di RDS Custom

Modello di responsabilità condivisa in RDS Custom

Con RDS Custom, utilizzi le funzionalità gestite di Amazon RDS, ma gestisci l'host e personalizzi il sistema operativo come fai in Amazon EC2. Sei responsabile di altre attività di gestione del database, oltre a ciò che devi fare in Amazon RDS. Il risultato è che hai un maggiore controllo sulla

gestione di database e istanze database rispetto ad Amazon RDS, pur continuando a beneficiare dell'automazione RDS.

Responsabilità condivisa significa quanto segue:

1. Sei proprietario di una parte del processo quando utilizzi una funzionalità di RDS Custom.

   Ad esempio, in RDS Custom per Oracle, puoi controllare quali patch del database Oracle utilizzare e quando applicarle alle tue istanze database.
2. È tua responsabilità assicurarti che tutte le personalizzazioni delle funzionalità RDS Custom funzionino correttamente.

   Per contribuire alla protezione da personalizzazioni non valide, RDS Custom dispone di un software di automazione che viene eseguito all'esterno dell'istanza database. Se l'istanza Amazon EC2 sottostante viene danneggiata, RDS Custom tenta di risolvere questi problemi automaticamente riavviando o sostituendo l'istanza EC2. L'unica modifica visibile all'utente è un nuovo indirizzo IP. Per ulteriori informazioni, consulta Sostituzione dell'host Amazon RDS Custom.

Nella tabella seguente viene illustrato il modello di responsabilità condivisa per le varie funzionalità RDS Custom.

Funzionalità	Responsabilità di Amazon EC2	Responsabilità di Amazon RDS	Responsabilità di RDS Custom per Oracle	Responsabilità di RDS Custom per SQL Server
Ottimizzazione dell'applicazione	Customer	Customer	Customer	Customer
Dimensionamento	Customer	AWS	Condiviso	Condiviso
Elevata disponibilità	Customer	AWS	Customer	AWS
Backup del database	Customer	AWS	Condiviso	AWS

Funzionalità	Responsabilità di Amazon EC2	Responsabilità di Amazon RDS	Responsabilità di RDS Custom per Oracle	Responsabilità di RDS Custom per SQL Server
Patching del software del database	Customer	AWS	Condiviso	AWS per RPEV, Customer per CEV 1
Installazione del software del database	Customer	AWS	Condiviso	AWS per RPEV, cliente per CEV 1
Patching del sistema operativo	Customer	AWS	Customer	AWS per RPEV, cliente per CEV 1
Installazione del sistema operativo	Customer	AWS	Condiviso	AWS
Manutenzione del server	AWS	AWS	AWS	AWS
Ciclo di vita hardware	AWS	AWS	AWS	AWS
Alimentazione, rete e raffredda mento	AWS	AWS	AWS	AWS

[1] Una versione del motore personalizzata (CEV) è un'istantanea di volume binario di una versione del database e di Amazon Machine Image (AMI). Una versione del motore fornita da RDS (RPEV) è l'installazione predefinita di Amazon Machine Image (AMI) e Microsoft SQL Server.

È possibile creare un'istanza DB personalizzata RDS utilizzando Microsoft SQL Server. In questo caso:

- Puoi scegliere tra due modelli di licenza: License Included (LI) e Bring Your Own Media (BYOM).

- Con LI, non è necessario acquistare separatamente le licenze di SQL Server. AWS detiene la licenza per il software di database SQL Server.
- Con BYOM, fornisci e installi i tuoi file binari e le tue licenze di Microsoft SQL Server.

È possibile creare un'istanza DB personalizzata RDS utilizzando Oracle Database. In questo caso, esegui queste operazioni:

- Gestisci i tuoi contenuti multimediali.

  Quando si utilizza RDS Custom, si caricano i file e le patch di installazione del database. È possibile creare una versione del motore personalizzata (CEV) da questi file. Quindi è possibile creare un'istanza DB personalizzata RDS utilizzando questo CEV.
- Gestisci le tue licenze.

  Porti le tue licenze Oracle Database personalizzate e gestisci le licenze da solo.

Perimetro di supporto e configurazioni non supportate in RDS Custom

RDS Custom fornisce una funzionalità di monitoraggio denominata perimetro di supporto. Questa funzionalità garantisce che l'ambiente host e l'ambiente del database siano configurati correttamente. Se apporti una modifica che fa sì che l'istanza database non sia più inclusa dal perimetro di supporto, RDS Custom modifica lo stato dell'istanza in `unsupported-configuration` fino a quando non risolvi manualmente i problemi di configurazione. Per ulteriori informazioni, consulta Perimetro di supporto RDS Custom.

Vantaggi principali di RDS Custom

Con RDS Custom puoi eseguire le seguenti operazioni:

- Automatizza molte delle stesse attività amministrative di Amazon RDS, tra cui:
  - Gestione del ciclo di vita dei database
  - Backup e ripristino automatizzati (PITR) point-in-time
  - Monitoraggio dello stato delle istanze DB RDS Custom e osservazione delle modifiche all'infrastruttura, al sistema operativo e ai processi del database.
  - Notifica o azione per risolvere i problemi a seconda dell'interruzione dell'istanza DB
- Installazione di applicazioni di terze parti.

È possibile installare software per eseguire applicazioni e agenti personalizzati. Poiché si dispone di accesso privilegiato all'host, è possibile modificare i file system per supportare le applicazioni legacy.

- Installa patch personalizzate.

  È possibile applicare patch di database personalizzate o modificare pacchetti del sistema operativo sulle istanze RDS Custom DB.

- Metti in scena un database locale prima di spostarlo in un servizio completamente gestito.

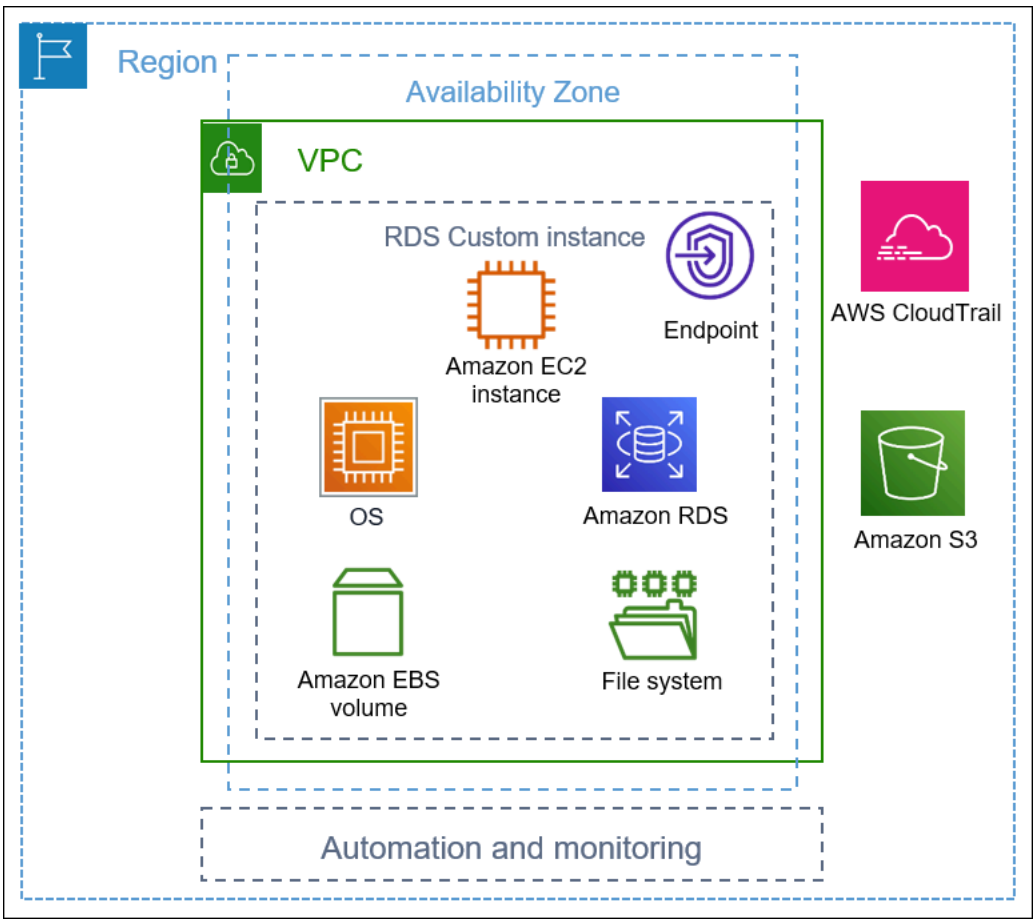
  Se gestisci il tuo database locale, puoi eseguire lo stage del database su RDS Custom così com'è. Dopo aver familiarizzato con l'ambiente cloud, puoi migrare il database in un'istanza database Amazon RDS completamente gestita.

- Creare un'automazione personalizzata.

  È possibile creare, pianificare ed eseguire script di automazione personalizzati per strumenti di reporting, gestione o diagnostica.

Architettura Amazon RDS Custom

L'architettura Amazon RDS Custom è basata su Amazon RDS, con differenze importanti. Il seguente diagramma illustra i componenti principali dell'architettura RDS Custom.



Argomenti

- VPC
- Automazione e monitoraggio RDS Custom
- Amazon S3
- AWS CloudTrail

VPC

Come in Amazon RDS, l'istanza database di RDS Custom risiede in un virtual private cloud (VPC).

L'istanza database RDS Custom è costituita dai seguenti componenti principali:

- Istanza Amazon EC2
- Endpoint dell'istanza
- Sistema operativo installato sull'istanza Amazon EC2
- Storage Amazon EBS, che contiene file system aggiuntivi

Automazione e monitoraggio RDS Custom

RDS Custom dispone di un software di automazione che funziona al di fuori dell'istanza DB. Questo software comunica con gli agenti sull'istanza DB e con altri componenti all'interno dell'ambiente RDS Custom generale.

Le funzioni di monitoraggio e ripristino RDS Custom offrono funzionalità simili a quelle di Amazon RDS. Per impostazione predefinita, RDS Custom è in modalità di automazione completa. Il software di automazione ha le seguenti responsabilità principali:

- Raccogli i parametri e invia notifiche
- Ripristino automatico dell'istanza

Un'importante responsabilità dell'automazione RDS Custom è la risposta ai problemi della tua istanza Amazon EC2. Per vari motivi, l'host potrebbe diventare compromesso o irraggiungibile. RDS Custom risolve questi problemi riavviando o sostituendo l'istanza Amazon EC2.

Argomenti

- Sostituzione dell'host Amazon RDS Custom
- Perimetro di supporto RDS Custom

Sostituzione dell'host Amazon RDS Custom

Se l'host Amazon EC2 viene danneggiato, RDS Custom tenta di riavviarlo. Se questo tentativo non riesce, RDS Custom utilizza la stessa funzione di arresto e avvio inclusa in Amazon EC2. L'unica modifica visibile dal cliente quando un host viene sostituito è un nuovo indirizzo IP pubblico.

Argomenti

- Arresto e avvio dell’host
- Effetti della sostituzione dell'host
- Best practice per gli host Amazon EC2

Arresto e avvio dell’host

RDS Custom adotta automaticamente i seguenti passaggi, senza alcun intervento da parte dell'utente:

1. Arresta l'host Amazon EC2.

   L'istanza EC2 esegue un normale arresto e l'esecuzione si arresta. Tutti i volumi Amazon EBS restano collegati all'istanza e i loro dati vengono conservati. Tutti i dati archiviati nei volumi di archivio istanza (non supportati da RDS Custom) o nella RAM del computer host vengono rimossi.

   Per ulteriori informazioni, consultare Avvio e arresto dell'istanza nella Guida per l'utente di Amazon EC2 per le istanze Linux.
2. Avvia l'host Amazon EC2.

   L'istanza EC2 esegue la migrazione a un nuovo hardware host sottostante. In alcuni casi, l'istanza database RDS Custom rimane sull'host originale.

Effetti della sostituzione dell'host

In RDS Custom, si ha pieno controllo sul volume del dispositivo di root e sui volumi di archiviazione Amazon EBS. Il volume root può contenere dati e configurazioni importanti da non perdere.

RDS Custom for Oracle conserva tutti i dati del database e dei clienti dopo l'operazione, inclusi i dati del volume root. Non sono richiesti interventi da parte dell'utente. In RDS Custom per SQL Server, i dati del database vengono conservati, ma tutti i dati sull'unità C:, inclusi il sistema operativo e i dati del cliente, vengono persi.

Dopo il processo di sostituzione, l'host Amazon EC2 dispone di un nuovo indirizzo IP pubblico. L'host conserva quanto segue:

- ID istanza
- Indirizzi IP privati
- Indirizzi IP elastici
- Metadati delle istanze
- Dati del volume di archiviazione dati
- Dati del volume root (in RDS Custom per Oracle)

Best practice per gli host Amazon EC2

La funzione di sostituzione dell'host Amazon EC2 copre la maggior parte degli scenari dei problemi di Amazon EC2. Consigliamo di seguire queste best practices:

- Prima di modificare la configurazione o il sistema operativo, eseguire il backup dei dati. Se il volume root o il sistema operativo diventano danneggiati, la sostituzione dell'host non è in grado di ripararli. Le uniche opzioni disponibili sono il ripristino da uno snapshot DB o il ripristino point-in-time.
- Non interrompere o terminare manualmente l'host Amazon EC2 fisico. Entrambe le azioni comportano l'inserimento dell'istanza al di fuori del perimetro di supporto RDS Custom.
- (RDS Custom per SQL Server) Se si allegano volumi aggiuntivi all'host Amazon EC2, configurarli in modo che vengano rimontati al riavvio. Se l'host è danneggiato, RDS Custom potrebbe arrestarsi e avviare automaticamente l'host.

Perimetro di supporto RDS Custom

RDS Custom fornisce funzionalità di monitoraggio aggiuntive denominate perimetro di supporto. Questo monitoraggio aggiuntivo assicura che l'istanza RDS Custom utilizzi un'infrastruttura, un sistema operativo e un database AWS supportati.

Il perimetro database verifica che l'istanza database sia conforme ai requisiti elencatiCorrezione delle configurazioni non supportate in RDS Custom per OracleeCorrezione delle configurazioni non supportate in RDS Custom per SQL Server. Se uno di questi requisiti non viene soddisfatto, RDS Custom considera l'istanza database al di fuori del perimetro di supporto.

Argomenti

- Configurazioni non supportate in RDS Custom
- Risoluzione dei problemi relativi alle configurazioni non supportate

Configurazioni non supportate in RDS Custom

Quando l'istanza database è al di fuori del perimetro di supporto, RDS Custom modifica lo stato dell'istanza database in`unsupported-configuration`e invia notifiche sugli eventi. Dopo aver risolto i problemi di configurazione, RDS Custom modifica lo stato dell'istanza database in `available`.

Mentre l'istanza database è nello stato `unsupported-configuration`, il caso è il seguente:

- Il tuo database è raggiungibile. Un'eccezione si verifica se l'istanza database si trova nel`unsupported-configuration`perché il database si chiude in modo imprevisto.
- Non è possibile modificare l'istanza database.
- Non è possibile fare snapshot DB.
- I backup automatici non vengono creati.
- Solo per le istanza database RDS Custom per SQL Server, RDS Custom non sostituisce l'istanza database RDS Custom per SQL Server, se questa viene compromessa. Per altre informazioni sulla sostituzioni dell'host, consulta Sostituzione dell'host Amazon RDS Custom.
- Puoi eliminare la tua istanza database, ma la maggior parte delle altre operazioni API RDS Custom non è disponibile.
- RDS Custom continua a supportare il ripristino point-in-time (PITR) archiviando i file di redo log e caricandoli su Amazon S3. Il PITR con stato `unsupported-configuration` si differenzia nei seguenti modi:
  - Il PITR può richiedere tempi lunghi per il ripristino completo su una nuova istanza database RDS Custom. Questo perché non è possibile acquisire snapshot automatici o manuali mentre lo stato dell'istanza database è `unsupported-configuration`.
  - PITR deve riprodurre più redo log a partire dallo snapshot più recente acquisito prima che l'istanza entrasse nello stato `unsupported-configuration`.

- In alcuni casi, lo stato dell'istanza database è `unsupported-configuration` perché hai apportato una modifica che ha impedito il caricamento dei file di redo log archiviati. Gli esempi includono l'arresto dell'istanza EC2, l'arresto dell'agente RDS Custom e lo scollegamento dei volumi EBS. In questi casi, il PITR non è in grado di ripristinare l'istanza database all'ultima ora ripristinabile.

Risoluzione dei problemi relativi alle configurazioni non supportate

RDS Custom fornisce linee guida per la risoluzione dei problemi relativi allo stato `unsupported-configuration`. Sebbene alcune indicazioni si applichino sia a RDS Custom per Oracle che a RDS Custom per SQL Server, le altre linee guida dipendono dal motore di database. Per informazioni specifiche del motore relative alla risoluzione dei problemi, consulta i seguenti argomenti:

- Correzione delle configurazioni non supportate in RDS Custom per Oracle
- Correzione delle configurazioni non supportate in RDS Custom per SQL Server

Amazon S3

Se utilizzi RDS Custom per Oracle, carichi i supporti di installazione in un bucket Amazon S3 creato dall'utente. RDS Custom for Oracle utilizza il supporto in questo bucket per creare una versione del motore personalizzata (CEV). UNCEVè uno snapshot di un volume binario di una versione di database e di Amazon Machine Image (AMI). Dal CEV, puoi creare un'istanza database RDS Custom. Per ulteriori informazioni, consulta Utilizzo di versioni del motore personalizzate per Amazon RDS Custom per Oracle.

Sia per RDS Custom for Oracle che RDS Custom for SQL Server, RDS Custom crea automaticamente un bucket Amazon S3 con il prefisso della stringa `do-not-delete-rds-custom-`. RDS Custom utilizza il bucket S3 `do-not-delete-rds-custom-` per archiviare i seguenti tipi di file:

- log AWS CloudTrail per il trail creato da RDS Custom
- Artefatti del perimetro di supporto (vedi Perimetro di supporto RDS Custom)
- File di log di ripristino database (solo RDS Custom per Oracle):
- Log delle transazioni (solo RDS Custom per SQL Server)
- Artefatti della versione del motore personalizzata (solo RDS Custom per Oracle)

RDS Custom genera il bucket S3 `do-not-delete-rds-custom-` quando crei una delle seguenti risorse:

- Il primo CEV per RDS Custom for Oracle
- La prima istanza database per RDS Custom for SQL Server

RDS Custom crea un bucket per ciascuna delle seguenti combinazioni:

- ID Account AWS
- Tipo motore (RDS Custom per Oracle o RDS Custom for SQL Server)
- Regione AWS

Ad esempio, se crei RDS Custom for Oracle CEVs in una singola Regione AWS, verrà creato un bucket `do-not-delete-rds-custom-`. Se crei più istanze RDS Custom per SQL Server che risiedono in diverse Regioni AWS, verrà creato un bucket `do-not-delete-rds-custom-` in ciascuna Regione AWS. Se crei un'istanza RDS Custom per Oracle e due istanze RDS Custom per SQL Server in una singola Regione AWS, verranno creati due bucket `do-not-delete-rds-custom-`.

AWS CloudTrail

RDS Custom crea automaticamente un trail AWS CloudTrail il cui nome inizia con `do-not-delete-rds-custom-`. Il perimetro di supporto RDS Custom si basa sugli eventi di CloudTrail per determinare se le azioni influiscono sull'automazione RDS Custom. Per ulteriori informazioni, consulta Risoluzione dei problemi relativi alle configurazioni non supportate.

RDS Custom genera il trail quando crei la prima istanza database. RDS Custom crea un trail per ciascuna delle seguenti combinazioni:

- ID Account AWS
- Tipo motore (RDS Custom per Oracle o RDS Custom for SQL Server)
- Regione AWS

Quando elimini un'istanza database RDS Custom, il CloudTrail per questa istanza non viene rimosso automaticamente. In questo caso, costi per il CloudTrail non eliminato continuano ad essere addebitati al tuo Account AWS. RDS Custom non è responsabile per l'eliminazione di questa risorsa.

Per informazioni su come rimuovere CloudTrail manualmente, consulta Eliminazione di un trail nella Guida per l'utente di AWS CloudTrail.

Sicurezza in Amazon RDS Custom

Acquisisci familiarità con le considerazioni sulla sicurezza di RDS Custom.

Argomenti

Gestione sicura delle attività da parte di RDS Custom per conto dell'utente

RDS Custom utilizza gli strumenti e le tecniche descritti di seguito per eseguire in modo sicuro le operazioni per conto dell'utente:

AWSServiceRoleForRDSCustom ruolo collegato al servizio

Un ruolo collegato al servizio è definito automaticamente dal servizio e include tutte le autorizzazioni richieste dal servizio per chiamare altri Servizi AWS per conto dell'utente. Per RDS Custom, `AWSServiceRoleForRDSCustom` è un ruolo collegato al servizio definito in base al principio del privilegio minimo. RDS Custom utilizza le autorizzazioni in `AmazonRDSCustomServiceRolePolicy`, ovvero la policy associata a questo ruolo, per eseguire la maggior parte delle attività di provisioning e tutte le attività di gestione off-host. Per ulteriori informazioni, consulta AmazonRDS. CustomServiceRolePolicy

Quando esegue attività sull'host, RDS Custom Automation utilizza le credenziali del ruolo collegato al servizio per eseguire comandi utilizzando. AWS Systems ManagerÈ possibile controllare la cronologia dei comandi tramite la cronologia dei comandi di Systems Manager e AWS CloudTrail. Systems Manager si connette all'istanza database RDS Custom utilizzando la configurazione di rete. Per ulteriori informazioni, consulta Fase 4: Configurazione di IAM for RDS Custom per Oracle.

Credenziali IAM temporanee

Durante il provisioning o l'eliminazione delle risorse, RDS Custom a volte utilizza credenziali temporanee derivate dalle credenziali del principale IAM chiamante. Queste credenziali IAM sono limitate dalle policy IAM associate a tale principale e scadono dopo il completamento dell'operazione. Per ulteriori informazioni sulle autorizzazioni richieste per i principali IAM che

utilizzano RDS Custom, consulta Passaggio 5: concedi le autorizzazioni necessarie al tuo utente o ruolo IAM.

Profilo dell'istanza Amazon EC2

Un profilo dell'istanza EC2 è un container per un ruolo IAM che è possibile utilizzare per passare le informazioni sul ruolo a un'istanza EC2. Un'istanza EC2 è alla base di un'istanza DB personalizzata RDS. Fornire un profilo dell'istanza quando viene creata un'istanza database RDS Custom. RDS Custom utilizza le credenziali del profilo dell'istanza EC2 quando esegue attività di gestione basate su host come i backup. Per ulteriori informazioni, consulta Creare manualmente il ruolo IAM e il profilo dell'istanza.

Coppia di chiavi SSH

Quando RDS Custom crea l'istanza EC2 alla base di un'istanza database, crea una coppia di chiavi SSH per conto dell'utente. La chiave utilizza il prefisso di denominazione. `do-not-delete-rds-custom-ssh-privatekey-db-` AWS Secrets Manager memorizza questa chiave privata SSH come segreta nel tuo. Account AWS Amazon RDS non archivia queste credenziali, né vi ha accesso né le utilizza. Per ulteriori informazioni, consulta Coppie di chiavi Amazon EC2 e istanze Linux.

Certificati SSL

Le istanze database RDS personalizzate non supportano i certificati SSL gestiti. Se desideri implementare l'SSL, puoi gestire autonomamente i certificati SSL nel tuo portafoglio e creare un ascoltatore SSL per proteggere le connessioni tra il database client o per la replica del database. Per ulteriori informazioni, consulta Configuring Transport Layer Security Authentication nella documentazione del database Oracle.

Protezione del bucket Amazon S3 dal problema del "confused deputy"

Quando crei una versione del motore personalizzato (CEV) per Amazon RDS Custom per Oracle o un'istanza database RDS Custom per SQL Server, RDS Custom crea un bucket Amazon S3. Il bucket S3 memorizza i file come artefatti CEV, registri di ripristino (transazioni), elementi di configurazione per il perimetro di supporto e registri AWS CloudTrail .

È possibile rendere più sicuri questi bucket S3 utilizzando le chiavi di contesto delle condizioni globali per evitare problemi di tipo confused deputy. Per ulteriori informazioni, consulta Prevenzione del problema "confused deputy" tra servizi.

Il seguente esempio di RDS Custom per Oracle mostra l'uso delle chiavi di contesto delle condizioni globali `aws:SourceArn` e `aws:SourceAccount` in una policy del bucket S3. Per RDS Custom per Oracle, assicurati di includere gli Amazon Resource Names (ARN) per i CEV e le istanze database. Per RDS Custom per SQL Server, assicurati di includere l'ARN per le istanze database.

```
...
{
  "Sid": "AWSRDSCustomForOracleInstancesObjectLevelAccess",
  "Effect": "Allow",
  "Principal": {
     "Service": "custom.rds.amazonaws.com"
  },
  "Action": [
     "s3:GetObject",
     "s3:GetObjectVersion",
     "s3:DeleteObject",
     "s3:DeleteObjectVersion",
     "s3:GetObjectRetention",
     "s3:BypassGovernanceRetention"
  ],
  "Resource": "arn:aws:s3:::do-not-delete-rds-custom-123456789012-us-east-2-c8a6f7/
RDSCustomForOracle/Instances/*",
  "Condition": {
     "ArnLike": {
        "aws:SourceArn": [
            "arn:aws:rds:us-east-2:123456789012:db:*",
            "arn:aws:rds:us-east-2:123456789012:cev:*/*"
        ]
     },
     "StringEquals": {
        "aws:SourceAccount": "123456789012"
    }
  }
},
...
```

Rotazione delle credenziali RDS Custom per Oracle per i programmi di conformità

Alcuni programmi di conformità richiedono la modifica periodica delle credenziali dell'utente del database, ad esempio ogni 90 giorni. RDS Custom per Oracle esegue automaticamente la rotazione delle credenziali per alcuni utenti del database predefiniti.

Argomenti

Rotazione automatica delle credenziali per gli utenti predefiniti

Se l'istanza DB RDS Custom per Oracle è ospitata in Amazon RDS, per i seguenti utenti Oracle predefiniti viene eseguita la rotazione automatica delle credenziali ogni 30 giorni. Le credenziali per gli utenti precedenti risiedono in. AWS Secrets Manager

Utenti Oracle predefiniti

Utente del database	Creato da	Versioni del motore supportate	Note
`SYS`	Oracle	custom-oracle-ee custom-oracle-ee-cdb custom-oracle-se2. custom-oracle-se2 cdb	
`SYSTEM`	Oracle	custom-oracle-ee custom-oracle-ee-cdb custom-oracle-se2 custom-oracle-se2 cdb	
`RDSADMIN`	RDS	custom-oracle-ee	

Utente del database	Creato da	Versioni del motore supportate	Note
		custom-oracle-se2	
`C##RDSADMIN`	RDS	custom-oracle-ee-cdb custom-oracle-se2 cdb	I nomi utente con `C##` prefisso esistono solo nei CDB. Per ulteriori informazioni, consulta Panoramica dell'architettura Amazon RDS Custom per Oracle.
`RDS_DATAGUARD`	RDS	custom-oracle-ee	Questo utente esiste solo nelle repliche di lettura, nei database di origine per le repliche di lettura e nei database sottoposti a migrazione fisica in RDS Custom tramite Oracle Data Guard.
`C##RDS_DATAGUARD`	RDS	custom-oracle-ee-cdb	Questo utente esiste solo nelle repliche di lettura, nei database di origine per le repliche di lettura e nei database sottoposti a migrazione fisica in RDS Custom tramite Oracle Data Guard. I nomi utente con `C##` prefisso esistono solo nei CDB. Per ulteriori informazioni, consulta Panoramica dell'architettura Amazon RDS Custom per Oracle.

Un'eccezione alla rotazione automatica delle credenziali è un'istanza DB RDS Custom per Oracle configurata manualmente come database in standby. RDS esegue la rotazione solo delle credenziali per le repliche di lettura create utilizzando il comando CLI `create-db-instance-read-replica` o l'API `CreateDBInstanceReadReplica`.

Linee guida per la rotazione delle credenziali utente

Per essere sicuro che le credenziali vengano ruotate in base al programma di conformità definito, tieni presente le seguenti linee guida:

- Se per l'istanza DB viene eseguita la rotazione automatica delle credenziali, non modificare o eliminare manualmente un segreto, un file di password o la password per gli utenti elencati nella tabella Utenti Oracle predefiniti. In caso contrario, RDS Custom potrebbe collocare l'istanza DB al di fuori del perimetro di supporto; in questo caso, viene sospesa la rotazione automatica.
- L'utente master RDS non è predefinito e pertanto sei tu il responsabile della modifica manuale della password o dell'impostazione della rotazione automatica in Secrets Manager. Per ulteriori informazioni, consulta Ruotare AWS Secrets Manager i segreti.

Rotazione manuale delle credenziali utente

Per le seguenti categorie di database, RDS non esegue la rotazione automatica delle credenziali per gli utenti elencati nella tabella Utenti Oracle predefiniti:

- Un database configurato manualmente per funzionare come database in standby.
- Database on-premise
- Un'istanza DB esterna al perimetro di supporto o in uno stato in cui l'automazione RDS Custom non può essere eseguita. In questo caso RDS Custom inoltre non esegue la rotazione delle chiavi.

Se il database rientra in una delle categorie precedenti, è necessario eseguire manualmente la rotazione delle credenziali utente.

Per ruotare manualmente le credenziali utente per un'istanza DB

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. In Database, assicurati che al momento RDS non stia eseguendo il backup dell'istanza DB o operazioni come la configurazione della disponibilità elevata.
3. Nella pagina dei dettagli del database, scegli Configurazione e annota l'ID risorsa dell'istanza DB. Oppure puoi usare il AWS CLI comando`describe-db-instances`.
4. Apri la console di Secrets Manager all'indirizzo https://console.aws.amazon.com/secretsmanager/.

5. Nella casella di ricerca inserisci l'ID risorsa dell'istanza DB e cerca il segreto nel modo seguente:

```
do-not-delete-rds-custom-db-resource-id-numeric-string
```

In questo segreto è archiviata la password per RDSADMIN, SYS e SYSTEM. La seguente chiave di esempio fa riferimento all'istanza DB con l'ID risorsa DB db-ABCDEFG12HIJKLNMNOPQRS3TUVWX:

```
do-not-delete-rds-custom-db-ABCDEFG12HIJKLNMNOPQRS3TUVWX-123456
```

> **Important**
>
> Se l'istanza DB è una replica di lettura e utilizza il motore custom-oracle-ee-cdb, esistono due segreti con il suffisso *db-resource-id-numeric-string*, uno per l'utente master e l'altro per RDSADMIN, SYS e SYSTEM. Per trovare il segreto corretto, esegui il seguente comando sull'host:
>
> ```
> cat /opt/aws/rdscustomagent/config/database_metadata.json | python3 -c
>  "import sys,json; print(json.load(sys.stdin)['dbMonitoringUserPassword'])"
> ```
>
> L'attributo dbMonitoringUserPassword indica il segreto per RDSADMIN, SYS eSYSTEM.

6. Se l'istanza DB esiste in una configurazione di Oracle Data Guard, cerca il segreto nel modo seguente:

```
do-not-delete-rds-custom-db-resource-id-numeric-string-dg
```

In questo segreto è archiviata la password per RDS_DATAGUARD. La seguente chiave di esempio fa riferimento all'istanza DB con l'ID risorsa DB db-ABCDEFG12HIJKLNMNOPQRS3TUVWX:

```
do-not-delete-rds-custom-db-ABCDEFG12HIJKLNMNOPQRS3TUVWX-789012-dg
```

7. Per tutti gli utenti del database elencati in Utenti Oracle predefiniti, aggiorna le password seguendo le istruzioni riportate in Modificare un AWS Secrets Manager segreto.
8. Se il database è un database autonomo o un database di origine in una configurazione di Oracle Data Guard:

a. Avvia il client Oracle SQL e accedi come SYS.

b. Esegui un'istruzione SQL nel seguente modo per ogni utente del database elencato nella tabella Utenti Oracle predefiniti:

```
ALTER USER user-name IDENTIFIED BY pwd-from-secrets-manager ACCOUNT UNLOCK;
```

Ad esempio, se la nuova password per RDSADMIN archiviata in Secrets Manager èpwd-123, esegui la seguente istruzione:

```
ALTER USER RDSADMIN IDENTIFIED BY pwd-123 ACCOUNT UNLOCK;
```

9. Se l'istanza DB è eseguita in Oracle Database 12c Release 1 (12.1) ed è gestita da Oracle Data Guard, copia manualmente il file di password (orapw) dall'istanza DB primaria in ciascuna istanza DB in standby.

Se l'istanza DB è ospitata in Amazon RDS, la posizione del file di password è /rdsdbdata/config/orapw. Per i database non ospitati in Amazon RDS, la posizione predefinita è $ORACLE_HOME/dbs/orapw$ORACLE_SID su Linux e UNIX e %ORACLE_HOME%\database\PWD%ORACLE_SID%.ora su Windows.

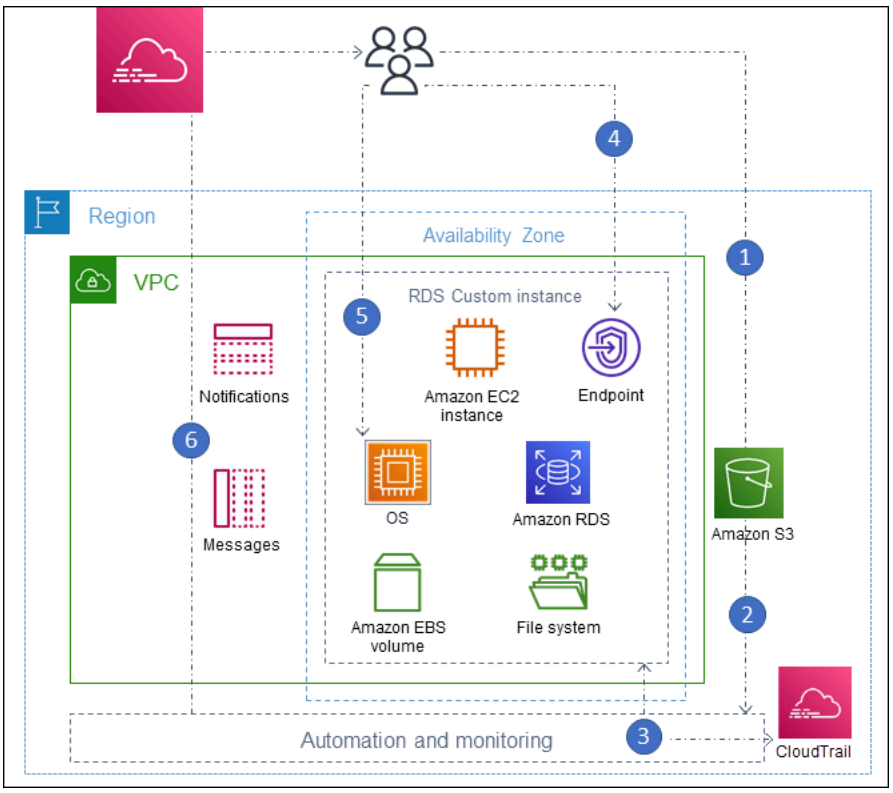
Utilizzo di CEV per RDS Custom for Oracle

Di seguito puoi trovare le istruzioni per creare, gestire e mantenere le tue istanze database RDS Custom for Oracle.

Argomenti

Flusso di lavoro RDS Custom per Oracle

Il seguente diagramma mostra il flusso di lavoro tipico di RDS Custom for Oracle.



I passaggi sono i seguenti:

1. Caricare il software del database nel bucket Amazon S3.

   Per ulteriori informazioni, consulta Fase 3: caricamento dei file di installazione in Amazon S3.

2. Crea una versione del motore personalizzato (CEV) RDS Custom per Oracle dal tuo supporto.

   Scegli l'architettura CDB o l'architettura tradizionale non CDB. Per ulteriori informazioni, consulta Creazione di un CEV.

3. Crea un'istanza DB RDS Custom per Oracle da un motore personalizzato (CEV).

   Per ulteriori informazioni, consulta Creazione di un'istanza database RDS Custom per Oracle.

4. Connetti l'applicazione all'endpoint dell'istanza DB.

   Per ulteriori informazioni, consulta Connessione all'istanza database RDS Custom tramite SSH e Connessione all'istanza database RDS Custom utilizzando Session Manager.

5. (Facoltativo) Accedi all'host per personalizzare il software.

6. Monitora le notifiche e i messaggi generati dall'automazione RDS Custom.

File di installazione del database

La tua responsabilità per i media è una differenza fondamentale tra Amazon RDS e RDS Custom. Amazon RDS, che è un servizio completamente gestito, fornisce Amazon Machine Image (AMI) e software di database. Il software di database Amazon RDS è preinstallato, quindi è necessario scegliere solo un motore di database e una versione e creare il database.

Per RDS Custom, fornisci i tuoi supporti. Quando crei una versione del motore personalizzata, RDS Custom installa il supporto fornito. Il supporto RDS Custom contiene i file e le patch di installazione del database. Questo modello di servizio è chiamato Porta i tuoi media (BYOM).

Versioni del motore personalizzate per RDS Custom per Oracle

Una versione del motore personalizzato (CEV, Custom Engine Version) RDS Custom per Oracle è uno snapshot del volume binario di una versione del database e dell'AMI. Per impostazione predefinita, RDS Custom per Oracle utilizza l'AMI più recente fornita da Amazon EC2. Puoi anche scegliere di riutilizzare un'AMI esistente.

Manifesto CEV

Dopo aver scaricato i file di installazione del database Oracle da Oracle, carichi tali file in un bucket Amazon S3. Quando crei la CEV, specifichi i nomi di file in un documento JSON denominato Manifesto CEV. RDS Custom per Oracle utilizza i file specificati e l'AMI per creare la CEV.

RDS Custom per Oracle fornisce modelli di manifesto JSON con i file .zip consigliati per ogni versione supportata del database Oracle. Ad esempio, il seguente modello è per la RU 19.17.0.0.0.

```
{
    "mediaImportTemplateVersion": "2020-08-14",
    "databaseInstallationFileNames": [
        "V982063-01.zip"
    ],
    "opatchFileNames": [
        "p6880880_190000_Linux-x86-64.zip"
    ],
    "psuRuPatchFileNames": [
        "p34419443_190000_Linux-x86-64.zip",
        "p34411846_190000_Linux-x86-64.zip"
    ],
    "otherPatchFileNames": [
        "p28852325_190000_Linux-x86-64.zip",
```

```
        "p29997937_190000_Linux-x86-64.zip",
        "p31335037_190000_Linux-x86-64.zip",
        "p32327201_190000_Linux-x86-64.zip",
        "p33613829_190000_Linux-x86-64.zip",
        "p34006614_190000_Linux-x86-64.zip",
        "p34533061_190000_Linux-x86-64.zip",
        "p34533150_190000_Generic.zip",
        "p28730253_190000_Linux-x86-64.zip",
        "p29213893_1917000DBRU_Generic.zip",
        "p33125873_1917000DBRU_Linux-x86-64.zip",
        "p34446152_1917000DBRU_Linux-x86-64.zip"
    ]
}
```

Puoi anche specificare i parametri di installazione nel manifesto JSON. Ad esempio, puoi impostare valori non predefiniti per la base Oracle, la home Oracle e l'ID e il nome dell'utente e del gruppo UNIX/Linux. Per ulteriori informazioni, consulta Campi JSON nel manifesto CEV.

Formato di denominazione della CEV

Assegna un nome alla CEV RDS Custom per Oracle usando una stringa specificata dal cliente. A seconda della versione di Oracle Database, il formato del nome è il seguente:

- `19.customized_string`
- `18.customized_string`
- `12.2.customized_string`
- `12.1.customized_string`

Il nome utente può contenere solo 1–50 caratteri alfanumerici, punti e trattini (-, _). È ad esempio possibile denominare il proprio ruolo `19.my_cev1`.

Architettura multitenant Oracle in RDS Custom per Oracle

L'architettura multitenant consente a un database Oracle di funzionare come database container (CDB). Un CDB può includere zero, uno o più database collegabili creati dal cliente (PDB). Un PDB è una raccolta portatile di schemi e oggetti visualizzata in un'applicazione come un tradizionale database non CDB. A partire da Oracle Database 21c, tutti i database Oracle sono CDB.

Quando crei una CEV RDS Custom per Oracle, è necessario specificare l'architettura CDB o non CDB. È possibile creare un CDB RDS Custom per Oracle CDB solo quando la CEV utilizzata per

crearlo utilizza l'architettura multitenant Oracle. Per ulteriori informazioni, consulta Utilizzo di versioni del motore personalizzate per Amazon RDS Custom per Oracle.

Creazione di un'istanza database RDS Custom per Oracle

Una volta creata, la CEV è disponibile per l'uso. È possibile creare più CEV e creare più istanze database RDS Custom per Oracle da qualsiasi CEV. È inoltre possibile modificare lo stato di un CEV per renderlo disponibile o inattivo.

Puoi creare la tua istanza DB RDS Custom for Oracle con l'architettura multitenant Oracle (`custom-oracle-ee-cdb`o il tipo di `custom-oracle-se2-cdb` motore) o con la tradizionale architettura non CDB (o tipo di motore). `custom-oracle-ee` `custom-oracle-se2` Quando crei un database container (CDB), include un database collegabile (PDB) e un seed PDB. È possibile creare manualmente altri PDB utilizzando Oracle SQL.

Per creare l'istanza RDS Custom for Oracle DB, utilizzare il comando `create-db-instance`. In questo comando, specificare quale CEV utilizzare. La procedura è simile a quella per la creazione di un'istanza database Amazon RDS. Tuttavia, alcuni parametri sono diversi. Per ulteriori informazioni, consulta Configurazione di un'istanza database per Amazon RDS Custom per Oracle.

Connessioni database

Come in Amazon RDS, l'istanza database personalizzato RDS Custom risiede in un cloud privato virtuale (VPC). L'applicazione si connette al database Oracle utilizzando un ascoltatore Oracle.

Se il database è un CDB, puoi utilizzare l'ascoltatore `L_RDSCDB_001` per connetterti alla root CDB e a un PDB. Se colleghi un non CDB a un CDB, assicurati di impostare `USE_SID_AS_SERVICE_LISTENER = ON` in modo che le applicazioni migrate mantengano le stesse impostazioni.

Quando ti connetti a un non CDB, l'utente master è l'utente del non CDB. Quando ti connetti a un CDB, l'utente master è l'utente del PDB. Per connetterti alla root CDB, accedi all'host, avvia un client SQL e crea un utente amministrativo con i comandi SQL.

Personalizzazione RDS Personalizza

È possibile accedere all'host RDS Custom per installare o personalizzare il software. Per evitare conflitti tra le modifiche e l'automazione personalizzata di RDS, è possibile sospendere l'automazione per un periodo specificato. Durante questo periodo, RDS Custom non esegue il monitoraggio o il

ripristino dell'istanza. Al termine del periodo, RDS Custom riprende l'automazione completa. Per ulteriori informazioni, consulta Sospensione e ripresa dell'istanza database RDS Custom.

Architettura dei database per Amazon RDS Custom per Oracle

RDS Custom per Oracle supporta sia l'architettura multitenant Oracle che quella non multitenant.

Argomenti

Architetture di database Oracle supportate

L'architettura multitenant Oracle, chiamata anche architettura CDB, consente a un database Oracle di funzionare come database container (CDB). Un CDB include database collegabili (PDB). Un PDB è una raccolta di schemi e oggetti visualizzata in un'applicazione come un tradizionale database Oracle. Per ulteriori informazioni, consulta l'introduzione all'architettura multilocazione nella Guida per l'amministratore di Oracle Multitenant.

Le architetture CDB e non CDB si escludono a vicenda. Se un database Oracle non è un CDB, è un database non CDB e quindi non può contenere PDB. In RDS Custom per Oracle, solo Oracle Database 19c supporta l'architettura CDB. Pertanto, se crei istanze database utilizzando versioni precedenti del database Oracle, puoi creare solo istanze non CDB. Per ulteriori informazioni, consulta Considerazioni sull'architettura multilocazione.

Tipi di motore supportati

Quando crei un'istanza Amazon RDS Custom per Oracle CEV o DB, scegli un tipo di motore CDB o un tipo di motore non CDB:

- `custom-oracle-ee-cdb` e `custom-oracle-se2-cdb`

  Questi tipi di motore specificano l'architettura multitenant Oracle. Questa opzione è disponibile solo per Oracle Database 19c. Quando crei un'istanza CDB RDS per Oracle utilizzando l'architettura multilocazione, il CDB include i seguenti container:

  - Root CBD (`CDB$ROOT`)
  - Seed PDB (`PDB$SEED`)
  - PDB iniziale

Puoi creare più PDB utilizzando il comando Oracle SQL `CREATE PLUGGABLE DATABASE`. Non puoi utilizzare le API RDS per creare o eliminare i PDB.

- `custom-oracle-ee` e `custom-oracle-se2`

  Questi tipi di motore specificano l'architettura tradizionale non CDB. Un non CDB non può contenere database collegabili (PDB).

Per ulteriori informazioni, consulta Considerazioni sull'architettura multilocazione.

Funzionalità supportate nell'architettura multitenant Oracle

Un'istanza CDB RDS Custom per Oracle supporta le seguenti funzionalità:

- Backup
- Ripristino e point-time-restore (PITR) dai backup
- Repliche di lettura
- Aggiornamenti della versione secondaria

Disponibilità e supporto delle funzionalità per RDS Custom for Oracle

In questo argomento, è possibile trovare un riepilogo della disponibilità e del supporto delle funzionalità di RDS Custom for Oracle per una rapida consultazione.

Argomenti

Regione AWS e supporto della versione del database per RDS Custom for Oracle

Il supporto varia a seconda delle versioni specifiche di ciascun motore di database e a seconda delle Regioni AWS. Per ulteriori informazioni sulla disponibilità di versioni e regioni di RDS Custom per Oracle, consulta Regioni e motori DB supportati per RDS Custom.

Supporto della versione del database per RDS Custom for Oracle

RDS Custom for Oracle supporta le seguenti versioni del database Oracle:

- Oracle Database 19c
- Oracle Database 18c
- Oracle Database 12c Release 2 (12.2)
- Oracle Database 12c Release 1 (12.1)

Supporto di edizioni e licenze per RDS Custom per Oracle

RDS Custom for Oracle supporta Enterprise Edition (EE) e Standard Edition 2 (SE2) sul modello BYOL.

Nota le seguenti limitazioni per Standard Edition 2:

- Oracle Data Guard non è supportato. Pertanto, non è possibile creare repliche Oracle Read.

- È possibile utilizzare solo classi di istanze DB con 16 o meno vCPU (fino a 4xlarge).
- Un'istanza CDB su Standard Edition 2 supporta un massimo di 3 database tenant.
- Non è possibile migrare i dati tra Enterprise Edition e Standard Edition 2.

Supporto delle classi di istanza database per RDS Custom per Oracle

RDS Custom per Oracle supporta le classi di istanza database indicate di seguito. Se crei un'istanza DB su Standard Edition 2, puoi utilizzare solo classi di istanze con 16 o meno vCPU (fino a 4 volte più grandi).

Type	Size
db.r6i	db.r6i.large |db.r6i.xlarge |db.r6i.2xlarge |db.r6i.4xlarge |db.r6i.8xlarge |db.r6i.12xlarge |db.r6i.16xlarge |db.r6i.24xlarge |db.r6i.32xlarge
db.r5b	db.r5b.large |db.r5b.xlarge |db.r5b.2xlarge |db.r5b.4xlarge |db.r5b.8xlarge |db.r5b.12xlarge |db.r5b.16xlarge |db.r5b.24xlarge
db.r5	db.r5.large |db.r5.xlarge |db.r5.2xlarge |db.r5.4xlarge | db.r5.8xlarge |db.r5.12xlarge |db.r5.16xlarge |db.r5.24xlarge
db.x2iec	db.x2iedn.xlarge |db.x2iedn.2xlarge |db.x2iedn.4xlarge | db.x2iedn.8xlarge |db.x2iedn.16xlarge |db.x2iedn.24xlarge | db.x2iedn.32xlarge
db.x2iez	db.x2iezn.2xlarge |db.x2iezn.4xlarge |db.x2iezn.6xlarge | db.x2iezn.8xlarge |db.x2iezn.12xlarge
db.m6i	db.m6i.large |db.m6i.xlarge |db.m6i.2xlarge |db.m6i.4xlarge |db.m6i.8xlarge |db.m6i.12xlarge |db.m6i.16xlarge |db.m6i.24xlarge |db.m6i.32xlarge
db.m5	db.m5.large |db.m5.xlarge |db.m5.2xlarge |db.m5.4xlarge | db.m5.8xlarge |db.m5.12xlarge |db.m5.16xlarge |db.m5.24xlarge
db.t3	db.t3.medium |db.t3.large |db.t3.xlarge |db.t3.2xlarge

Supporto per gruppi di opzioni per RDS Custom for Oracle

È possibile specificare un gruppo di opzioni quando si crea o si modifica un'istanza DB RDS Custom for Oracle. Per ulteriori informazioni, consulta Utilizzo dei gruppi di opzioni in RDS Custom for Oracle.

Requisiti e limitazioni di RDS Custom for Oracle

In questo argomento è riportato un riepilogo della disponibilità e dei requisiti di Amazon RDS Custom per Oracle per una rapida consultazione.

Argomenti

Requisiti generali per RDS Custom per Oracle

Assicurati di soddisfare i seguenti requisiti per Amazon RDS Custom for Oracle:

- È possibile accedere a My Oracle Support e Oracle Software Delivery Cloud per scaricare l'elenco dei file di installazione e delle patch supportati per RDS Custom for Oracle. Se si utilizza una patch sconosciuta, la creazione della versione del motore personalizzato (CEV) non riesce. In questo caso, contatta il team di supporto RDS Custom e chiedi di aggiungere la patch mancante. Per ulteriori informazioni, consulta Fase 2: download di file e patch di installazione del database da Oracle Software Delivery Cloud.
- Si dispone dell'accesso ad Amazon S3. È necessario questo servizio per i seguenti motivi:
  - I file di installazione Oracle vengono caricati nei bucket S3. Utilizzi i file di installazione caricati per creare la tua CEV RDS Custom.
  - RDS Custom per Oracle utilizza gli script scaricati dai bucket S3 definiti internamente per eseguire operazioni sulle istanze DB. Questi script sono necessari per l'onboarding e l'automazione di RDS Custom.
  - RDS Custom per Oracle carica determinati file nei bucket S3 presenti nell'account del cliente. Questi bucket utilizzano il seguente formato di denominazione: `do-not-delete-rds-custom-`*`account_id-region-six_character_alphanumeric_string`*. Ad esempio, potresti avere un bucket denominato `do-not-delete-rds-custom-123456789012-us-east-1-12a3b4`.

  Per ulteriori informazioni, consulta Fase 3: caricamento dei file di installazione in Amazon S3 e Creazione di un CEV.

- Si utilizzano le classi di istanze DB elencate in Supporto delle classi di istanza database per RDS Custom per Oracle per creare le istanze DB RDS Custom for Oracle.
- Le tue istanze DB RDS Custom for Oracle eseguono Oracle Linux 7 Update 9 o versioni successive.
- Devi specificare le unità a stato solido gp2, gp3 o io1 per lo storage Amazon EBS. La dimensione massima di archiviazione è di 64 TiB.
- È disponibile una AWS KMS chiave per creare un'istanza RDS Custom for Oracle DB. Per ulteriori informazioni, consulta Fase 1: creazione o riutilizzo di una chiave AWS KMS di crittografia simmetrica.
- Hai il ruolo AWS Identity and Access Management (IAM) e il profilo di istanza necessari per creare istanze RDS Custom for Oracle DB. Per ulteriori informazioni, consulta Fase 4: Configurazione di IAM for RDS Custom per Oracle.
- L'utente AWS Identity and Access Management (IAM) che crea un'istanza DB personalizzata CEV o RDS dispone delle autorizzazioni necessarie per IAM e Amazon CloudTrail S3.

  Per ulteriori informazioni, consulta Passaggio 5: concedi le autorizzazioni necessarie al tuo utente o ruolo IAM.
- Fornisci il cloud privato virtuale (VPC) e la configurazione dei gruppi di sicurezza. Per ulteriori informazioni, consulta Fase 6: Configurazione del VPC per RDS Custom for Oracle.
- Fornisci una configurazione di rete che RDS Custom for Oracle può utilizzare per accedere ad altre. Servizi AWS Per requisiti specifici, consulta Fase 4: Configurazione di IAM for RDS Custom per Oracle.

Limitazioni generali di RDS Custom per Oracle

Le seguenti limitazioni si applicano a RDS Custom for Oracle:

- Non è possibile modificare l'identificatore dell'istanza database di un'istanza database RDS Custom for Oracle esistente.
- È possibile specificare l'architettura multitenant Oracle solo per Oracle Database 19c.
- Non è possibile creare più database Oracle in un'unica istanza database RDS Custom per Oracle.
- Non puoi interrompere un'istanza database RDS Custom per Oracle o l'istanza Amazon EC2 sottostante. La fatturazione per un'istanza database RDS Custom per Oracle non può essere interrotta.

- Non è possibile utilizzare la gestione automatica della memoria condivisa perché RDS Custom for Oracle supporta solo la gestione automatica della memoria. Per ulteriori informazioni, consulta l'argomento relativo alla gestione automatica della memoria nel manuale Oracle Database Administrator's Guide.
- Assicurati di non modificare `DB_UNIQUE_NAME` per un'istanza database primaria. La modifica del nome causa il blocco di qualsiasi operazione di ripristino.

Per le limitazioni specifiche della modifica di un'istanza database RDS Custom per Oracle, consulta Modifica dell'istanza database RDS Custom per Oracle. Per le limitazioni della replica, consulta Limitazioni generali per la replica RDS Custom per Oracle.

Limitazioni CEV e AMI per RDS Custom for Oracle

Le seguenti limitazioni si applicano a RDS Custom per Oracle CEV e AMI:

- Non puoi fornire la tua AMI da utilizzare in un RDS Custom per Oracle CEV. È possibile specificare l'AMI predefinito o un AMI utilizzato in precedenza da un RDS Custom per Oracle CEV.

> **Note**
>
> RDS Custom for Oracle rilascia una nuova AMI predefinita quando vengono scoperte vulnerabilità ed esposizioni comuni. Non è disponibile o garantito alcun programma fisso. RDS Custom for Oracle tende a pubblicare una nuova AMI predefinita ogni 30 giorni.

- Non è possibile modificare una CEV per utilizzare una AMI diversa.
- Non è possibile creare un'istanza CDB da un CEV che utilizza i tipi di motore or. `custom-oracle-ee` `custom-oracle-se2` Il CEV deve utilizzare o. `custom-oracle-ee-cdb` `custom-oracle-se2-cdb`
- RDS Custom for Oracle attualmente non consente di aggiornare il sistema operativo dell'istanza DB RDS Custom for Oracle con chiamate API RDS. Come soluzione alternativa, puoi aggiornare il tuo sistema operativo manualmente con il seguente comando:. `sudo yum update --security`

Impostazioni non supportate per creare e modificare flussi di lavoro

Quando crei o modifichi un'istanza DB RDS Custom for Oracle, non puoi fare quanto segue:

- Modificare il numero di thread per core e di core CPU sulla classe di istanza database.

- Attivare il calcolo automatico dello storage.
- Creazione di una implementazione Multi-AZ.

> **Note**
>
> Per una soluzione HA alternativa, consulta l'articolo del AWS blog Crea alta disponibilità per Amazon RDS Custom for Oracle usando repliche di lettura.

- Impostazione della conservazione del backup su `0`.
- Configurazione dell'autenticazione Kerberos.
- Specifica del gruppo di parametri database o del gruppo di opzioni.
- Attivare Performance Insights.
- Attivazione degli aggiornamenti a versioni secondarie automatiche.

Quote di istanze DB per Account AWS

Assicurati che il numero combinato di istanze RDS Custom e Amazon RDS DB non superi il limite di quota. Ad esempio, se la quota per Amazon RDS è di 40 istanze DB, puoi avere 20 istanze RDS personalizzate per Oracle DB e 20 istanze Amazon RDS DB.

Configurazione dell'ambiente per Amazon RDS Custom per Oracle

Prima di creare un'istanza database Amazon RDS Custom per Oracle, esegui le seguenti attività.

Argomenti

Fase 1: creazione o riutilizzo di una chiave AWS KMS di crittografia simmetrica

Le chiavi gestite dai clienti si trovano AWS KMS keys nel tuo AWS account che crei, possiedi e gestisci. Per RDS Custom è necessaria una chiave KMS di crittografia simmetrica gestita dal cliente. Quando crei un'istanza database RDS Custom for Oracle, è necessario fornire l'identificatore KMS della chiave. Per ulteriori informazioni, consulta Configurazione di un'istanza database per Amazon RDS Custom per Oracle.

Sono disponibili le seguenti opzioni:

- Se disponi già di una chiave KMS gestita dal cliente Account AWS, puoi utilizzarla con RDS Custom. Non è richiesta alcuna operazione aggiuntiva.
- Se hai già creato una chiave KMS di crittografia simmetrica gestita dal cliente per un motore RDS Custom diverso, puoi riutilizzare la stessa chiave KMS. Non è richiesta alcuna operazione aggiuntiva.
- Se non disponi di una chiave KMS di crittografia simmetrica gestita dal cliente esistente nel tuo account, crea una chiave KMS seguendo le istruzioni in Creazione di chiavi nella Guida per gli sviluppatori di AWS Key Management Service .
- Se stai creando un'istanza DB personalizzata CEV o RDS e la tua chiave KMS si trova in un'altra Account AWS, assicurati di utilizzare la. AWS CLI Non puoi utilizzare la AWS console con chiavi KMS per più account.

> **Important**
>
> RDS Custom non supporta le chiavi KMS AWS gestite.

Assicurati che la tua chiave di crittografia simmetrica conceda l'accesso al ruolo `kms:Decrypt` and `kms:GenerateDataKey` operations to the AWS Identity and Access Management (IAM) nel profilo dell'istanza IAM. Se hai una nuova chiave di crittografia simmetrica nel tuo account, non sono necessarie modifiche. Altrimenti, assicurati che la policy della chiave di crittografia simmetrica fornisca l'accesso a queste operazioni.

Per ulteriori informazioni, consulta Fase 4: Configurazione di IAM for RDS Custom per Oracle.

Per ulteriori informazioni sulla configurazione di IAM per RDS Custom per Oracle, consulta Fase 4: Configurazione di IAM for RDS Custom per Oracle.

Passaggio 2: scarica e installa il AWS CLI

AWS fornisce un'interfaccia a riga di comando per utilizzare le funzionalità RDS Custom. È possibile utilizzare la versione 1 o la versione 2 di AWS CLI.

Per informazioni sul download e l'installazione di AWS CLI, vedere Installazione o aggiornamento della versione più recente di. AWS CLI

Ignora questo passaggio se si verifica una delle seguenti condizioni:

- Si prevede di accedere a RDS Custom solo da. AWS Management Console
- Hai già scaricato AWS CLI per Amazon RDS o un altro motore RDS Custom DB.

Fase 3: Estrarre i CloudFormation modelli per RDS Custom for Oracle

Per semplificare la configurazione, si consiglia vivamente di utilizzare i AWS CloudFormation modelli per creare CloudFormation pile. Se prevedi di configurare IAM e il tuo VPC manualmente, salta questo passaggio.

Argomenti

- Passaggio 3a: scarica i file del modello CloudFormation
- Passaggio 3b: Estrarre .json custom-oracle-iam

- Fase 3c: Estrarre custom-vpc.json

Passaggio 3a: scarica i file del modello CloudFormation

Un CloudFormation modello è una dichiarazione delle AWS risorse che compongono uno stack. Il modello viene archiviato come un file JSON.

Per scaricare i file CloudFormation modello

1. Apri il menu contestuale (fai clic con il pulsante destro del mouse) per il link custom-oracle-iam.zip e scegli Salva collegamento con nome.
2. Salvare il file sul computer.
3. Ripeti i passaggi precedenti per il collegamento custom-vpc.json.

   Se si ha già configurato il VPC per RDS Custom, questo passaggio può essere ignorato.

Passaggio 3b: Estrarre .json custom-oracle-iam

Apri il `custom-oracle-iam.zip` file che hai scaricato, quindi estrai il file. `custom-oracle-iam.json` L'inizio del file è simile al seguente.

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Parameters": {
    "EncryptionKey": {
      "Type": "String",
      "Default": "*",
      "Description": "KMS Key ARN for encryption of data managed by RDS Custom and by
 DB Instances."
    }
  },
  "Resources": {
    "RDSCustomInstanceServiceRole": {
      "Type": "AWS::IAM::Role",
      "Properties": {
        "RoleName": { "Fn::Sub": "AWSRDSCustomInstanceRole-${AWS::Region}" },
        "AssumeRolePolicyDocument": {
          "Version": "2012-10-17",
          "Statement": [
            {
              "Action": "sts:AssumeRole",
```

```
            "Effect": "Allow",
            "Principal": {
              "Service": "ec2.amazonaws.com"
            }
          }
        ]
      },...
```

Fase 3c: Estrarre custom-vpc.json

> **Note**
>
> Se hai già configurato un VPC esistente per RDS Custom for Oracle, salta questo passaggio. Per ulteriori informazioni, consulta Configurazione manuale del VPC per RDS Custom for Oracle.

Apri il `custom-vpc.zip` file che hai scaricato, quindi estrai il file. `custom-vpc.json` L'inizio del file è simile al seguente.

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Parameters": {
    "PrivateVpc": {
      "Type": "AWS::EC2::VPC::Id",
      "Description": "Private VPC Id to use for RDS Custom DB Instances"
    },
    "PrivateSubnets": {
      "Type": "List<AWS::EC2::Subnet::Id>",
      "Description": "Private Subnets to use for RDS Custom DB Instances"
    },
    "RouteTable": {
      "Type": "String",
      "Description": "Route Table that must be associated with the PrivateSubnets and
 used by S3 VPC Endpoint",
      "AllowedPattern": "rtb-[0-9a-z]+"
    }
  },
  "Resources": {
    "DBSubnetGroup": {
      "Type": "AWS::RDS::DBSubnetGroup",
      "Properties": {
```

```
        "DBSubnetGroupName": "rds-custom-private",
        "DBSubnetGroupDescription": "RDS Custom Private Network",
        "SubnetIds": {
          "Ref": "PrivateSubnets"
        }
      }
    },...
```

Fase 4: Configurazione di IAM for RDS Custom per Oracle

Utilizzi un ruolo IAM o un utente IAM (conosciuto come entità IAM) per creare un'istanza database RDS Custom tramite la console o la AWS CLI. Questa entità IAM deve disporre delle autorizzazioni necessarie per la creazione dell'istanza.

Puoi configurare IAM utilizzando una delle due procedure CloudFormation o quelle manuali.

> **Important**
>
> Ti consigliamo vivamente di configurare l'ambiente RDS Custom for Oracle utilizzando AWS CloudFormation. Questa tecnica è la più semplice e meno soggetta a errori.

Argomenti

- Configura IAM utilizzando CloudFormation
- Creare manualmente il ruolo IAM e il profilo dell'istanza

Configura IAM utilizzando CloudFormation

Quando utilizzi il CloudFormation modello per IAM, crea le seguenti risorse richieste:

- Un profilo di istanza denominato `AWSRDSCustomInstanceProfile-`*`region`*
- Un ruolo di servizio denominato `AWSRDSCustomInstanceRole-`*`region`*
- Una politica di accesso `AWSRDSCustomIamRolePolicy` denominata associata al ruolo di servizio

Per configurare IAM utilizzando CloudFormation

1. Apri la CloudFormation console all'indirizzo https://console.aws.amazon.com/cloudformation.
2. Avviare la procedura guidata Crea stack e scegliere Create Stack (Crea stack).

3. Nella pagina Create stack (Crea stack), esegui le operazioni seguenti:

   a. In Prepare template (Prepara modello) scegli Template is ready (Il modello è pronto).

   b. Come Template source (Origine modello), scegliere Upload a template file (Carica un file di modello).

   c. Per Scegli file, vai a, quindi scegli custom-oracle-iam.json.

   d. Seleziona Successivo.

4. Nella pagina Specify stack details (Specifica dettagli), procedere come segue:

   a. In Nome stack, immetti **`custom-oracle-iam`**.

   b. Seleziona Successivo.

5. Nella pagina Configure stack options (Configura opzioni pila), scegliere Next (Successivo).

6. Nella custom-oracle-iam pagina Revisione, procedi come segue:

   a. Seleziona la casella di spunta I acknowledge that AWS CloudFormation might create IAM resources with custom names (Sono consapevole che AWS CloudFormation potrebbe creare risorse IAM con nomi personalizzati).

   b. Scegli Invia.

   CloudFormation crea i ruoli IAM richiesti da RDS Custom for Oracle. Nel pannello di sinistra, quando viene custom-oracle-iamvisualizzato CREATE_COMPLETE, procedi al passaggio successivo.

7. Nel pannello di sinistra, scegliete. custom-oracle-iam Nel riquadro d i destra esegui queste operazioni:

   a. Scegli Informazioni stack. *`Il tuo stack ha un ID nel formato`* ***`arn:aws:cloudformation: region: account-no:stack/`****`/identifier. custom-oracle-iam`*

   b. Scegliere Resources (Risorse). Verrà visualizzato un codice analogo al seguente:

      - Un AWSRDSCustomInstanceProfileprofilo ***`di`*** istanza denominato - region
      - Un ruolo di servizio denominato AWSRDSCustomInstanceRole- ***`region`***

      Quando viene creata l'istanza database RDS Custom, è necessario fornire l'ID del profilo dell'istanza.

Creare manualmente il ruolo IAM e il profilo dell'istanza

La configurazione è più semplice quando si utilizza CloudFormation. Tuttavia, è possibile configurare IAM anche manualmente. Per la configurazione manuale, procedi come segue:

- Fase 1: creazione di un ruolo IAM per AWSRDSCustomInstanceRoleForRdsCustomInstance.
- Passaggio 2: aggiungere una politica di accesso a AWSRDSCustomInstanceRoleForRdsCustomInstance.
- Passaggio 2: aggiungere una politica di accesso a AWSRDSCustomInstanceRoleForRdsCustomInstance.
- Fase 4: Aggiungi AWSRDSCustomInstanceRoleForRdsCustomInstance a AWSRDSCustomInstanceProfile.

Fase 1: creazione di un ruolo IAM per AWSRDSCustomInstanceRoleForRdsCustomInstance

In questo passaggio, crei il ruolo utilizzando il formato di denominazione `AWSRDSCustomInstanceRole-`*`region`*. Utilizzando la policy di affidabilità, Amazon EC2 può assumere il ruolo. L'esempio seguente presuppone che tu abbia impostato la variabile di ambiente `$REGION` nella Regione AWS in cui desideri creare l'istanza database.

```
aws iam create-role \
  --role-name AWSRDSCustomInstanceRole-$REGION \
  --assume-role-policy-document '{
    "Version": "2012-10-17",
      "Statement": [
        {
          "Action": "sts:AssumeRole",
          "Effect": "Allow",
          "Principal": {
              "Service": "ec2.amazonaws.com"
          }
        }
      ]
    }'
```

Passaggio 2: aggiungere una politica di accesso a AWSRDSCustomInstanceRoleForRdsCustomInstance

Quando si incorpora una policy in linea in un ruolo IAM, la policy in linea viene utilizzata come parte della policy di accesso (autorizzazioni) del ruolo. Creare la policy `AWSRDSCustomIamRolePolicy` che consente ad Amazon EC2 di inviare e ricevere messaggi ed eseguire varie azioni.

Nell'esempio seguente viene creata la policy di accesso denominata `AWSRDSCustomIamRolePolicy` e la si aggiunge al ruolo IAM`AWSRDSCustomInstanceRole-`*`region`*. Questo esempio presuppone che siano state impostate le seguenti variabili di ambiente:

`$REGION`

Imposta questa variabile sulla variabile Regione AWS in cui intendi creare l'istanza DB.

`$ACCOUNT_ID`

Imposta questa variabile sul tuo Account AWS numero.

`$KMS_KEY`

Imposta questa variabile sul nome della risorsa Amazon (ARN) della AWS KMS key da utilizzare per le istanze database RDS Custom. Per specificare più di una chiave KMS, aggiungerla alla sezione `Resources` dell'istruzione ID (Sid) 11.

```
aws iam put-role-policy \
  --role-name AWSRDSCustomInstanceRole-$REGION \
  --policy-name AWSRDSCustomIamRolePolicy \
  --policy-document '{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "1",
            "Effect": "Allow",
            "Action": [
                "ssm:DescribeAssociation",
                "ssm:GetDeployablePatchSnapshotForInstance",
                "ssm:GetDocument",
                "ssm:DescribeDocument",
                "ssm:GetManifest",
                "ssm:GetParameter",
                "ssm:GetParameters",
```

```
                "ssm:ListAssociations",
                "ssm:ListInstanceAssociations",
                "ssm:PutInventory",
                "ssm:PutComplianceItems",
                "ssm:PutConfigurePackageResult",
                "ssm:UpdateAssociationStatus",
                "ssm:UpdateInstanceAssociationStatus",
                "ssm:UpdateInstanceInformation",
                "ssm:GetConnectionStatus",
                "ssm:DescribeInstanceInformation",
                "ssmmessages:CreateControlChannel",
                "ssmmessages:CreateDataChannel",
                "ssmmessages:OpenControlChannel",
                "ssmmessages:OpenDataChannel"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Sid": "2",
            "Effect": "Allow",
            "Action": [
                "ec2messages:AcknowledgeMessage",
                "ec2messages:DeleteMessage",
                "ec2messages:FailMessage",
                "ec2messages:GetEndpoint",
                "ec2messages:GetMessages",
                "ec2messages:SendReply"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Sid": "3",
            "Effect": "Allow",
            "Action": [
                "logs:PutRetentionPolicy",
                "logs:PutLogEvents",
                "logs:DescribeLogStreams",
                "logs:DescribeLogGroups",
                "logs:CreateLogStream",
                "logs:CreateLogGroup"
```

```
            ],
            "Resource": [
                "arn:aws:logs:'$REGION':'$ACCOUNT_ID':log-group:rds-custom-instance*"
            ]
        },
        {
            "Sid": "4",
            "Effect": "Allow",
            "Action": [
                "s3:putObject",
                "s3:getObject",
                "s3:getObjectVersion"
            ],
            "Resource": [
                "arn:aws:s3:::do-not-delete-rds-custom-*/*"
            ]
        },
        {
            "Sid": "5",
            "Effect": "Allow",
            "Action": [
                "cloudwatch:PutMetricData"
            ],
            "Resource": [
                "*"
            ],
            "Condition": {
                "StringEquals": {
                    "cloudwatch:namespace": [
                        "RDSCustomForOracle/Agent"
                    ]
                }
            }
        },
        {
            "Sid": "6",
            "Effect": "Allow",
            "Action": [
                "events:PutEvents"
            ],
            "Resource": [
                "*"
            ]
        },
```

```
        {
            "Sid": "7",
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue",
                "secretsmanager:DescribeSecret"
            ],
            "Resource": [
                "arn:aws:secretsmanager:'$REGION':'$ACCOUNT_ID':secret:do-not-delete-
rds-custom-*"
            ]
        },
        {
           "Sid": "8",
           "Effect": "Allow",
           "Action": [
             "s3:ListBucketVersions"
           ],
           "Resource": [
             "arn:aws:s3:::do-not-delete-rds-custom-*"
           ]
         },
         {
            "Sid": "9",
            "Effect": "Allow",
            "Action": "ec2:CreateSnapshots",
            "Resource": [
                "arn:aws:ec2:*:*:instance/*",
                "arn:aws:ec2:*:*:volume/*"
            ],
            "Condition": {
                "StringEquals": {
                    "ec2:ResourceTag/AWSRDSCustom": "custom-oracle"
                }
            }
          },
          {
            "Sid": "10",
            "Effect": "Allow",
            "Action": "ec2:CreateSnapshots",
            "Resource": [
                "arn:aws:ec2:*::snapshot/*"
            ]
          },
```

```
          {
            "Sid": "11",
            "Effect": "Allow",
            "Action": [
              "kms:Decrypt",
              "kms:GenerateDataKey"
            ],
            "Resource": [
              "arn:aws:kms:'$REGION':'$ACCOUNT_ID':key/'$KMS_KEY'"
            ]
          },
          {
            "Sid": "12",
            "Effect": "Allow",
            "Action": "ec2:CreateTags",
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "ec2:CreateAction": [
                        "CreateSnapshots"
                    ]
                }
            }
        }
    ]
}'
```

Passaggio 3: creare il profilo dell'istanza RDS Custom AWSRDSCustomInstanceProfile

Un profilo dell'istanza è un container che include un ruolo IAM singolo. RDS Custom utilizza il profilo dell'istanza per trasferire il ruolo all'istanza.

Se utilizzi la CLI per creare un ruolo, è necessario creare il ruolo e il profilo dell'istanza come operazioni distinte, con nomi potenzialmente diversi. Crea il profilo dell'istanza IAM come segue, denominandolo utilizzando il formato `AWSRDSCustomInstanceProfile-`*`region`*. L'esempio seguente presuppone che la variabile di ambiente sia stata impostata `$REGION` su quella Regione AWS in cui si desidera creare l'istanza DB.

```
aws iam create-instance-profile \
    --instance-profile-name AWSRDSCustomInstanceProfile-$REGION
```

Fase 4: Aggiungi AWSRDSCustomInstanceRoleForRdsCustomInstance a AWSRDSCustomInstanceProfile

Aggiungi il ruolo IAM al profilo dell'istanza creato in precedenza. L'esempio seguente presuppone che la variabile di ambiente sia stata impostata `$REGION` su quella Regione AWS in cui si desidera creare l'istanza DB.

```
aws iam add-role-to-instance-profile \
    --instance-profile-name AWSRDSCustomInstanceProfile-$REGION \
    --role-name AWSRDSCustomInstanceRole-$REGION
```

Passaggio 5: concedi le autorizzazioni necessarie al tuo utente o ruolo IAM

Assicurati che il principale IAM (utente o ruolo) che crea l'istanza DB personalizzata CEV o RDS disponga di una delle seguenti politiche:

- La policy `AdministratorAccess`
- La `AmazonRDSFullAccess` policy con le autorizzazioni richieste per Amazon S3 AWS KMS e la creazione di CEV e la creazione di istanze DB

Argomenti

- Autorizzazioni IAM obbligatorie per Amazon S3 e AWS KMS
- Autorizzazioni IAM richieste per la creazione di una CEV
- Autorizzazioni richieste per la creazione di un'istanza database da una CEV

Autorizzazioni IAM obbligatorie per Amazon S3 e AWS KMS

Per creare CEV o RDS Custom per istanze DB Oracle, il tuo principale IAM deve accedere ad Amazon S3 e. AWS KMS La policy JSON di esempio seguente fornisce le autorizzazioni necessarie.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "CreateS3Bucket",
            "Effect": "Allow",
            "Action": [
                "s3:CreateBucket",
```

```
                "s3:PutBucketPolicy",
                "s3:PutBucketObjectLockConfiguration",
                "s3:PutBucketVersioning"
            ],
            "Resource": "arn:aws:s3:::do-not-delete-rds-custom-*"
        },
        {
            "Sid": "CreateKmsGrant",
            "Effect": "Allow",
            "Action": [
                "kms:CreateGrant",
                "kms:DescribeKey"
            ],
            "Resource": "*"
        }
    ]
}
```

Per ulteriori informazioni sull'autorizzazione `kms:CreateGrant`, consulta Gestione di AWS KMS key.

Autorizzazioni IAM richieste per la creazione di una CEV

Per creare un CEV, il tuo principale IAM necessita delle seguenti autorizzazioni aggiuntive:

```
s3:GetObjectAcl
s3:GetObject
s3:GetObjectTagging
s3:ListBucket
mediaimport:CreateDatabaseBinarySnapshot
```

La seguente policy JSON di esempio concede le autorizzazioni aggiuntive necessarie per accedere al bucket e al suo contenuto. *my-custom-installation-files*

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AccessToS3MediaBucket",
            "Effect": "Allow",
            "Action": [
                "s3:GetObjectAcl",
```

```
                "s3:GetObject",
                "s3:GetObjectTagging",
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::my-custom-installation-files",
                "arn:aws:s3:::my-custom-installation-files/*"
            ]
        },
        {
            "Sid": "PermissionForByom",
            "Effect": "Allow",
            "Action": [
                "mediaimport:CreateDatabaseBinarySnapshot"
            ],
            "Resource": "*"
        }
    ]
}
```

Puoi concedere autorizzazioni simili per Amazon S3 agli account dei chiamanti utilizzando una politica del bucket S3.

Autorizzazioni richieste per la creazione di un'istanza database da una CEV

Per creare un'istanza DB RDS Custom for Oracle da un CEV esistente, il principale IAM necessita delle seguenti autorizzazioni aggiuntive.

```
iam:SimulatePrincipalPolicy
cloudtrail:CreateTrail
cloudtrail:StartLogging
```

La seguente policy JSON di esempio concede le autorizzazioni necessarie per convalidare un ruolo IAM e registrare le informazioni su un AWS CloudTrail.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ValidateIamRole",
            "Effect": "Allow",
            "Action": "iam:SimulatePrincipalPolicy",
```

```
            "Resource": "*"
        },
        {
            "Sid": "CreateCloudTrail",
            "Effect": "Allow",
            "Action": [
                "cloudtrail:CreateTrail",
                "cloudtrail:StartLogging"
            ],
            "Resource": "arn:aws:cloudtrail:*:*:trail/do-not-delete-rds-custom-*"
        }
    ]
}
```

Fase 6: Configurazione del VPC per RDS Custom for Oracle

L'istanza database di RDS Custom si trova in un cloud privato virtuale (VPC) basato sul servizio Amazon VPC, proprio come un'istanza Amazon EC2 o un'istanza Amazon RDS. Fornisci e configuri il VPC personalizzato. A differenza di RDS Custom per SQL Server, RDS Custom per Oracle non crea una lista di controllo degli accessi (ACL) o gruppi di sicurezza. È necessario associare il proprio gruppo di sicurezza, le proprie sottoreti e le proprie tabelle di routing.

Puoi configurare il tuo cloud privato virtuale (VPC) utilizzando uno dei due CloudFormation o un processo manuale.

> **Important**
>
> Ti consigliamo vivamente di configurare l'ambiente RDS Custom for Oracle utilizzando. AWS CloudFormation Questa tecnica è la più semplice e meno soggetta a errori.

Argomenti

- Configura il tuo VPC usando CloudFormation (consigliato)
- Configurazione manuale del VPC per RDS Custom for Oracle

Configura il tuo VPC usando CloudFormation (consigliato)

Se hai già configurato il VPC per un motore RDS Custom diverso e vuoi riutilizzare il VPC esistente, questo passaggio può essere ignorato. In questa sezione si presuppone quanto segue:

- Hai già creato il profilo e il ruolo dell'istanza IAM. CloudFormation
- Conosci l'ID della tabella di routing.

  Affinché un'istanza database sia privata, deve trovarsi in una sottorete privata. Affinché una sottorete sia privata, non deve essere associata a una tabella di routing con un gateway Internet. Per maggiori informazioni, consulta Configurazione delle tabelle di instradamento nella Guida per l'utente di Amazon VPC.

Quando usi il CloudFormation modello per il tuo VPC, crea le seguenti risorse:

- Un VPC privato
- Un gruppo di sottoreti denominato `rds-custom-private`
- I seguenti endpoint VPC, che l'istanza DB utilizza per comunicare con, dipendono: Servizi AWS
  - `com.amazonaws.region.ec2messages`
  - `com.amazonaws.region.events`
  - `com.amazonaws.region.logs`
  - `com.amazonaws.region.monitoring`
  - `com.amazonaws.region.s3`
  - `com.amazonaws.region.secretsmanager`
  - `com.amazonaws.region.ssm`
  - `com.amazonaws.region.ssmmessages`

> **Note**
>
> Per una configurazione di rete complessa con account esistenti, consigliamo di configurare manualmente l'accesso ai servizi dipendenti se l'accesso non esiste già. Per ulteriori informazioni, consulta Assicurati che il tuo VPC possa accedere in modo dipendente Servizi AWS.

Per configurare il tuo VPC utilizzando CloudFormation

1. Apri la CloudFormation console all'indirizzo https://console.aws.amazon.com/cloudformation.
2. Avvia la procedura guidata Crea stack, scegli Crea stack e poi Con nuove risorse (standard).
3. Nella pagina Create stack (Crea stack), esegui le operazioni seguenti:

a. In Prepare template (Prepara modello) scegli Template is ready (Il modello è pronto).

b. Come Template source (Origine modello), scegliere Upload a template file (Carica un file di modello).

c. Per Scegliere file, andare su e scegliere `custom-vpc.json`.

d. Seleziona Successivo.

4. Nella pagina Specify stack details (Specifica dettagli), procedere come segue:

a. In Nome stack, immetti **`custom-vpc`**.

b. Come Parameters (Parametri), scegliere le sottoreti private da utilizzare per le istanze database RDS Custom.

c. Scegliere l'ID VPC privato da utilizzare per le istanze database RDS Custom.

d. Inserire la tabella di routing associata alle sottoreti private.

e. Seleziona Successivo.

5. Nella pagina Configure stack options (Configura opzioni pila), scegliere Next (Successivo).

6. Nella pagina Verifica custom-vpc, scegli Invia.

CloudFormation configura il tuo VPC privato. Nel pannello a sinistra, quando custom-vpc indica CREATE_COMPLETE, esegui al passaggio successivo.

7. (Facoltativo) Verifica i dettagli del VPC. Nel riquadro Stack, scegli custom-vpc. Nel riquadro di destra eseguire queste operazioni:

a. Scegli Informazioni stack. Lo stack ha un ID nel formato arn:aws:cloudformation:***`region`***:***`account-no`***:stack/custom-vpc/***`identifier`***.

b. Scegliere Resources (Risorse). ***`Dovresti vedere un gruppo di sottoreti denominato rds-custom-privatee diversi endpoint VPC che utilizzano il formato di denominazione vpce- string.`*** Ogni endpoint corrisponde a un dispositivo con cui RDS Custom deve Servizio AWS comunicare. Per ulteriori informazioni, consulta Assicurati che il tuo VPC possa accedere in modo dipendente Servizi AWS.

c. Scegli Aggiungi parametro. Dovresti vedere le sottoreti private, il VPC privato e la tabella di routing che hai specificato quando hai creato lo stack. Quando viene creata un'istanza database, è necessario fornire l'ID VPC e il gruppo di sottoreti.

Configurazione manuale del VPC per RDS Custom for Oracle

In alternativa all'automazione della creazione di VPC AWS CloudFormation con, puoi configurare il tuo VPC manualmente. Questa opzione potrebbe essere la migliore quando si dispone di una configurazione di rete complessa che utilizza le risorse esistenti.

Argomenti

- Assicurati che il tuo VPC possa accedere in modo dipendente Servizi AWS
- Configurazione del servizio di metadati dell'istanza

Assicurati che il tuo VPC possa accedere in modo dipendente Servizi AWS

RDS Custom invia la comunicazione dall'istanza database ad altri Servizi AWS. Assicurati che i seguenti servizi siano accessibili dalla sottorete in cui crei le tue istanze DB personalizzate RDS:

- Amazon CloudWatch
- CloudWatch Registri Amazon
- CloudWatch Eventi Amazon
- Amazon EC2
- Amazon EventBridge
- Amazon S3
- AWS Secrets Manager
- AWS Systems Manager

Se si creano implementazioni Multi-AZ

- Amazon Simple Queue Service

Se RDS Custom non è in grado di comunicare con i servizi necessari, pubblica i seguenti eventi:

```
Database instance in incompatible-network. SSM Agent connection not available. Amazon
 RDS can't connect to the dependent AWS services.
```

```
Database instance in incompatible-network. Amazon RDS can't connect to dependent AWS
 services. Make sure port 443 (HTTPS) allows outbound connections, and try again.
 "Failed to connect to the following services: s3 events"
```

Per evitare `incompatible-network` errori, assicurati che i componenti VPC coinvolti nella comunicazione tra l'istanza DB personalizzata di RDS Servizi AWS soddisfino i seguenti requisiti:

- L'istanza database può effettuare connessioni in uscita sulla porta 443 ad altri Servizi AWS.
- Il VPC consente risposte in entrata alle richieste che originano dall'istanza database RDS Custom.
- RDS Custom può risolvere correttamente i nomi di dominio degli endpoint per ogni Servizio AWS.

Se hai già configurato un VPC per un motore di database RDS Custom diverso, puoi riutilizzare tale VPC e ignorare questo processo.

Configurazione del servizio di metadati dell'istanza

Verificare che l'istanza possa fare:

- Accedere ai metadati dell'istanza utilizzando la versione 2 del servizio di metadati dell'istanza (IMDSv2).
- Consenti comunicazioni in uscita tramite la porta 80 (HTTP) all'indirizzo IP del collegamento IMDS.
- Richiedere metadati dell'istanza da`http://169.254.169.254`, il link IMDSv2.

Per ulteriori informazioni, consultare Utilizzare IMDSv2 nella Guida per l'utente di Amazon EC2 per le istanze Linux.

L'automazione RDS Custom per Oracle utilizza IMDSv2 per impostazione predefinita, impostando `HttpTokens=enabled`sull'istanza Amazon EC2 sottostante. Tuttavia, puoi utilizzare IMDSv1, se necessario. Per ulteriori informazioni, consultare Configurazione delle opzioni di metadati dell'istanza nella Guida per l'utente di Amazon EC2 per le istanze Linux.

Utilizzo di versioni del motore personalizzate per Amazon RDS Custom per Oracle

Una versione del motore personalizzato (CEV)per Amazon RDS Custom per Oracle è una snapshot del volume binario di un motore di database e di una specifica Amazon Machine Image (AMI). Per impostazione predefinita, RDS Custom per Oracle utilizza l'AMI più recente disponibile gestita da RDS Custom, ma puoi specificare un'AMI utilizzata in una CEV precedente. Archiviare i file di installazione del database in Amazon S3. RDS Custom utilizza i file di installazione e l'AMI per creare automaticamente la CEV.

Argomenti

Preparazione alla creazione di un CEV

Per creare un CEV, accedi ai file di installazione e alle patch archiviati nel bucket Amazon S3 per una delle seguenti versioni:

- Oracle Database 19c
- Oracle Database 18c
- Oracle Database 12c Release 2 (12.2)
- Oracle Database 12c Release 1 (12.1)

Ad esempio, è possibile utilizzare RU/RUR di aprile 2021 per Oracle Database 19c o qualsiasi combinazione valida di file di installazione e patch. Per ulteriori informazioni sulle versioni e sulle regioni supportate da RDS Custom per Oracle, consulta RDS Custom con RDS per Oracle.

Argomenti

- Passaggio 4 (opzionale): condividi i supporti di installazione in S3 su Account AWS
- Fase 5: preparazione del manifesto CEV
- Fase 6 (facoltativo): convalida del manifesto CEV
- Fase 7: aggiunta delle autorizzazioni IAM necessarie

Fase 1 (facoltativo): download dei modelli di manifesto

Un manifesto CEV è un documento JSON che include l'elenco dei file di installazione del database in formato .zip per la tua CEV. Per creare una CEV, procedi come descritto qui di seguito:

1. Individua i file di installazione del database Oracle che desideri includere nella CEV.
2. Scarica i file di installazione.
3. Crea un manifesto JSON contenente l'elenco dei file di installazione.

RDS Custom per Oracle fornisce modelli di manifesto JSON con i file .zip consigliati per ogni versione supportata del database Oracle. Ad esempio, il seguente modello è per la RU 19.17.0.0.0.

```
{
    "mediaImportTemplateVersion": "2020-08-14",
    "databaseInstallationFileNames": [
        "V982063-01.zip"
    ],
    "opatchFileNames": [
        "p6880880_190000_Linux-x86-64.zip"
    ],
    "psuRuPatchFileNames": [
        "p34419443_190000_Linux-x86-64.zip",
        "p34411846_190000_Linux-x86-64.zip"
    ],
    "otherPatchFileNames": [
        "p28852325_190000_Linux-x86-64.zip",
        "p29997937_190000_Linux-x86-64.zip",
        "p31335037_190000_Linux-x86-64.zip",
        "p32327201_190000_Linux-x86-64.zip",
        "p33613829_190000_Linux-x86-64.zip",
        "p34006614_190000_Linux-x86-64.zip",
        "p34533061_190000_Linux-x86-64.zip",
        "p34533150_190000_Generic.zip",
        "p28730253_190000_Linux-x86-64.zip",
```

```
        "p29213893_1917000DBRU_Generic.zip",
        "p33125873_1917000DBRU_Linux-x86-64.zip",
        "p34446152_1917000DBRU_Linux-x86-64.zip"
    ]
}
```

A ogni modello è associato un file Readme contenente le istruzioni per scaricare le patch, gli URL per i file. zip e i checksum dei file. Puoi usare questi modelli così come sono o modificarli con le tue patch. Per rivedere i modelli, scaricate il custom-oracle-manifestfile.zip sul disco locale e apritelo con un'applicazione per l'archiviazione dei file. Per ulteriori informazioni, consulta Fase 5: preparazione del manifesto CEV.

Fase 2: download di file e patch di installazione del database da Oracle Software Delivery Cloud

Dopo aver individuato i file di installazione per la tua CEV, scaricali sul tuo sistema locale. I file di installazione e le patch di Oracle Database sono ospitati su Oracle Software Delivery Cloud. Ogni CEV richiede una versione di base, ad esempio Oracle Database 19c Release 2 (12.2) e un elenco facoltativo di patch.

Per scaricare i file di installazione del database per Oracle Database

1. Andare su https://edelivery.oracle.com/ e accedere.
2. Nella casella di ricerca, inserisci **`Oracle Database Enterprise Edition`** o **`Oracle Database Standard Edition 2`** e scegli Cerca.
3. Seleziona una delle seguenti versioni base:

Versione del database	Enterprise Edition	Standard Edition 2
Oracle Database 19c	DLP: Oracle Database 19c Enterprise Edition 19.3.0.0.0 (Oracle Database Enterprise Edition)	DLP: Oracle Database 19c Standard Edition 2 19.3.0.0.0 (Oracle Database Standard Edition 2)
Oracle Database 18c	DLP: Oracle Database 18c Enterprise Edition 18.0.0.0.0 (Oracle Database Enterprise Edition)	DLP: Oracle Database Standard Edition 2 18.0.0.0.0 (Oracle Database Standard Edition 2)

Versione del database	Enterprise Edition	Standard Edition 2
Oracle Database 12c Release 2 (12.2.0.1)	DLP: Oracle Database 12c Enterprise Edition 12.2.0.1.0 (Oracle Database Enterprise Edition)	DLP: Oracle Database Standard Edition 2 12.2.0.1.0 (Oracle Database Standard Edition 2)
Oracle Database 12c Release 1 (12.1.0.2)	DLP: Oracle Database 12c Enterprise Edition 12.1.0.2.0 (Oracle Database Enterprise Edition)	DLP: Oracle Database Standard Edition 2 12.1.0.2.0 (Oracle Database Standard Edition 2)

4. Scegli Continua.
5. Deseleziona la casella di controllo Download Queue (Scarica coda).
6. Seleziona l'opzione che corrisponde alla versione di base:
   - Oracle Database 19.3.0.0.0: versione a lungo termine
   - Oracle Database 18.0.0.0.0
   - Oracle Database 12.2.0.1.0
   - Oracle Database 12.1.0.2.0
7. Scegli Linux x86-64 in Piattaforma/lingue.
8. Scegli Continua, quindi firma il contratto di licenza Oracle.
9. Seleziona il file .zip corrispondente alla versione del database:

Versione ed edizione del database	File in formato .zip	SHA-256 hash
19c EE e SE2	V982063-01.zip	`BA8329C757133DA313ED3B6D7F86C5AC42CD9970A28BF2E6233F3235233AA8D8`
18c EE e SE2	V978967-01.zip	`C96A4FD768787AF98272008833FE10B172691CF84E42816B138C12D4DE63AB96`

Versione ed edizione del database	File in formato .zip	SHA-256 hash
12.2.0.1 EE e SE2	V839960-01.zip	96ED97D21F15C1AC0CCE3749DA6C3DAC7059BB60672D76B008103FC754D22DDE
12.1.0.2 EE	V46095-01_1of2.zip V46095-01_2of2.zip	31FDC2AF41687B4E547A3A18F796424D8C1AF36406D2160F65B0AF6A9CD47355 per V46095-01_1of2.zip 03DA14F5E875304B28F0F3BB02AF0EC33227885B99C9865DF70749D1E220ACCD per V46095-01_2of2.zip
12.1.0.2 SE2	V77388-01_1of2.zip V77388-01_2of2.zip	73873369753230F5A0921F95ACEADB591388CB06ED72A7F3AEA7BCBCEA2403BC per V77388-01_1of2.zip 2492E1BE1E3E3531DA83D0843C09C08E435AC8CEFD9A00C0DF56BE4F15CEEBF3 per V77388-01_2of2.zip

10. Scarica le patch Oracle desiderate da `updates.oracle.com` o `support.oracle.com` nel tuo sistema locale. Gli URL delle patch sono disponibili nelle seguenti posizioni:

    - I file Readme contenuti nel file. zip scaricato in Fase 1 (facoltativo): download dei modelli di manifesto
    - Le patch elencate in ciascun RU (Release Update, aggiornamento rilascio) riportato nella pagina relativa alle note di rilascio per Amazon Relational Database Service (Amazon RDS) per Oracle

Fase 3: caricamento dei file di installazione in Amazon S3

Caricare i file di installazione e patch Oracle in Amazon S3 tramite AWS CLI. Il bucket S3 che contiene i file di installazione deve trovarsi nella stessa AWS regione del CEV.

Gli esempi in questa sezione utilizzano i seguenti segnaposto:

- *`install-or-patch-file.zip`* – File multimediale di installazione Oracle. Ad esempio, p32126828_190000_Linux-x86-64.zip è una patch.
- *`my-custom-installation-files`* – Il bucket Amazon S3 designato per i file di installazione caricati.
- *`123456789012/cev1`* – Un prefisso opzionale nel bucket Amazon S3.
- *`source-bucket`* – Un bucket Amazon S3 in cui è possibile facoltativamente organizzare i file.

Argomenti

Passaggio 3a: verifica che il bucket S3 sia nella posizione corretta Regione AWS

Verifica che il bucket S3 si trovi nella AWS regione in cui intendi eseguire il comando. `create-custom-db-engine-version`

```
aws s3api get-bucket-location --bucket my-custom-installation-files
```

Fase 3b: verifica delle autorizzazioni corrette della policy del bucket S3

È possibile creare una CEV da zero o da una CEV di origine. Se hai intenzione di creare una nuova CEV dalle CEV di origine, assicurati che la tua policy sui bucket S3 disponga delle autorizzazioni corrette:

1. Identifica il bucket S3 riservato da RDS Custom. Il formato del nome del bucket è `do-not-delete-rds-custom-`*`account`*`-`*`region`*`-`*`string`*. Ad esempio, il nome del bucket potrebbe essere `do-not-delete-rds-custom-`*`123456789012`*`-`*`us-east-1`*`-`*`abc123EXAMPLE`*.
2. Assicurati che la seguente autorizzazione sia aggiunta alla policy dei bucket S3. Sostituisci `do-not-delete-rds-custom-`*`123456789012`*`-`*`us-east-1`*`-`*`abc123EXAMPLE`* con il nome del tuo bucket.

```
{
```

```
    "Sid": "AWSRDSCustomForOracleCustomEngineVersionGetObject",
    "Effect": "Allow",
    "Principal": {
        "Service": "custom.rds.amazonaws.com"
    },
    "Action": [
        "s3:GetObject",
        "s3:GetObjectTagging"
    ],
    "Resource": "arn:aws:s3:::do-not-delete-rds-custom-123456789012-us-
east-1-abc123EXAMPLE/CustomEngineVersions/*"
}, ...
```

Fase 3c: caricamento dei file utilizzando i comandi cp o sync

Seleziona una delle seguenti opzioni:

- Utilizza `aws s3 cp` per caricare un singolo file.zip.

  Carica ogni file con estensione zip di installazione separatamente. Non combinare i file.zip in un singolo file.zip.
- Utilizza `aws s3 sync` per caricare una directory.

Example

Viene caricato l'esempio *install-or-patch-file.zip* riportato di seguito nella cartella *123456789012/cev1* nel bucket Amazon S3 RDS Custom. Esegui un comando `aws s3` separato per ogni .zip che si desidera caricare.

PerLinux, omacOS: Unix

```
aws s3 cp install-or-patch-file.zip \
    s3://my-custom-installation-files/123456789012/cev1/
```

Per Windows:

```
aws s3 cp install-or-patch-file.zip ^
    s3://my-custom-installation-files/123456789012/cev1/
```

Example

Nel seguente esempio vengono caricati i file nella propria cartella locale *cev1* nella cartella *123456789012/cev1* nel bucket Amazon S3.

Per LinuxmacOS, oUnix:

```
aws s3 sync cev1 \
    s3://my-custom-installation-files/123456789012/cev1/
```

Per Windows:

```
aws s3 sync cev1 ^
    s3://my-custom-installation-files/123456789012/cev1/
```

Example

Nel seguente esempio vengono caricati tutti i file in *source-bucket* nella cartella *123456789012/cev1* nel bucket Amazon S3.

Per LinuxmacOS, oUnix:

```
aws s3 sync s3://source-bucket/ \
    s3://my-custom-installation-files/123456789012/cev1/
```

Per Windows:

```
aws s3 sync s3://source-bucket/ ^
    s3://my-custom-installation-files/123456789012/cev1/
```

Fase 3d: elenco dei file nel bucket S3

L'esempio seguente utilizza il comando `s3 ls` per elencare i file nel bucket S3 Amazon RDS Custom.

```
aws s3 ls \
    s3://my-custom-installation-files/123456789012/cev1/
```

Passaggio 4 (opzionale): condividi i supporti di installazione in S3 su Account AWS

Ai fini di questa sezione, il bucket Amazon S3 che contiene i file di installazione Oracle caricati è il bucket dei file di installazione. La tua organizzazione potrebbe utilizzarne più di uno Account AWS in un Regione AWS. In tal caso, potresti volerne usare uno Account AWS per popolare il tuo bucket multimediale e un altro Account AWS per creare CEV. Se non desideri condividere il bucket dei file di installazione, passa alla sezione successiva.

In questa sezione si presuppone quanto segue:

- Puoi accedere all'account che ha creato il bucket dei file di installazione e a un altro account in cui intendi creare il CEV.
- Hai intenzione di creare CEV in una sola Regione AWS. Se intendi utilizzare più regioni, crea un bucket dei file di installazione in ciascuna regione.
- Stai usando la CLI. Se stai utilizzando la console Amazon S3, procedi nel seguente modo:

Per configurare il tuo bucket multimediale per la condivisione su Account AWS

1. Accedi al bucket S3 Account AWS che contiene il bucket S3 in cui hai caricato il supporto di installazione.
2. Inizia con un modello di policy JSON vuoto o con una policy esistente che puoi adattare.

   Il comando seguente recupera una policy esistente e la salva come *my-policy.json*. In questo esempio, viene denominato il bucket S3 contenente i file di installazione. *oracle-media-bucket*

   ```
   aws s3api get-bucket-policy \
       --bucket oracle-media-bucket \
       --query Policy \
       --output text > my-policy.json
   ```

3. Modifica le autorizzazioni del bucket dei file di installazione come segue:

   - Nell'elemento `Resource` del modello, specifica il bucket S3 in cui hai caricato i file di installazione di Oracle Database.
   - Nell'`Principal`elemento, specificate gli ARN per tutto ciò Account AWS che intendete utilizzare per creare CEV. Puoi aggiungere il root, un utente o un ruolo all'elenco degli indirizzi consentiti del bucket S3. Per ulteriori informazioni, consultare Identificatori IAM nella Guida per l'utente di AWS Identity and Access Management .

```
{
    "Version": "2008-10-17",
    "Statement": [
        {
            "Sid": "GrantAccountsAccess",
            "Effect": "Allow",
            "Principal": {
                "AWS": [
                    "arn:aws:iam::account-1:root",
                    "arn:aws:iam::account-2:user/user-name-with-path",
                    "arn:aws:iam::account-3:role/role-name-with-path",
                    ...
                ]
            },
            "Action": [
                "s3:GetObject",
                "s3:GetObjectAcl",
                "s3:GetObjectTagging",
                "s3:ListBucket",
                "s3:GetBucketLocation"
            ],
            "Resource": [
                "arn:aws:s3:::oracle-media-bucket",
                "arn:aws:s3:::oracle-media-bucket/*"
            ]
        }
    ]
}
```

4. Collega la policy al bucket dei file di installazione.

   Nell'esempio seguente, *oracle-media-bucket*è il nome del bucket S3 che contiene i file di installazione e *my-policy.json* è il nome del file JSON.

   ```
   aws s3api put-bucket-policy \
       --bucket oracle-media-bucket \
       --policy file://my-policy.json
   ```

5. Accedi a un file in cui intendi creare CEV Account AWS .
6. Verifica che questo account possa accedere al bucket multimediale nel quale è stato creato Account AWS .

```
aws s3 ls --query "Buckets[].Name"
```

Per ulteriori informazioni, consulta aws s3 ls nel Riferimento ai comandi della AWS CLI Command Reference.

7. Crea un CEV seguendo i passaggi descritti in Creazione di un CEV.

Fase 5: preparazione del manifesto CEV

Un manifesto CEV è un documento JSON che include quanto segue:

- (Obbligatorio) L'elenco dei file .zip di installazione caricati in Amazon S3. RDS Custom applica le patch nell'ordine in cui sono elencate nel manifesto.
- (Facoltativo) I parametri di installazione che impostano valori non predefiniti per la base Oracle, la home Oracle e l'ID e il nome dell'utente e del gruppo UNIX/Linux. Tieni presente che non puoi modificare i parametri di installazione per una CEV esistente o un'istanza database esistente. Inoltre, non è possibile eseguire l'aggiornamento da una CEV a un'altra quando i parametri di installazione hanno impostazioni diverse.

Per esempi di manifesti CEV, consulta i modelli JSON scaricati in Fase 1 (facoltativo): download dei modelli di manifesto. Puoi esaminare gli esempi in Esempi di manifesto CEV.

Argomenti

- Campi JSON nel manifesto CEV
- Creazione del manifest CEV
- Esempi di manifesto CEV

Campi JSON nel manifesto CEV

La seguente tabella descrive i campi JSON nel manifest.

Campi JSON nel manifesto CEV

Campo JSON	Descrizione
`MediaImportTemplateVersion`	Versione del manifest CEV. Il formato della data è `YYYY-MM-DD` .

Campo JSON	Descrizione
`databaseInstallationFileNames`	Elenco ordinato dei file di installazione per il database.
`opatchFileNames`	Elenco ordinato dei programmi di installazione OPatch utilizzati per il motore di database Oracle. È valido solo un valore. I valori per `opatchFileNames` devono iniziare con `p6880880_` .
`psuRuPatchFileNames`	Patch PSU e RU per questo database. **Important** Se includi `psuRuPatchFileNames` , `opatchFileNames` è obbligatorio. I valori per `opatchFileNames` devono iniziare con `p6880880_` .
`OtherPatchFileNames`	Le patch che non sono incluse nell'elenco delle patch PSU e RU. RDS Custom applica queste patch dopo aver applicato le patch PSU e RU. **Important** Se includi `OtherPatchFileNames` , `opatchFileNames` è obbligatorio. I valori per `opatchFileNames` devono iniziare con `p6880880_` .

Campo JSON	Descrizione
`installationParameters`	Le impostazioni non predefinite per la base Oracle, la home Oracle e l'ID e il nome dell'utente e del gruppo UNIX/Linux. Puoi impostare i seguenti parametri: oracleBase La directory in cui sono installati i file binari Oracle. È il punto di montaggio del volume binario in cui vengono archiviati i file. La directory di base Oracle può includere più home Oracle. Ad esempio, se `/home/oracle/oracle.19.0.0.0.ru-2020-04.rur-2020-04.r1.EE.1` è una home directory Oracle, `/home/oracle` è la directory di base Oracle. Una directory di base Oracle specificata dall'utente non è un collegamento simbolico. Se non specifichi la base Oracle, la directory predefinita è `/rdsdbbin` . oracleHome La directory in cui sono installati i file binari del database Oracle. Ad esempio, se si specifica `/home/oracle/` come base Oracle, è possibile specificare`/home/oracle/oracle.19.0.0.0.ru-2020-04.rur-2020-04.r1.EE.1/` come home Oracle. Una directory home Oracle specificata dall'utente non è un collegamento simbolico. Al valore home Oracle fa riferimento la variabile di ambiente `$ORACLE_HOME` . Se non specifichi la home Oracle, il formato di denominazione predefinito è `/rdsdbbin/oracle.` *`major-engine-version`* `.custom.r1.` *`engine-edition`* `.1`. unixUname Il nome dell'utente UNIX proprietario del software Oracle. RDS Custom assume questo utente durante l'esecuzione dei comandi del database locale. Se si specificano entrambi

Campo JSON	Descrizione
	`unixUid` e `unixUname` , RDS Custom crea l'utente se non esiste e quindi assegna l'UID all'utente se non è uguale all'UID iniziale. Il nome utente predefinito è `rdsdb`. unixUid L'ID (UID) dell'utente UNIX proprietario del software Oracle. Se si specificano entrambi `unixUid` e `unixUname` , RDS Custom crea l'utente se non esiste e quindi assegna l'UID all'utente se non è uguale all'UID iniziale. Il valore predefinito UID è `61001`. Questo è l'UID dell'utente `rdsdb`. unixGroupName Il nome del gruppo UNIX. L'utente UNIX proprietario del software Oracle appartiene a questo gruppo. Il nome predefinito del gruppo è `rdsdb`. unixGroupId L'ID del gruppo UNIX a cui appartiene l'utente UNIX. L'ID di gruppo predefinito è `1000`. Questo è l'ID del gruppo `rdsdb`.

Ogni versione di Oracle Database ha un elenco diverso di file di installazione supportati. Quando crei il manifesto CEV, assicurati di specificare solo i file supportati da RDS Custom per Oracle. In caso contrario, la creazione di CEV ha esito negativo e restituisce un errore. Sono supportate tutte le patch riportate nella pagina relativa alle [note di rilascio per Amazon Relational Database Service (Amazon RDS) per Oracle]

Creazione del manifest CEV

Per creare un manifesto CEV

1. Elenca tutti i file di installazione da applicare nell'ordine desiderato.
2. Associa i file di installazione ai campi JSON descritti in Campi JSON nel manifesto CEV.
3. Esegui una delle operazioni seguenti:
   - Crea il manifesto CEV come file di testo JSON.
   - Modifica il modello di manifesto CEV quando crei il CEV nella console. Per ulteriori informazioni, consulta Creazione di un CEV.

Esempi di manifesto CEV

Gli esempi seguenti mostrano i file manifesto CEV per diverse versioni di Oracle Database. Se includi un campo JSON nel manifesto, assicurati che non sia vuoto. Ad esempio, il manifesto seguente non è valido perché `otherPatchFileNames` è vuoto.

```
{
    "mediaImportTemplateVersion": "2020-08-14",
    "databaseInstallationFileNames": [
        "V982063-01.zip"
    ],
    "opatchFileNames": [
        "p6880880_190000_Linux-x86-64.zip"
    ],
    "psuRuPatchFileNames": [
        "p32126828_190000_Linux-x86-64.zip"
    ],
    "otherPatchFileNames": [
    ]
}
```

Argomenti

- Sample CEV manifest for Oracle Database 12c Release 1 (12.1)
- Sample CEV manifest for Oracle Database 12c Release 2 (12.2)
- Sample CEV manifest for Oracle Database 18c
- Sample CEV manifest for Oracle Database 19c

Example Esempio di manifesto CEV per Oracle Database 12c Release 1 (12.1)

Nell'esempio seguente per la PSU di luglio 2021 per Oracle Database 12c Release 1 (12.1), RDS Custom applica le patch nell'ordine specificato. Pertanto, RDS Custom applica p32768233, p32876425, quindi p18759211 e così via. L'esempio imposta nuovi valori per l'utente e il gruppo UNIX, la home Oracle e la base Oracle.

```
{
    "mediaImportTemplateVersion":"2020-08-14",
    "databaseInstallationFileNames":[
        "V46095-01_1of2.zip",
        "V46095-01_2of2.zip"
    ],
    "opatchFileNames":[
        "p6880880_121010_Linux-x86-64.zip"
    ],
    "psuRuPatchFileNames":[
        "p32768233_121020_Linux-x86-64.zip"
    ],
    "otherPatchFileNames":[
        "p32876425_121020_Linux-x86-64.zip",
        "p18759211_121020_Linux-x86-64.zip",
        "p19396455_121020_Linux-x86-64.zip",
        "p20875898_121020_Linux-x86-64.zip",
        "p22037014_121020_Linux-x86-64.zip",
        "p22873635_121020_Linux-x86-64.zip",
        "p23614158_121020_Linux-x86-64.zip",
        "p24701840_121020_Linux-x86-64.zip",
        "p25881255_121020_Linux-x86-64.zip",
        "p27015449_121020_Linux-x86-64.zip",
        "p28125601_121020_Linux-x86-64.zip",
        "p28852325_121020_Linux-x86-64.zip",
        "p29997937_121020_Linux-x86-64.zip",
        "p31335037_121020_Linux-x86-64.zip",
        "p32327201_121020_Linux-x86-64.zip",
        "p32327208_121020_Generic.zip",
        "p17969866_12102210119_Linux-x86-64.zip",
        "p20394750_12102210119_Linux-x86-64.zip",
        "p24835919_121020_Linux-x86-64.zip",
        "p23262847_12102201020_Linux-x86-64.zip",
        "p21171382_12102201020_Generic.zip",
        "p21091901_12102210720_Linux-x86-64.zip",
        "p33013352_12102210720_Linux-x86-64.zip",
```

```
        "p25031502_12102210720_Linux-x86-64.zip",
        "p23711335_12102191015_Generic.zip",
        "p19504946_121020_Linux-x86-64.zip"
    ],
    "installationParameters": {
        "unixGroupName": "dba",
        "unixGroupId": 12345,
        "unixUname": "oracle",
        "unixUid": 12345,
        "oracleHome": "/home/oracle/oracle.12.1.0.2",
        "oracleBase": "/home/oracle"
    }
}
```

Example Esempio di manifesto CEV per Oracle Database 12c Release 2 (12.2)

Nell'esempio seguente per la PSU di ottobre 2021 per Oracle Database 12c Release 2 (12.2), RDS Custom applica p33261817, p33192662, quindi p29213893 e così via. L'esempio imposta nuovi valori per l'utente e il gruppo UNIX, la home Oracle e la base Oracle.

```
{
    "mediaImportTemplateVersion":"2020-08-14",
    "databaseInstallationFileNames":[
        "V839960-01.zip"
    ],
    "opatchFileNames":[
        "p6880880_122010_Linux-x86-64.zip"
    ],
    "psuRuPatchFileNames":[
        "p33261817_122010_Linux-x86-64.zip"
    ],
    "otherPatchFileNames":[
        "p33192662_122010_Linux-x86-64.zip",
        "p29213893_122010_Generic.zip",
        "p28730253_122010_Linux-x86-64.zip",
        "p26352615_12201211019DBOCT2021RU_Linux-x86-64.zip",
        "p23614158_122010_Linux-x86-64.zip",
        "p24701840_122010_Linux-x86-64.zip",
        "p25173124_122010_Linux-x86-64.zip",
        "p25881255_122010_Linux-x86-64.zip",
        "p27015449_122010_Linux-x86-64.zip",
        "p28125601_122010_Linux-x86-64.zip",
        "p28852325_122010_Linux-x86-64.zip",
```

```
        "p29997937_122010_Linux-x86-64.zip",
        "p31335037_122010_Linux-x86-64.zip",
        "p32327201_122010_Linux-x86-64.zip",
        "p32327208_122010_Generic.zip"
    ],
    "installationParameters": {
        "unixGroupName": "dba",
        "unixGroupId": 12345,
        "unixUname": "oracle",
        "unixUid": 12345,
        "oracleHome": "/home/oracle/oracle.12.2.0.1",
        "oracleBase": "/home/oracle"
    }
}
```

Example Esempio di manifesto CEV per Oracle Database 18c

Nell'esempio seguente per la PSU di ottobre 2021 per Oracle Database 18c, RDS Custom applica p32126855, p28730253, quindi p27539475 e così via. L'esempio imposta nuovi valori per l'utente e il gruppo UNIX, la home Oracle e la base Oracle.

```
{
    "mediaImportTemplateVersion":"2020-08-14",
    "databaseInstallationFileNames":[
        "V978967-01.zip"
    ],
    "opatchFileNames":[
        "p6880880_180000_Linux-x86-64.zip"
    ],
    "psuRuPatchFileNames":[
        "p32126855_180000_Linux-x86-64.zip"
    ],
    "otherPatchFileNames":[
        "p28730253_180000_Linux-x86-64.zip",
        "p27539475_1813000DBRU_Linux-x86-64.zip",
        "p29213893_180000_Generic.zip",
        "p29374604_1813000DBRU_Linux-x86-64.zip",
        "p29782284_180000_Generic.zip",
        "p28125601_180000_Linux-x86-64.zip",
        "p28852325_180000_Linux-x86-64.zip",
        "p29997937_180000_Linux-x86-64.zip",
        "p31335037_180000_Linux-x86-64.zip",
        "p31335142_180000_Generic.zip"
```

```
    ]
    "installationParameters": {
        "unixGroupName": "dba",
        "unixGroupId": 12345,
        "unixUname": "oracle",
        "unixUid": 12345,
        "oracleHome": "/home/oracle/18.0.0.0.ru-2020-10.rur-2020-10.r1",
        "oracleBase": "/home/oracle/"
    }
}
```

Example Esempio di manifesto CEV per Oracle Database 19c

Nell'esempio seguente per Oracle Database 19c, RDS Custom applica p32126828, quindi p29213893, quindi p29782284 e così via. L'esempio imposta nuovi valori per l'utente e il gruppo UNIX, la home Oracle e la base Oracle.

```
{
    "mediaImportTemplateVersion": "2020-08-14",
    "databaseInstallationFileNames": [
        "V982063-01.zip"
    ],
    "opatchFileNames": [
        "p6880880_190000_Linux-x86-64.zip"
    ],
    "psuRuPatchFileNames": [
        "p32126828_190000_Linux-x86-64.zip"
    ],
    "otherPatchFileNames": [
        "p29213893_1910000DBRU_Generic.zip",
        "p29782284_1910000DBRU_Generic.zip",
        "p28730253_190000_Linux-x86-64.zip",
        "p29374604_1910000DBRU_Linux-x86-64.zip",
        "p28852325_190000_Linux-x86-64.zip",
        "p29997937_190000_Linux-x86-64.zip",
        "p31335037_190000_Linux-x86-64.zip",
        "p31335142_190000_Generic.zip"
    ],
    "installationParameters": {
        "unixGroupName": "dba",
        "unixGroupId": 12345,
        "unixUname": "oracle",
        "unixUid": 12345,
```

```
        "oracleHome": "/home/oracle/oracle.19.0.0.0.ru-2020-04.rur-2020-04.r1.EE.1",
        "oracleBase": "/home/oracle"
    }
}
```

Fase 6 (facoltativo): convalida del manifesto CEV

Facoltativamente, verificare che manifest sia un file JSON valido eseguendo lo script Python `json.tool`. Ad esempio, se si passa alla directory contenente un manifest CEV denominato `manifest.json`, esegui il comando riportato di seguito.

```
python -m json.tool < manifest.json
```

Fase 7: aggiunta delle autorizzazioni IAM necessarie

Verifica che il principale IAM che crea il CEV disponga delle policy necessarie descritte in Passaggio 5: concedi le autorizzazioni necessarie al tuo utente o ruolo IAM.

Creazione di un CEV

È possibile creare un CEV utilizzando AWS Management Console o il AWS CLI. Specifica l'architettura multilocazione o non multilocazione. Per ulteriori informazioni, consulta Considerazioni sull'architettura multilocazione.

In genere, la creazione di un CEV richiede circa due ore. Dopo aver creato la versione del motore personalizzato (CEV), puoi usare la CEV per creare un'istanza database RDS Custom. Per ulteriori informazioni, consulta Creazione di un'istanza database RDS Custom per Oracle.

Tieni presente i seguenti requisiti e limitazioni per la creazione di un CEV:

- Il bucket Amazon S3 contenente i file di installazione deve trovarsi nello stesso Regione AWS file del tuo CEV. In caso contrario, il processo di creazione fallisce.
- Il nome CEV deve essere nel formato seguente*major-engine-version*.*customized_string*. `19.cdb_cev1`
- Il nome CEV deve contenere da 1 a 50 caratteri alfanumerici, trattini bassi o punti.
- Il nome CEV non può contenere punti consecutivi, come in. `19..cdb_cev1`

Console

Per creare un CEV

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel pannello di navigazione, scegliere Versioni motore personalizzate.

   La pagina Versioni motore personalizzate mostra tutti i CEV attualmente esistenti. Se non è stato creato alcun CEV, la pagina è vuota.
3. Scegliere Creazione della versione del motore personalizzata.
4. In Opzioni motore, procedi nel modo seguente:

   a. Per Engine type (Tipo di motore), seleziona Oracle.

   b. Per le impostazioni dell'architettura, scegli facoltativamente Architettura multitenant per creare un CEV multitenant Oracle, che utilizza il motore DB o. `custom-oracle-ee-cdb` `custom-oracle-se2-cdb` È possibile creare un RDS Custom per Oracle solo con una CEV multi-tenant. Se non scegli questa opzione, il tuo CEV non è un CDB, che utilizza il motore o. `custom-oracle-ee` `custom-oracle-se2`

      > **Note**
      >
      > L'architettura selezionata è una caratteristica permanente della CEV. Non puoi modificare la CEV in modo che utilizzi un'architettura diversa in un secondo momento.

   c. Seleziona una delle seguenti opzioni:

      - Crea un nuovo CEV: crea una nuova versione del motore personalizzato da zero. In questo caso, è necessario specificare un manifesto JSON che definisca i file binari del database.
      - Crea un CEV dall'origine: in Specifica il CEV da copiare scegli una versione del motore personalizzato esistente da utilizzare come CEV di origine. In questo caso, puoi specificare una nuova Amazon Machine Image (AMI), ma non puoi definire file binari del database diversi.

   d. In Versione principale, scegli la versione principale del motore.
5. In Dettagli versione, procedi come segue:

a. Inserisci un nome valido in Nome della versione del motore personalizzato. Ad esempio, è possibile inserire il nome **`19.cdb_cev1`**.

b. (Facoltativo) Inserisci una descrizione per la CEV.

6. In Media di installazione, esegui le operazioni indicate di seguito:

a. (Facoltativo) Lascia il campo ID AMI vuoto per utilizzare l'AMI più recente fornita dal servizio oppure specifica l'AMI che hai usato in precedenza per creare una versione del motore personalizzato. Per ottenere ID AMI validi, usa una delle seguenti tecniche:

- Nella console, scegli Versioni del motore personalizzato nel riquadro di navigazione a sinistra e scegli il nome di una CEV. L'ID AMI utilizzato dalla CEV viene visualizzato nella scheda Configurazione.
- Nel AWS CLI, usa il comando. `describe-db-engine-versions` Cerca nell'output per `ImageID`.

b. Per la posizione S3 dei file manifest, inserisci la posizione del bucket Amazon S3 specificata in Fase 3: caricamento dei file di installazione in Amazon S3. Ad esempio, specifica **`s3://my-custom-installation-files/123456789012/cev1/`**.

> **Note**
>
> Il Regione AWS file in cui crei il CEV deve trovarsi nella stessa regione del bucket S3.

c. (Solo Crea un nuovo CEV) In Manifest CEV inserisci il manifesto JSON creato in Creazione del manifest CEV.

7. Nella sezione chiave KMS, seleziona Inserisci una chiave ARN per elencare le chiavi disponibili. AWS KMS Selezionare quindi la propria chiave KMS dall'elenco.

   È richiesta una AWS KMS chiave per RDS Custom. Per ulteriori informazioni, consulta Fase 1: creazione o riutilizzo di una chiave AWS KMS di crittografia simmetrica.

8. (Facoltativo) Scegli Aggiungi nuovo tag per creare una coppia chiave-valore per la tua CEV.

9. Scegliere Creazione della versione del motore personalizzata.

   Se il formato del manifesto JSON non è valido, la console visualizza Error validating the CEV manifest. Risolvi i problemi e riprova.

La pagina Versioni motore personalizzate viene visualizzata. Il tuo CEV viene mostrato con lo stato Creating (Creazione). Il processo di creazione della versione del motore personalizzato richiede circa due ore.

AWS CLI

Per creare un CEV utilizzando AWS CLI, esegui il comando create-custom-db-engine-version.

Sono richieste le seguenti opzioni:

- `--engine`— Specificare il tipo di motore. Per un CDB, specificare `custom-oracle-ee-cdb` o`custom-oracle-se2-cdb`. Per un non-CDB, specifica o. `custom-oracle-ee` `custom-oracle-se2` È possibile creare CDB solo da un CEV creato con o. `custom-oracle-ee-cdb` `custom-oracle-se2-cdb` È possibile creare documenti non CDB solo da un CEV creato con o. `custom-oracle-ee` `custom-oracle-se2`
- `--engine-version`: specifica la versione del motore. Il formato è. *`major-engine-version stringa_personalizzata`*. Il nome CEV deve contenere da 1 a 50 caratteri alfanumerici, trattini bassi o punti. Il nome CEV non può contenere punti consecutivi, come in. `19..cdb_cev1`
- `--kms-key-id`— Specificare un AWS KMS key.
- `--manifest`: specifica *`manifest_json_string`* o `--manifest file:`*`file_name`*. I caratteri di nuova riga non sono consentiti in *`manifest_json_string`*. Assicurati di evitare le doppie virgolette (") nel codice JSON precedendole con il backslash (\).

  L'esempio seguente mostra il *`manifest_json_string`* per 19c da Fase 5: preparazione del manifesto CEV. L'esempio imposta nuovi valori per la base Oracle, la home Oracle e l'ID e il nome dell'utente e del gruppo UNIX/Linux. Se copi questa stringa, rimuovi tutti i caratteri di nuova riga prima di incollarla nel comando.

```
"{\"mediaImportTemplateVersion\": \"2020-08-14\",
\"databaseInstallationFileNames\": [\"V982063-01.zip\"],
\"opatchFileNames\": [\"p6880880_190000_Linux-x86-64.zip\"],
\"psuRuPatchFileNames\": [\"p32126828_190000_Linux-x86-64.zip\"],
\"otherPatchFileNames\": [\"p29213893_1910000DBRU_Generic.zip\",
\"p29782284_1910000DBRU_Generic.zip\",\"p28730253_190000_Linux-
x86-64.zip\",\"p29374604_1910000DBRU_Linux-x86-64.zip\",
\"p28852325_190000_Linux-x86-64.zip\",\"p29997937_190000_Linux-x86-64.zip
\",\"p31335037_190000_Linux-x86-64.zip\",\"p31335142_190000_Generic.zip
\"]\"installationParameters\":{ \"unixGroupName\":\"dba\",
```

```
\ \"unixUname\":\"oracle\", \ \"oracleHome\":\"/home/oracle/
oracle.19.0.0.0.ru-2020-04.rur-2020-04.r1.EE.1\", \ \"oracleBase\":\"/
home/oracle/\"}}"
```

- `--database-installation-files-s3-bucket-name`: specifica lo stesso nome del bucket specificato in Fase 3: caricamento dei file di installazione in Amazon S3. Il Regione AWS bucket in cui viene eseguito `create-custom-db-engine-version` deve trovarsi nella stessa regione del bucket Amazon S3.

È anche possibile specificare le seguenti opzioni:

- `--description`: specifica una descrizione della CEV.
- `--database-installation-files-s3-prefix`: specifica il nome della cartella specificato in Fase 3: caricamento dei file di installazione in Amazon S3.
- `--image-id`: specifica un ID AMI che si desidera riutilizzare. Per trovare ID validi, esegui il comando `describe-db-engine-versions`, quindi cerca l'output per `ImageID`. Per impostazione predefinita, RDS Custom per Oracle utilizza l'AMI disponibile più recente.

L'esempio seguente crea una CEV multitenant Oracle denominata `19.cdb_cev1`. L'esempio riutilizza un'AMI esistente anziché utilizzare l'ultima AMI disponibile. Assicurati che il nome del tuo CEV inizi con il numero di versione principale del motore.

Example

PerLinux, omacOS: Unix

```
aws rds create-custom-db-engine-version \
    --engine custom-oracle-se2-cdb \
    --engine-version 19.cdb_cev1 \
    --database-installation-files-s3-bucket-name us-east-1-123456789012-custom-
installation-files \
    --database-installation-files-s3-prefix 123456789012/cev1 \
    --kms-key-id my-kms-key \
    --description "test cev" \
    --manifest manifest_string \
    --image-id ami-012a345678901bcde
```

Per Windows:

```
aws rds create-custom-db-engine-version ^
    --engine custom-oracle-se2-cdb ^
    --engine-version 19.cdb_cev1 ^
    --database-installation-files-s3-bucket-name us-east-1-123456789012-custom-
installation-files ^
    --database-installation-files-s3-prefix 123456789012/cev1 ^
    --kms-key-id my-kms-key ^
    --description "test cev" ^
    --manifest manifest_string ^
    --image-id ami-012a345678901bcde
```

Example

Ottenere informazioni sul CEV tramite il comando `describe-db-engine-versions`.

```
aws rds describe-db-engine-versions \
    --engine custom-oracle-se2-cdb \
    --include-all
```

Il seguente output parziale mostra il motore, i gruppi di parametri, il manifesto e altre informazioni.

```
{
    "DBEngineVersions": [
        {
            "Engine": "custom-oracle-se2-cdb",
            "EngineVersion": "19.cdb_cev1",
            "DBParameterGroupFamily": "custom-oracle-se2-cdb-19",
            "DBEngineDescription": "Containerized Database for Oracle Custom SE2",
            "DBEngineVersionDescription": "test cev",
            "Image": {
                "ImageId": "ami-012a345678901bcde",
                "Status": "active"
            },
            "ValidUpgradeTarget": [],
            "SupportsLogExportsToCloudwatchLogs": false,
            "SupportsReadReplica": true,
            "SupportedFeatureNames": [],
            "Status": "available",
            "SupportsParallelQuery": false,
            "SupportsGlobalDatabases": false,
            "MajorEngineVersion": "19",
            "DatabaseInstallationFilesS3BucketName": "us-east-1-123456789012-custom-
installation-files",
```

```
            "DatabaseInstallationFilesS3Prefix": "123456789012/cev1",
            "DBEngineVersionArn": "arn:aws:rds:us-east-1:123456789012:cev:custom-
oracle-se2-cdb/19.cdb_cev1/abcd12e3-4f5g-67h8-i9j0-k1234l56m789",
            "KMSKeyId": "arn:aws:kms:us-
east-1:732027699161:key/1ab2345c-6d78-9ef0-1gh2-3456i7j89k01",
            "CreateTime": "2023-03-07T19:47:58.131000+00:00",
            "TagList": [],
            "SupportsBabelfish": false,
...
```

Impossibile creare un CEV

Se il processo di creazione di un CEV non riesce, RDS Custom emette `RDS-EVENT-0198` con il messaggio `Creation failed for custom engine version` *`major-engine-version.cev_name`* e include i dettagli sull'errore. Ad esempio, l'evento stampa i file mancanti.

Non è possibile modificare un CEV fallito. È possibile solamente eliminarlo, quindi riprovare a creare un CEV dopo aver risolto le cause dell'errore. Per informazioni sulla risoluzione dei problemi relativi all'errore di creazione del CEV, consulta Risoluzione dei problemi relativi alla creazione di versioni personalizzate del motore per RDS Custom per Oracle.

Modifica dello stato del CEV

È possibile modificare un CEV utilizzando AWS Management Console o il AWS CLI. È possibile modificare la descrizione CEV o il relativo stato di disponibilità. Il CEV ha uno dei seguenti valori di stato:

- `available` – È possibile utilizzare questo CEV per creare una nuova istanza database RDS Custom o aggiornare un'istanza database. Questo è lo stato predefinito per un CEV appena creato.
- `inactive` – Non è possibile creare o aggiornare un'istanza RDS Custom con questo CEV. Non è possibile ripristinare una snapshot DB per creare una nuova istanza database RDS Custom con questo CEV.

È possibile modificare il CEV da qualsiasi stato supportato a qualsiasi altro stato supportato. È possibile modificare lo stato per impedire l'uso accidentale di un CEV o rendere nuovamente idoneo l'uso di un CEV sospeso. Ad esempio, puoi modificare lo stato del tuo CEV da `available` a `inactive`, nonché da `inactive` tornare a `available`.

Console

Per modificare un CEV

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel pannello di navigazione, scegliere Versioni motore personalizzate.
3. Scegliere un CEV di cui si desidera modificare la descrizione o lo stato.
4. Per Operazioni, scegli Modifica.
5. Effettua una qualsiasi delle seguenti modifiche:
   - Per CEV status settings (Impostazioni dello stato del CEV) scegliere un nuovo stato di disponibilità.
   - In Version description (Descrizione versione), inserire una nuova descrizione.
6. Scegliere Modify CEV (Modifica CEV).

   Se il CEV è in uso, la console visualizza You can't modify the CEV status (Non puoi modificare lo stato CEV). Risolvi i problemi e riprova.

La pagina Versioni motore personalizzate viene visualizzata.

AWS CLI

Per modificare un CEV utilizzando il AWS CLI, esegui il comando modify-custom-db-engine-version. È possibile trovare i CEV da modificare eseguendo il comando. describe-db-engine-versions

Sono richieste le seguenti opzioni:

- `--engine` *`engine-type`*, dove il *`tipo di motore è`*,, o `custom-oracle-ee` `custom-oracle-se2` `custom-oracle-ee-cdb` `custom-oracle-se2-cdb`
- `--engine-version` *`cev`*, dove *`cev`* è il nome della versione del motore personalizzata che si desidera modificare
- `--status` *`status`*, dove *`status`* è lo stato di disponibilità che si desidera assegnare al CEV

L'esempio seguente cambia un CEV denominato `19.my_cev1` dal suo stato attuale a `inactive`.

Example

PerLinux, o: macOS Unix

```
aws rds modify-custom-db-engine-version \
    --engine custom-oracle-se2 \
    --engine-version 19.my_cev1 \
    --status inactive
```

Per Windows:

```
aws rds modify-custom-db-engine-version ^
    --engine custom-oracle-se2 ^
    --engine-version 19.my_cev1 ^
    --status inactive
```

Visualizzazione dei dettagli della CEV

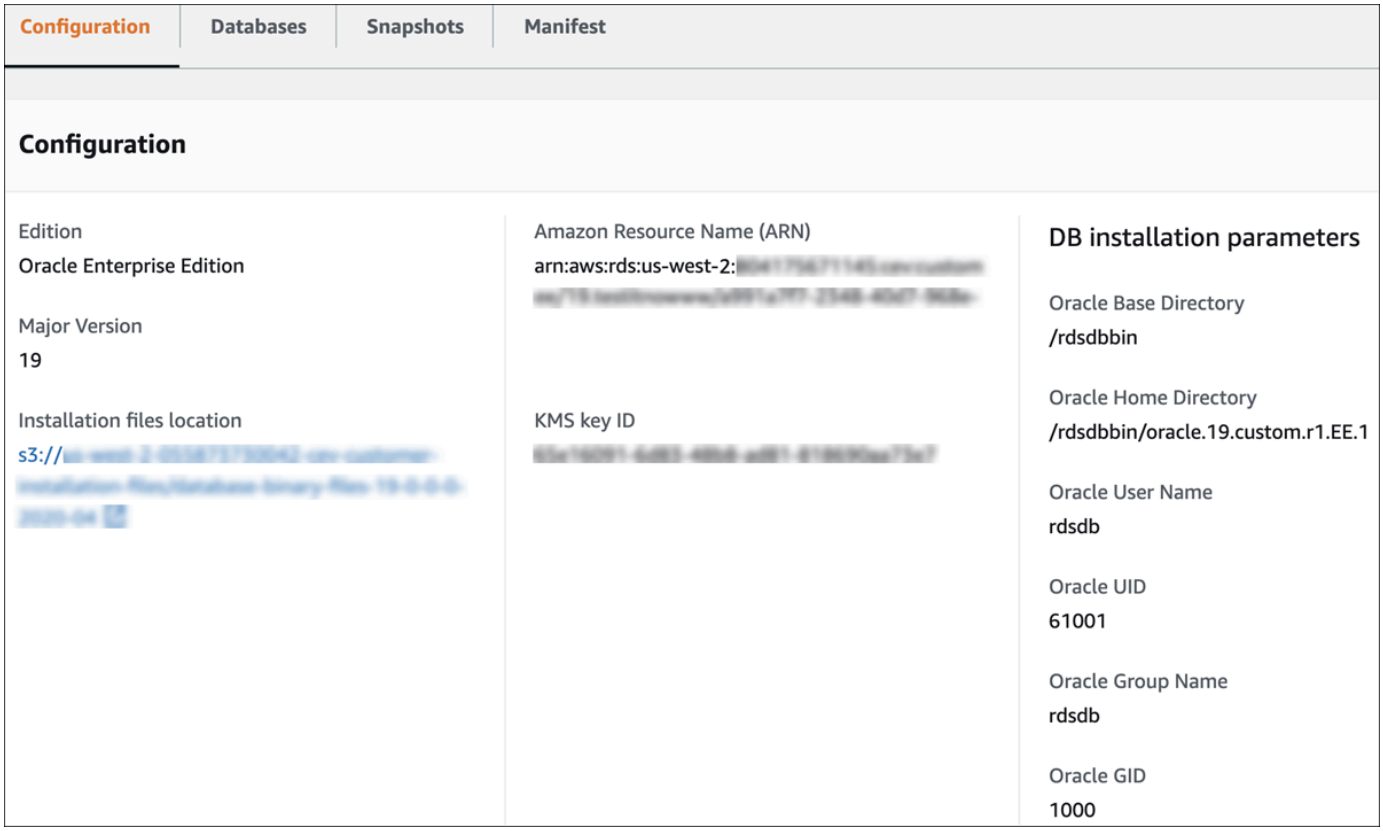
È possibile visualizzare i dettagli sul manifesto CEV e sul comando utilizzato per creare il proprio CEV utilizzando AWS Management Console o il. AWS CLI

Console

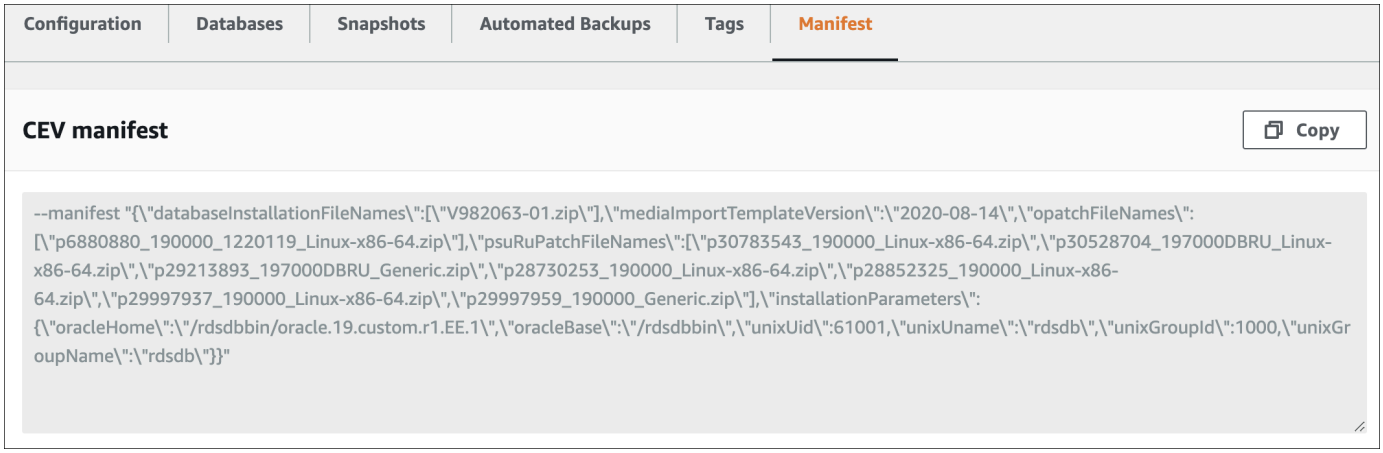
Per visualizzare i dettagli della CEV

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel pannello di navigazione, scegliere Versioni motore personalizzate.

   La pagina Versioni motore personalizzate mostra tutti i CEV attualmente esistenti. Se non è stato creato alcun CEV, la pagina è vuota.
3. Seleziona il nome della CEV che vuoi visualizzare.
4. Scegli Configuration (Configurazione) per visualizzare i parametri di installazione specificati nel manifesto.



5. Scegli Manifest (Manifesto) per visualizzare i parametri di installazione specificati nell'opzione `--manifest` del comando `create-custom-db-engine-version`. È possibile copiare questo testo, sostituire i valori in base alle esigenze e utilizzarli in un nuovo comando.



AWS CLI

Per visualizzare i dettagli su un CEV utilizzando il AWS CLI, esegui il comando. [describe-db-engine-versions]

Sono richieste le seguenti opzioni:

- `--engine` *`engine-type`*, dove il *`tipo di motore`* è`custom-oracle-ee`,, o `custom-oracle-se2` `custom-oracle-ee-cdb` `custom-oracle-se2-cdb`
- `--engine-version` *`major-engine-version`*`.`*`customized_string`*

L'esempio seguente crea un CEV non CDB che utilizza Enterprise Edition. Il nome CEV `19.my_cev1` inizia con il numero di versione principale del motore, che è obbligatorio.

Example

Per LinuxmacOS, oUnix:

```
aws rds describe-db-engine-versions \
    --engine custom-oracle-ee \
    --engine-version 19.my_cev1
```

Per Windows:

```
aws rds describe-db-engine-versions ^
    --engine custom-oracle-ee ^
    --engine-version 19.my_cev1
```

Il seguente output parziale mostra il motore, i gruppi di parametri, il manifesto e altre informazioni.

```
"DBEngineVersions": [
    {
        "Engine": "custom-oracle-ee",
        "MajorEngineVersion": "19",
        "EngineVersion": "19.my_cev1",
        "DatabaseInstallationFilesS3BucketName": "us-east-1-123456789012-cev-customer-
installation-files",
        "DatabaseInstallationFilesS3Prefix": "123456789012/cev1",
        "CustomDBEngineVersionManifest": "{\n\"mediaImportTemplateVersion\":
 \"2020-08-14\",\n\"databaseInstallationFileNames\": [\n\"V982063-01.zip\"\n],
\n\"installationParameters\": {\n\"oracleBase\":\"/tmp\",\n\"oracleHome\":\"/
tmp/Oracle\"\n},\n\"opatchFileNames\": [\n\"p6880880_190000_Linux-x86-64.zip
\"\n],\n\"psuRuPatchFileNames\": [\n\"p32126828_190000_Linux-x86-64.zip
\"\n],\n\"otherPatchFileNames\": [\n\"p29213893_1910000DBRU_Generic.zip\",\n
\"p29782284_1910000DBRU_Generic.zip\",\n\"p28730253_190000_Linux-x86-64.zip\",\n
\"p29374604_1910000DBRU_Linux-x86-64.zip\",\n\"p28852325_190000_Linux-x86-64.zip\",
```

```
\n\"p29997937_190000_Linux-x86-64.zip\",\n\"p31335037_190000_Linux-x86-64.zip\",\n
\"p31335142_190000_Generic.zip\"\n]\n}\n",
        "DBParameterGroupFamily": "custom-oracle-ee-19",
        "DBEngineDescription": "Oracle Database server EE for RDS Custom",
        "DBEngineVersionArn": "arn:aws:rds:us-west-2:123456789012:cev:custom-oracle-
ee/19.my_cev1/0a123b45-6c78-901d-23e4-5678f901fg23",
        "DBEngineVersionDescription": "test",
        "KMSKeyId": "arn:aws:kms:us-east-1:123456789012:key/ab1c2de3-f4g5-6789-h012-
h3ijk4567l89",
        "CreateTime": "2022-11-18T09:17:07.693000+00:00",
        "ValidUpgradeTarget": [
        {
            "Engine": "custom-oracle-ee",
            "EngineVersion": "19.cev.2021-01.09",
            "Description": "test",
            "AutoUpgrade": false,
            "IsMajorVersionUpgrade": false
        }
]
```

Eliminazione di un CEV

È possibile eliminare un CEV utilizzando AWS Management Console o il AWS CLI. In genere, l'eliminazione richiede pochi minuti.

Per eliminare un CEV, non può essere utilizzato da nessuno dei seguenti elementi:

- Un'istanza database RDS Custom
- Una snapshot di un'istanza database RDS Custom
- Backup automatico dell'istanza database RDS Custom

Console

Per eliminare un CEV

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel pannello di navigazione, scegliere Versioni motore personalizzate.
3. Scegliere un CEV di cui si desidera eliminare la descrizione o lo stato.
4. In Actions (Azioni), scegliere Delete (Elimina).

Viene visualizzata la finestra di dialogo Delete ***cev_name***? (Elimina cev_name?).

5. Immettere **`delete me`**, quindi scegliere Delete (Elimina).

   Nella pagina Versioni motore personalizzate, il banner mostra che il tuo CEV è stato eliminato.

AWS CLI

Per eliminare un CEV utilizzando il AWS CLI, esegui il comando delete-custom-db-engine-version.

Sono richieste le seguenti opzioni:

- `--engine` *`engine-type`*, dove il *`tipo di motore è`*,, o `custom-oracle-ee` `custom-oracle-se2` `custom-oracle-ee-cdb` `custom-oracle-se2-cdb`
- `--engine-version` *`cev`*, in cui *`cev`* è il nome della versione del motore personalizzata da eliminare

L'esempio seguente elimina un CEV denominato `19.my_cev1`.

Example

PerLinux, o: macOS Unix

```
aws rds delete-custom-db-engine-version \
    --engine custom-oracle-ee \
    --engine-version 19.my_cev1
```

Per Windows:

```
aws rds delete-custom-db-engine-version ^
    --engine custom-oracle-ee ^
    --engine-version 19.my_cev1
```

Configurazione di un'istanza database per Amazon RDS Custom per Oracle

Puoi creare un'istanza database RDS Custom e quindi connetterti ad essa utilizzando Secure Shell (SSH) o AWS Systems Manager.

Argomenti

Considerazioni sull'architettura multilocazione

Se crei un'istanza database Amazon RDS Custom for Oracle con l'architettura Oracle multitenant (`custom-oracle-ee-cdb`o tipo di `custom-oracle-se2-cdb` motore), il database è un database container (CDB). Se non specifichi l'architettura multitenant Oracle, il tuo database è un database tradizionale non CDB che utilizza il tipo di motore or. `custom-oracle-ee` `custom-oracle-se2` Un non CDB non può contenere database collegabili (PDB). Per ulteriori informazioni, consulta Architettura dei database per Amazon RDS Custom per Oracle.

Quando crei un'istanza CDB RDS Custom per Oracle, considera quanto segue:

- Puoi creare un database multilocazione solo da una CEV Oracle Database 19c.
- È possibile creare un'istanza CDB solo se il CEV utilizza il tipo di motore o. `custom-oracle-ee-cdb` `custom-oracle-se2-cdb`
- Se crei un'istanza CDB utilizzando Standard Edition 2, il CDB può contenere un massimo di 3 PDB.
- Per impostazione predefinita, il CDB viene denominato `RDSCDB`, che è anche l'ID di sistema Oracle (Oracle SID). È possibile scegliere un nome diverso.
- Il tuo CDB contiene solo un PDB iniziale. Il nome predefinito del PDB è `ORCL`. Puoi scegliere un nome diverso per il PDB iniziale, ma il SID Oracle e il nome PDB non possono essere uguali.
- RDS Custom per Oracle non fornisce API per PDB. Per creare PDB aggiuntivi, utilizza il comando Oracle SQL `CREATE PLUGGABLE DATABASE`. RDS Custom per Oracle non limita il numero di

PDB che è possibile creare. In generale, sei responsabile della creazione e della gestione dei PDB, come in una implementazione on-premise.

- Non puoi utilizzare le API RDS per creare, modificare ed eliminare i PDB, usa invece le istruzioni SQL Oracle. Quando crei un PDB utilizzando Oracle SQL, ti consigliamo di scattare successivamente un'istantanea manuale nel caso in cui sia necessario eseguire il ripristino (PITR). point-in-time
- Non puoi rinominare i PDB esistenti utilizzando le API Amazon RDS. Inoltre, non è possibile rinominare il CDB utilizzando il comando `modify-db-instance`.
- La modalità aperta per la root CDB è `READ WRITE` sul database primario e `MOUNTED` su un database di standby montato. RDS Custom per Oracle tenta di aprire tutti i PDB all'apertura del CDB. Se RDS Custom per Oracle non è in grado di aprire tutti i PDB, genera l'evento `tenant database shutdown`.

Creazione di un'istanza database RDS Custom per Oracle

Crea un'istanza Amazon RDS Custom for Oracle DB utilizzando AWS Management Console o. AWS CLI La procedura è simile alla procedura per la creazione di un'istanza database Amazon RDS. Per ulteriori informazioni, consulta Creazione di un'istanza database Amazon RDS.

Se hai incluso i parametri di installazione nel manifesto CEV, l'istanza database utilizza la base Oracle, la home Oracle e l'ID e il nome dell'utente e del gruppo UNIX/Linux specificati. Il file `oratab`, creato da Oracle Database durante l'installazione, punta alla posizione di installazione reale anziché a un collegamento simbolico. RDS Custom per Oracle esegue i comandi come utente del sistema operativo configurato anziché come utente predefinito `rdsdb`. Per ulteriori informazioni, consulta Fase 5: preparazione del manifesto CEV.

Completa le attività presenti in Configurazione dell'ambiente per Amazon RDS Custom per Oracle prima di poter creare o connettere a un'istanza database RDS Custom.

Console

Per creare un'istanza database RDS Custom per Oracle

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, scegliere Databases (Database).
3. Scegliere Create database (Crea database).

4. In Choose a database creation method (Seleziona metodo di creazione del database), scegli Standard create (Creazione standard).
5. Nella sezione Opzioni motore, procedi nel modo seguente:

   a. Per Engine type (Tipo di motore), seleziona Oracle.

   b. Per il tipo di gestione del database, selezionare Amazon RDS Custom.

   c. In Impostazioni dell'architettura, effettua una delle seguenti operazioni:

      - Seleziona Architettura multi-tenant per creare un database container (CDB). Al momento della creazione, il CDB contiene un seed PDB e un PDB iniziale.

        > **Note**
        >
        > L'architettura multilocazione è supportata solo in Oracle Database 19c.

      - Deseleziona Architettura multi-tenant per creare un database non di tipo container. Un database non di tipo container non può contenere database collegabili (PDB).

   d. Per Edition, scegli Oracle Enterprise Edition o Oracle Standard Edition 2.

   e. In Versioni del motore personalizzato, scegli una versione del motore personalizzato (CEV) RDS Custom esistente. Una CEV ha il formato seguente: *`major-engine-version`*`.`*`customized_string`*. Un identificatore di esempio è `19.cdb_cev1`.

      Se hai scelto l'architettura Multitenant nel passaggio precedente, puoi specificare solo un CEV che utilizza il `custom-oracle-ee-cdb` tipo di motore o. `custom-oracle-se2-cdb` La console filtra i CEV creati con diversi tipi di motore.
6. Per Templates (Modelli), scegli Production (Produzione).
7. Nella sezione Rule settings (Impostazioni regole), procedi nel seguente modo:

   a. In Identificatore di istanza database, immetti un nome univoco per l'istanza database.

   b. In Nome utente master, immetti un nome utente. È possibile recuperare questo valore dalla console in un secondo momento.

      Quando ti connetti a un non CDB, l'utente master è l'utente del non CDB. Quando ti connetti a un CDB, l'utente master è l'utente del PDB. Per connetterti alla root CDB, accedi all'host, avvia un client SQL e crea un utente amministrativo con i comandi SQL.

   c. Deseleziona Genera automaticamente una password.
8. Scegli una classe in Classe di istanza database.

Per le classi supportate, consultare Supporto delle classi di istanza database per RDS Custom per Oracle.

9. Nella sezione Storage (Archiviazione), procedi come segue:

   a. In Tipo di storage, scegli un tipo di SSD: io1, gp2 o gp3. Sono disponibili le seguenti opzioni aggiuntive:

      - Per io1 o gp3, scegli una tariffa in Capacità di IOPS allocata. L'impostazione predefinita è 1000 per io1 e 12000 per gp3.
      - Per gp3, scegli una tariffa in Throughput di storage. L'impostazione predefinita è 500. MiBps

   b. In Storage allocato, scegli una dimensione di archiviazione. L'impostazione predefinita è 40 GiB.

10. In Connettività, specifica un valore nei campi Cloud privato virtuale (VPC), Gruppi di sottoreti DB e Gruppo di sicurezza VPC (firewall).

11. Per la sicurezza RDS Custom, procedere come segue:

    a. Per il profilo dell'istanza IAM, selezionare il profilo dell'istanza per l'istanza database RDS Custom per Oracle.

       Il profilo dell'istanza IAM deve iniziare con`AWSRDSCustom`, ad esempio *`AWSRDSCustomInstanceProfileForRdsCustomInstance`*.

    b. Per Crittografia, scegli Inserisci una chiave ARN per elencare le chiavi disponibili AWS KMS . Scegliere quindi la propria chiave dall'elenco.

       È richiesta una AWS KMS chiave per RDS Custom. Per ulteriori informazioni, consulta Fase 1: creazione o riutilizzo di una chiave AWS KMS di crittografia simmetrica.

12. In Opzioni database, esegui le operazioni indicate di seguito:

    a. (Facoltativo) In ID sistema ID (SID), inserisci un valore per il SID Oracle, che è anche il nome del tuo CDB. Il valore del campo ID sistema ID (SID) è il nome dell'istanza database Oracle che gestisce i file del database. In questo contesto, il termine "istanza database Oracle" si riferisce esclusivamente all'area globale del sistema (SGA) e ai processi in background di Oracle. Se non specifichi un valore, il valore predefinito è **`RDSCDB`**.

    b. (Facoltativo) In Nome database iniziale, immetti un nome. Il valore predefinito è **`ORCL`**. Nell'architettura multi-tenant, il nome del database iniziale è il nome del PDB.

> **Note**
>
> Il SID e il nome PDB devono essere diversi.

c. Per Gruppo di opzioni, scegliete un gruppo di opzioni o accettate quello predefinito.

> **Note**
>
> L'unica opzione supportata per RDS Custom for Oracle è`Timezone`. Per ulteriori informazioni, consulta Fuso orario Oracle.

d. In Periodo di conservazione dei backup, scegli un valore. Non puoi scegliere 0 giorni.

e. Per le restanti sezioni, specifica le impostazioni dell'istanza database RDS Custom preferite. Per informazioni su ciascuna impostazione, consulta Impostazioni per istanze database. Le impostazioni seguenti non appaiono nella console e non sono supportate:

- Caratteristiche processore
- Storage autoscaling (Auto Scaling dello storage)
- Opzione Password and Kerberos authentication (Password e autenticazione Kerberos) in Database authentication (Autenticazione del database) (solo Autenticazione password è supportata)
- Approfondimenti sulle prestazioni
- Log exports (Esportazioni log)
- Abilita aggiornamento automatico della versione secondaria
- Deletion protection (Protezione da eliminazione)

13. Scegliere Crea database.

> **Important**
>
> Quando crei un'istanza database RDS Custom per Oracle, potresti ricevere il seguente errore: Il ruolo collegato ai servizi è nel processo di creazione. Riprova più tardi. In questo caso, attendere alcuni minuti e riprovare a creare l'istanza database.

Il pulsante View credential details (Vedi dettagli delle credenziali) viene visualizzato sulla pagina Database.

Per vedere nome utente e password per l'istanza database RDS Custom, seleziona View credential details (Vedi dettagli delle credenziali).

Per connetterti all'istanza database come utente principale, utilizza il nome utente e la password visualizzati.

> **Important**
>
> Non è possibile visualizzare di nuovo la password dell'utente master nella console. Se non la registri, potresti doverla modificare. Per modificare la password dell'utente master dopo che l'istanza database RDS Custom è disponibile, accedi al database ed esegui un comando `ALTER USER`. Puoi ripristinare la password utilizzando l'opzione Modifica nella console.

14. Scegliere Database per visualizzare l'elenco delle istanze database RDS Custom.
15. Scegliere l'istanza database RDS Custom appena creata.

    Nella console RDS vengono visualizzati i dettagli per la nuova istanza database RDS Custom:

    - L'istanza database RDS Custom rimane nello stato creating (creazione in corso) fino a quando non è stata creata e non è pronta per l'uso. Quando lo stato cambia in available (disponibile) è possibile connettersi all'istanza database. A seconda della classe di istanza e dello storage allocato, potrebbero trascorrere diversi minuti prima che la nuova istanza database sia disponibile.
    - Ruolo ha il valore Istanza (RDS Custom).
    - Modalità di automazione RDS Custom ha il valore Automazione completa. Questa impostazione indica che l'istanza database fornisce il monitoraggio automatico e il ripristino dell'istanza.

AWS CLI

È possibile creare un'istanza DB personalizzata RDS utilizzando il create-db-instance AWS CLI comando.

Sono richieste le seguenti opzioni:

- `--db-instance-identifier`
- `--db-instance-class` (per l'elenco delle classi di istanza supportate, vedere Supporto delle classi di istanza database per RDS Custom per Oracle)
- `--engine` *`engine-type`*, dove il *`tipo di motore`* è`custom-oracle-ee`,, o `custom-oracle-se2` `custom-oracle-ee-cdb` `custom-oracle-se2-cdb`
- `--engine-version` *`cev`* (dove *`cev`* è il nome della versione del motore personalizzata specificata in Creazione di un CEV)
- `--kms-key-id` *`my-kms-key`*
- `--backup-retention-period` *`days`* (dove *`days`* è un valore maggiore di `0`)
- `--no-auto-minor-version-upgrade`
- `--custom-iam-instance-profile AWSRDSCustomInstanceProfile-`*`us-east-1`* (dove *`region`* è la Regione AWS in cui stai creando l'istanza DB)

Nell'esempio seguente viene creata un'istanza database RDS Custom per denominata `my-cfo-cdb-instance`. Il database è un CDB con il nome non predefinito *`MYCDB`*. Il nome non predefinito del PDB è *`MYPDB`*. Il periodo di retention dei backup è di tre giorni.

Example

PerLinux, o: macOS Unix

```
aws rds create-db-instance \
    --engine custom-oracle-ee-cdb \
    --db-instance-identifier my-cfo-cdb-instance \
    --engine-version 19.cdb_cev1 \
    --db-name MYPDB \
    --db-system-id MYCDB \
    --allocated-storage 250 \
    --db-instance-class db.m5.xlarge \
    --db-subnet-group mydbsubnetgroup \
    --master-username myuser \
    --master-user-password mypassword \
    --backup-retention-period 3 \
    --port 8200 \
    --kms-key-id my-kms-key \
    --no-auto-minor-version-upgrade \
```

```
    --custom-iam-instance-profile AWSRDSCustomInstanceProfile-us-east-1
```

Per Windows:

```
aws rds create-db-instance ^
    --engine custom-oracle-ee-cdb ^
    --db-instance-identifier my-cfo-cdb-instance ^
    --engine-version 19.cdb_cev1 ^
    --db-name MYPDB ^
    --db-system-id MYCDB ^
    --allocated-storage 250 ^
    --db-instance-class db.m5.xlarge ^
    --db-subnet-group mydbsubnetgroup ^
    --master-username myuser ^
    --master-user-password mypassword ^
    --backup-retention-period 3 ^
    --port 8200 ^
    --kms-key-id my-kms-key ^
    --no-auto-minor-version-upgrade ^
    --custom-iam-instance-profile AWSRDSCustomInstanceProfile-us-east-1
```

> **Note**
>
> Specifica una password diversa dal prompt mostrato qui come best practice per la sicurezza.

Ottenere informazioni sull'istanza tramite il comando `describe-db-instances`.

Example

```
aws rds describe-db-instances --db-instance-identifier my-cfo-cdb-instance
```

Il seguente output parziale mostra il motore, i gruppi di parametri e altre informazioni.

```
        {
            "DBInstanceIdentifier": "my-cfo-cdb-instance",
            "DBInstanceClass": "db.m5.xlarge",
            "Engine": "custom-oracle-ee-cdb",
            "DBInstanceStatus": "available",
            "MasterUsername": "admin",
            "DBName": "MYPDB",
```

```
            "DBSystemID": "MYCDB",
            "Endpoint": {
                "Address": "my-cfo-cdb-instance.abcdefghijkl.us-
east-1.rds.amazonaws.com",
                "Port": 1521,
                "HostedZoneId": "A1B2CDEFGH34IJ"
            },
            "AllocatedStorage": 100,
            "InstanceCreateTime": "2023-04-12T18:52:16.353000+00:00",
            "PreferredBackupWindow": "08:46-09:16",
            "BackupRetentionPeriod": 7,
            "DBSecurityGroups": [],
            "VpcSecurityGroups": [
                {
                    "VpcSecurityGroupId": "sg-0a1bcd2e",
                    "Status": "active"
                }
            ],
            "DBParameterGroups": [
                {
                    "DBParameterGroupName": "default.custom-oracle-ee-cdb-19",
                    "ParameterApplyStatus": "in-sync"
                }
            ],
...
```

Ruolo collegato ai servizi RDS Custom

Un ruolo collegato al servizio offre ad Amazon RDS Custom l'accesso alle risorse del tuo Account AWS Ciò rende più semplice l'utilizzo di RDS Custom perché non si devono aggiungere manualmente le autorizzazioni necessarie. RDS Custom definisce le autorizzazioni dei ruoli associato ai servizi e, salvo diversamente definito, solo RDS Custom può assumere tali ruoli. Le autorizzazioni definite includono la policy di attendibilità e la policy delle autorizzazioni che non può essere collegata a nessun'altra entità IAM.

Quando crei un'istanza DB personalizzata RDS, vengono creati e utilizzati sia i ruoli collegati ai servizi (se non già esistenti) Amazon RDS che RDS Custom. Per ulteriori informazioni, consulta Utilizzo di ruoli collegati ai servizi per Amazon RDS.

La prima volta che crei un'istanza database RDS Custom per Oracle, potresti ricevere il seguente errore: Il ruolo collegato ai servizi è nel processo di creazione. Riprova più tardi. In questo caso, attendere alcuni minuti e riprovare a creare l'istanza database.

Connessione all'istanza database RDS Custom utilizzando Session Manager

Dopo aver creato l'istanza DB personalizzata RDS, puoi connetterti ad essa utilizzando. AWS Systems Manager Session Manager Questa è la tecnica preferita quando l'istanza database non è accessibile pubblicamente.

Session Manager consente di accedere alle istanze Amazon EC2 tramite una shell (interprete di comandi) basata su browser o tramite la AWS CLI. Per ulteriori informazioni, consulta AWS Systems Manager Session Manager.

Console

Per connettersi all'istanza database utilizzando Session Manager

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, scegliere Databases (Database) e quindi scegliere l'istanza database RDS Custom a cui desideri connetterti.
3. Scegliere Configuration (Configurazione).
4. Annota Resource ID (Risorsa ID)per l'istanza database. Ad esempio, l'ID risorsa potrebbe essere `db-ABCDEFGHIJKLMNOPQRS0123456`.
5. Apri la console Amazon EC2 all'indirizzo https://console.aws.amazon.com/ec2/.
6. Nel riquadro di navigazione, seleziona Istanze.
7. Cerca il nome dell'istanza EC2, quindi fai clic sull'ID istanza associato con esso. Ad esempio, l'istanza ID potrebbe essere `i-abcdefghijklm01234`.
8. Scegli Connetti.
9. Scegli Session Manager.
10. Scegli Connetti.

    Si apre una finestra per la sessione.

AWS CLI

Puoi connettere l'istanza database RDS Custom tramite AWS CLI. Questa tecnica richiede il plugin Session Manager per AWS CLI. Per informazioni su come installare il plugin, consultare Installare il plugin di Session Manager per AWS CLI.

Per trovare l'ID della risorsa DB dell'istanza database RDS Custom, utilizzare `aws rds describe-db-instances`.

```
aws rds describe-db-instances \
    --query 'DBInstances[*].[DBInstanceIdentifier,DbiResourceId]' \
    --output text
```

Il seguente output di esempio mostra l'ID della risorsa per l'istanza RDS Custom. Il prefisso è `db-`.

```
db-ABCDEFGHIJKLMNOPQRS0123456
```

Per trovare l'ID dell'istanza EC2 della tua istanza database, utilizzare `aws ec2 describe-instances`. Nell'esempio seguente viene utilizzato `db-ABCDEFGHIJKLMNOPQRS0123456` per l'ID risorsa.

```
aws ec2 describe-instances \
    --filters "Name=tag:Name,Values=db-ABCDEFGHIJKLMNOPQRS0123456" \
    --output text \
    --query 'Reservations[*].Instances[*].InstanceId'
```

L'output di esempio seguente mostra l'ID dell'istanza EC2.

```
i-abcdefghijklm01234
```

Utilizzo del comando `aws ssm start-session`, che fornisce l'ID istanza EC2 nel parametro `--target`.

```
aws ssm start-session --target "i-abcdefghijklm01234"
```

Se l'operazione riesce, la connessione sarà simile al seguente.

```
Starting session with SessionId: yourid-abcdefghijklm1234
[ssm-user@ip-123-45-67-89 bin]$
```

Connessione all'istanza database RDS Custom tramite SSH

Secure Shell Protocol (SSH) è un protocollo di rete che supporta la comunicazione crittografata su una rete non protetta. Dopo aver creato l'istanza database RDS Custom, è possibile connettersi

utilizzando un client SSH. Per ulteriori informazioni, consultare Connessione all'istanza Linux tramite SSH.

La connessione SSH dipende dal fatto che l'istanza DB è privata, ovvero l'istanza non accetta connessioni dalla rete Internet pubblica. In questo caso, è necessario utilizzare il tunneling SSH per connettere l'utilità ssh alla propria istanza. Questa tecnica trasporta i dati con un flusso di dati dedicato (tunnel) all'interno di una sessione SSH esistente. È possibile configurare il tunneling SSH utilizzando AWS Systems Manager.

> **Note**
>
> Sono supportate varie strategie per accedere alle istanze private. Per scoprire come connettere un client ssh a istanze private utilizzando gli host bastione, consulta Host bastione Linux su AWS. Per informazioni su come configurare la funzionalità di inoltro alla porta, consulta l'argomento relativo all'inoltro alla porta tramite AWS Systems Manager Session Manager.

Se l'istanza DB si trova in una sottorete pubblica e per tale istanza è stata abilitata l'opzione Disponibile pubblicamente, non è richiesto il tunneling SSH. Puoi connetterti tramite SSH con la stessa procedura usata per un'istanza Amazon EC2 pubblica.

Per connettere un client SSH all'istanza database, completa la procedura riportata di seguito:

1. Fase 1: configurazione dell'istanza database per consentire connessioni SSH
2. Fase 2: recupero della chiave segreta SSH e l'ID dell'istanza EC2
3. Fase 3: connessione all'istanza EC2 utilizzando l'utility ssh

Fase 1: configurazione dell'istanza database per consentire connessioni SSH

Per assicurarti che l'istanza database accetti connessioni SSH, procedi nel modo seguente:

- Assicurati che il gruppo di sicurezza dell'istanza database consenta le connessioni in entrata sulla porta 22 per TCP.

  Per informazioni su come configurare il gruppo di sicurezza per l'istanza DB, consulta Controllo dell'accesso con i gruppi di sicurezza.
- Se non prevedi di utilizzare il tunneling SSH, assicurati che l'istanza DB risieda in una sottorete pubblica e che sia accessibile al pubblico.

Nella console, il campo pertinente è Disponibile pubblicamente nella scheda Connettività e sicurezza della pagina dei dettagli del database. Per controllare le impostazioni nella CLI, esegui il comando riportato di seguito:

```
aws rds describe-db-instances \
--query 'DBInstances[*].
{DBInstanceIdentifier:DBInstanceIdentifier,PubliclyAccessible:PubliclyAccessible}' \
--output table
```

Per modificare le impostazioni di accessibilità per l'istanza database, consulta Modifica di un'istanza database Amazon RDS.

Fase 2: recupero della chiave segreta SSH e l'ID dell'istanza EC2

Per connettersi all'istanza database tramite SSH, è necessario disporre della coppia di chiavi SSH associata all'istanza. RDS Custom crea la coppia di chiavi SSH per conto dell'utente, denominandola con il prefisso. `do-not-delete-rds-custom-ssh-privatekey-db-` AWS Secrets Manager memorizza la tua chiave privata SSH come segreta.

Recupera la tua chiave segreta SSH utilizzando uno o AWS Management Console il. AWS CLI Se l'istanza ha un DNS pubblico e non intendi utilizzare il tunneling SSH, recupera anche il nome DNS. Specifica il nome DNS delle connessioni pubbliche.

Console

Per recuperare la chiave SSH segreta

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, scegliere Databases (Database) e quindi scegliere l'istanza database RDS Custom a cui desideri connetterti.
3. Scegliere Configuration (Configurazione).
4. Annota il valore Resource ID (Risorsa ID). Ad esempio, l'ID risorsa dell'istanza DB potrebbe essere `db-ABCDEFGHIJKLMNOPQRS0123456`.
5. Apri la console Amazon EC2 all'indirizzo https://console.aws.amazon.com/ec2/.
6. Nel riquadro di navigazione, seleziona Istanze.

7. Trova il nome dell'istanza EC2 e scegli l'ID istanza associato con esso. Ad esempio, l'ID istanza EC2 potrebbe essere `i-abcdefghijklm01234`.
8. In Details (Dettagli), trovare Key pair name (Nome della coppia di chiavi). Il nome della coppia include l'ID risorsa dell'istanza DB. Ad esempio, il nome della coppia potrebbe essere `do-not-delete-rds-custom-ssh-privatekey-db-ABCDEFGHIJKLMNOPQRS0123456-0d726c`.
9. Se l'istanza EC2 è pubblica, prendi nota del DNS IPv4 pubblico. Ad esempio, l'indirizzo del Domain Name System (DNS) pubblico potrebbe essere `ec2-12-345-678-901.us-east-2.compute.amazonaws.com`.
10. Apri la AWS Secrets Manager console all'indirizzo https://console.aws.amazon.com/secretsmanager/.
11. Scegliere il segreto che ha lo stesso nome della tua coppia di chiavi.
12. Scegli Retrieve secret value (Recupera il valore del segreto).
13. Copia la chiave SSH privata in un file di testo e salva il file con l'estensione `.pem`. Ad esempio, salva il file come `/tmp/do-not-delete-rds-custom-ssh-privatekey-db-ABCDEFGHIJKLMNOPQRS0123456-0d726c.pem`.

AWS CLI

Per recuperare la chiave SSH privata e salvarla in un file .pem, puoi usare la AWS CLI.

1. Cerca l'ID risorsa dell'istanza database RDS Custom tramite `aws rds describe-db-instances`.

```
aws rds describe-db-instances \
    --query 'DBInstances[*].[DBInstanceIdentifier,DbiResourceId]' \
    --output text
```

   Il seguente output di esempio mostra l'ID della risorsa per l'istanza RDS Custom. Il prefisso è `db-`.

```
db-ABCDEFGHIJKLMNOPQRS0123456
```

2. Cerca l'ID dell'istanza EC2 dell'istanza database tramite `aws ec2 describe-instances`. Nell'esempio seguente viene utilizzato `db-ABCDEFGHIJKLMNOPQRS0123456` per l'ID risorsa.

```
aws ec2 describe-instances \
    --filters "Name=tag:Name,Values=db-ABCDEFGHIJKLMNOPQRS0123456" \
```

```
    --output text \
    --query 'Reservations[*].Instances[*].InstanceId'
```

L'output di esempio seguente mostra l'ID dell'istanza EC2.

```
i-abcdefghijklm01234
```

3. Per trovare il nome chiave, specifica l'ID istanza EC2. L'esempio seguente descrive l'istanza EC2 *i-0bdc4219e66944afa*.

```
aws ec2 describe-instances \
    --instance-ids i-0bdc4219e66944afa \
    --output text \
    --query 'Reservations[*].Instances[*].KeyName'
```

Il seguente output di esempio mostra il nome della chiave, che utilizza il prefisso `do-not-delete-rds-custom-ssh-privatekey-`.

```
do-not-delete-rds-custom-ssh-privatekey-db-ABCDEFGHIJKLMNOPQRS0123456-0d726c
```

4. Salva la chiave privata in un file .pem avente lo stesso nome della chiave tramite `aws secretsmanager`. Nell'esempio seguente viene salvato il file nella tua directory `/tmp`.

```
aws secretsmanager get-secret-value \
    --secret-id do-not-delete-rds-custom-ssh-privatekey-db-
ABCDEFGHIJKLMNOPQRS0123456-0d726c \
    --query SecretString \
    --output text >/tmp/do-not-delete-rds-custom-ssh-privatekey-db-
ABCDEFGHIJKLMNOPQRS0123456-0d726c.pem
```

Fase 3: connessione all'istanza EC2 utilizzando l'utility ssh

La tecnica di connessione dipende dalla connessione a un'istanza DB privata o a un'istanza pubblica. Una connessione privata richiede la configurazione del tunneling SSH tramite AWS Systems Manager.

Per connettersi all'istanza EC2 tramite l'utility ssh

1. Per le connessioni private, modifica il file di configurazione SSH per i comandi proxy impostando AWS Systems Manager Session Manager. Per le connessioni pubbliche, esegui la fase 2.

   Aggiungi le righe seguenti a `~/.ssh/config`. Queste righe eseguono i comandi SSH in modalità proxy per gli host i cui nomi iniziano con `i-` o `mi-`.

   ```
   Host i-* mi-*
       ProxyCommand sh -c "aws ssm start-session --target %h --document-name AWS-
   StartSSHSession --parameters 'portNumber=%p'"
   ```

2. Passare alla directory che contiene il file .pem. Tramite `chmod`, imposta le autorizzazioni su `400`.

   ```
   cd /tmp
   chmod 400 do-not-delete-rds-custom-ssh-privatekey-db-
   ABCDEFGHIJKLMNOPQRS0123456-0d726c.pem
   ```

3. Esegui l'utilità ssh, specificando il file. pem e il nome DNS pubblico (per le connessioni pubbliche) o l'ID dell'istanza EC2 (per le connessioni private). Accedi come utente `ec2-user`.

   L'esempio seguente stabilisce una connessione a un'istanza pubblica utilizzando il nome DNS *ec2-12-345-678-901.us-east-2.compute.amazonaws.com*.

   ```
   ssh -i \
       "do-not-delete-rds-custom-ssh-privatekey-db-
   ABCDEFGHIJKLMNOPQRS0123456-0d726c.pem" \
       ec2-user@ec2-12-345-678-901.us-east-2.compute.amazonaws.com
   ```

   L'esempio seguente stabilisce una connessione a un'istanza privata utilizzando l'ID istanza EC2 *i-0bdc4219e66944afa*.

   ```
   ssh -i \
       "do-not-delete-rds-custom-ssh-privatekey-db-
   ABCDEFGHIJKLMNOPQRS0123456-0d726c.pem" \
       ec2-user@i-0bdc4219e66944afa
   ```

Accesso al database RDS Custom per Oracle come SYS

Dopo aver creato l'istanza database RDS Custom, è possibile accedere al database Oracle come utente SYS, ruolo che dispone dei privilegi SYSDBA. Sono disponibili le seguenti opzioni di accesso:

- Recupera la password SYS da Secrets Manager e specificala nel client SQL.
- Usa l'autenticazione del sistema operativo per accedere al database. In questo caso, non è necessario inserire una password.

Individuazione della password SYS per il database RDS Custom per Oracle

È possibile accedere al database Oracle come SYS o SYSTEM o specificando il nome utente principale in una chiamata API. La password per SYS e SYSTEM è archiviata in Secrets Manager. *Il segreto utilizza il formato di denominazione do-not-delete-rds -custom-resource_id - uuid.* Puoi cercare la password usando la AWS Management Console.

Console

Per cercare la password SYS per il tuo database in Secrets Manager

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Utilizzando la console RDS esegui i passaggi seguenti:
    a. Nel riquadro di navigazione, scegli Databases (Database).
    b. Scegli il nome dell'istanza database RDS Custom per Oracle.
    c. Scegliere Configuration (Configurazione).
    d. Copia il valore riportato sotto ID risorsa. Ad esempio, l'ID risorsa potrebbe essere db-ABC12cde3fgh4i5jKLMNO6PQR7.
3. Apri la console di Secrets Manager all'indirizzo https://console.aws.amazon.com/secretsmanager/.
4. Utilizzando la console Secrets Manager esegui i passaggi seguenti:
    a. Nel pannello di navigazione a sinistra, seleziona Segreti.
    b. Filtra i segreti in base all'ID risorsa copiato nel passaggio 5.
    c. Scegli il segreto denominato do-not-delete-rds-custom- ***resource_id - uuid, dove resource_id è l'ID*** della risorsa che hai copiato nel passaggio 5. Ad esempio, se l'ID

della risorsa è db-ABC12CDE3FGH4i5JKLMNO6PQR7, il segreto sarà denominato do-not-delete-rds-Custom-DB-ABC12CDE3FGH4i5JKLMNO6PQR7.

d. Nella sezione Valore segreto, scegli Recupera il valore di un segreto.

e. In Chiave/valore, copia il valore del campo Password.

5. Installa SQL*Plus sull'istanza DB e accedi al database come SYS. Per ulteriori informazioni, consulta Fase 3: connessione del client SQL a un'istanza database Oracle..

Accesso al database RDS Custom per Oracle utilizzando l'autenticazione del sistema operativo

L'utente del sistema operativo `rdsdb` possiede i file binari del database Oracle. È possibile passare all'utente `rdsdb` e accedere al database RDS Custom per Oracle senza password.

1. Connect alla propria istanza DB con AWS Systems Manager. Per ulteriori informazioni, consulta Connessione all'istanza database RDS Custom utilizzando Session Manager.
2. In un browser web, passa a https://www.oracle.com/database/technologies/instant-client/linux-x86-64-downloads.html.
3. Per la versione più recente del database visualizzata nella pagina web, copia i collegamenti .rpm (non i collegamenti .zip) per il pacchetto Instant Client Basic e il pacchetto SQL*Plus. Ad esempio, i seguenti link si riferiscono alla versione 21.9 di Oracle Database:

   - https://download.oracle.com/otn_software/linux/instantclient/219000/ oracle-instantclient-basic -21.9.0.0.0-1.el8.x86_64.rpm
   - https://download.oracle.com/otn_software/linux/instantclient/219000/ oracle-instantclient-sqlplus -21.9.0.0.0-1.el8.x86_64.rpm

4. Nella sessione SSH, esegui il comando `wget` per scaricare i file .rpm dai collegamenti che hai ottenuto nel passaggio precedente. L'esempio seguente scarica i file .rpm per Oracle Database versione 21.9:

```
wget https://download.oracle.com/otn_software/linux/instantclient/219000/oracle-instantclient-basic-21.9.0.0.0-1.el8.x86_64.rpm
wget https://download.oracle.com/otn_software/linux/instantclient/219000/oracle-instantclient-sqlplus-21.9.0.0.0-1.el8.x86_64.rpm
```

5. Installa i pacchetti eseguendo il comando `yum` come segue:

```
sudo yum install oracle-instantclient-*.rpm
```

6. Passa all'utente `rdsdb`.

```
sudo su - rdsdb
```

7. Accedi al database utilizzando l'autenticazione del sistema operativo.

```
$ sqlplus / as sysdba

SQL*Plus: Release 21.0.0.0.0 - Production on Wed Apr 12 20:11:08 2023
Version 21.9.0.0.0

Copyright (c) 1982, 2020, Oracle.  All rights reserved.


Connected to:
Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.10.0.0.0
```

Installazione di componenti software aggiuntivi sull'istanza database RDS Custom per Oracle

In un'istanza database appena creata, l'ambiente del database include foòe binari Oracle, un database e un ascoltatore di database. Potresti voler installare software aggiuntivo sul sistema operativo host dell'istanza database. Ad esempio, potresti voler installare Oracle Application Express (APEX), l'agente Oracle Enterprise Manager (OEM) o l'agente Guardium S-TAP. Per linee guida e istruzioni di alto livello, consulta il post dettagliato del AWS blog Installa componenti software aggiuntivi su Amazon RDS Custom for Oracle.

Gestione di istanze database Amazon RDS Custom for Oracle

Amazon RDS Custom supporta un sottoinsieme delle normali attività di gestione per le istanze database Amazon RDS. Di seguito, puoi trovare le istruzioni per le attività di gestione RDS Custom for Oracle supportate utilizzando la AWS Management Console e AWS CLI.

Argomenti

Utilizzo dei database container (CDB) in RDS Custom per Oracle

Puoi creare la tua istanza DB RDS Custom for Oracle con l'architettura multitenant Oracle (`custom-oracle-ee-cdb`o il tipo di `custom-oracle-se2-cdb` motore) o con l'architettura tradizionale non CDB (o tipo di motore). `custom-oracle-ee` `custom-oracle-se2` Quando crei un database container (CDB), include un database collegabile (PDB) e un seed PDB. È possibile creare manualmente altri PDB utilizzando Oracle SQL.

Nomi PDB e CDB

Quando crei un'istanza CDB RDS Custom per Oracle, è necessario specificare un nome del PDB iniziale. Per impostazione predefinita, il nome iniziale del PDB è `ORCL`. È possibile scegliere un nome diverso.

Per impostazione predefinita, il nome del tuo CDB è `RDSCDB`. È possibile scegliere un nome diverso. Il nome CDB è anche il nome dell'identificatore di sistema (SID) Oracle, che identifica in modo univoco la memoria e i processi che gestiscono il CDB. Per ulteriori informazioni sul SID Oracle, consulta la sezione relativa all'identificatore di sistema (SID) Oracle nel manuale Oracle Database Concepts.

Non puoi rinominare i PDB esistenti utilizzando le API Amazon RDS. Inoltre, non è possibile rinominare il CDB utilizzando il comando `modify-db-instance`.

Gestione dei PDB

Nel modello di responsabilità condivisa RDS Custom per Oracle, sei responsabile della gestione dei PDB e della creazione di eventuali PDB aggiuntivi. RDS Custom non limita il numero di PDB. È possibile creare, modificare ed eliminare manualmente i PDB collegandosi alla root CDB ed eseguendo un'istruzione SQL. Crea i PDB su un volume di dati Amazon EBS per evitare che l'istanza database esca dal perimetro di supporto.

Per modificare i CDB o i PDB, completa la procedura seguente:

1. Sospendi l'automazione per evitare interferenze con le azioni di RDS Custom.
2. Modifica i tuoi CDB o PDB.
3. Esegui il backup di tutti i PDB modificati.
4. Riprendere l'automazione RDS Custom.

Ripristino automatico della root del CDB

RDS Custom mantiene aperta la root CDB nello stesso modo in cui mantiene aperto un non CDB. Se lo stato della root CDB cambia, il monitoraggio e l'automazione del ripristino tenta di ripristinare la root CDB allo stato desiderato. Vengono generate le notifiche di eventi RDS quando il CDB root viene chiuso (`RDS-EVENT-0004`) o riavviato (`RDS-EVENT-0006`), in modo simile all'architettura non CDB. RDS Custom tenta di aprire tutti i PDB in modalità `READ WRITE` all'avvio dell'istanza database. Se alcuni PDB non possono essere aperti, RDS Custom pubblica il seguente evento: `tenant database shutdown`.

Utilizzo di funzionalità ad alta disponibilità per RDS Custom per Oracle

Per supportare la replica tra istanze RDS Custom for Oracle DB, puoi configurare l'alta disponibilità (HA) con Oracle Data Guard. L'istanza database principale sincronizza automaticamente i dati con le istanze di standby. Questa funzionalità è supportata solo in Enterprise Edition.

Puoi configurare l'ambiente ad elevata disponibilità nei seguenti modi:

- Configurare le istanze in standby in zone di disponibilità (AZ) diverse in modo che siano resilienti agli errori AZ.

- Posizionare i database in standby in modalità montata o di sola lettura.
- Si passa o si esegue il failover dal database principale a un database in standby senza perdita di dati.
- Eseguire la migrazione dei dati configurando l'elevata disponibilità per l'istanza On-Premise, quindi eseguendo il failover o passando al database standby RDS Custom.

Per informazioni su come configurare la disponibilità elevata, consulta il white paper Build high availability for Amazon RDS Custom for Oracle using read replicas (Abilitazione della disponibilità elevata per Amazon RDS Custom per Oracle usando le repliche di lettura). Si possono eseguire queste attività:

- Utilizzare un tunnel Virtual Private Network (VPN) per crittografare i dati in transito per le istanze ad alta disponibilità. La crittografia in transito non è configurata automaticamente da RDS Custom.
- Configurare Oracle Fast-Failover Observer (FSFO) per monitorare le istanze ad alta disponibilità.
- Consentire all'osservatore di eseguire il failover automatico quando sono soddisfatte le condizioni necessarie.

Personalizzazione dell'ambiente RDS Custom

RDS Custom per Oracle include funzionalità dedicate che consentono di personalizzare l'ambiente delle istanze DB senza interrompere l'automazione. Ad esempio, è possibile utilizzare le API RDS per personalizzare l'ambiente come segue:

- Creare e ripristinare gli snapshot DB per creare un ambiente di clonazione.
- Creare repliche di lettura.
- Modificare le impostazioni di archiviazione.
- Modificare la CEV per applicare gli aggiornamenti dei rilasci.

Per alcune personalizzazioni, come la modifica del set di caratteri, non è possibile utilizzare le API RDS. In questi casi, è necessario modificare l'ambiente manualmente accedendo all'istanza Amazon EC2 come utente root o accedendo al database Oracle come `SYSDBA`.

Per personalizzare l'istanza manualmente, è necessario sospendere e riprendere l'automazione RDS Custom. La pausa garantisce che le personalizzazioni non interferiscano con l'automazione di RDS Custom. In questo modo, si evita di interrompere il perimetro di supporto, che pone l'istanza

nello stato `unsupported-configuration` finché non vengono risolti i problemi sottostanti. La sospensione e la ripresa sono le uniche attività di automazione supportate durante la modifica di un'istanza database RDS Custom per Oracle.

Passaggi generali per personalizzare l'ambiente RDS Custom

Per personalizzare l'istanza database RDS Custom, occorre eseguire le seguenti operazioni:

1. Sospendi l'automazione RDS Custom per un periodo di tempo specificato tramite la console o la CLI.
2. Identifica l'istanza Amazon EC2 sottostante.
3. Stabilisci la connessione all'istanza Amazon EC2 sottostante usando le chiavi SSH o AWS Systems Manager.
4. Verifica le impostazioni di configurazione correnti a livello di database o sistema operativo.

   È possibile convalidare le modifiche confrontando la configurazione iniziale con quella modificata. A seconda del tipo di personalizzazione, utilizza gli strumenti del sistema operativo o le query di database.
5. Personalizza l'istanza database RDS Custom per Oracle secondo necessità.
6. Riavvia l'istanza o il database, se necessario.

   > **Note**
   >
   > In un CDB Oracle on-premise, è possibile mantenere una modalità aperta specificata per i PDB utilizzando un comando integrato o dopo un trigger di avvio. Questo meccanismo porta i PDB in uno stato specificato al riavvio del CDB. Quando apri il CDB, l'automazione RDS Custom elimina sempre gli stati conservati specificati dall'utente e tenta di aprire tutti i PDB. Se RDS Custom non riesce ad aprire tutti i PDB, viene emesso il seguente evento: `The following PDBs failed to open: list-of-PDBs`.
7. Verifica le nuove impostazioni di configurazione confrontandole con le impostazioni precedenti.
8. Riprendi l'automazione RDS Custom in uno dei seguenti modi:
   - Riprendi l'automazione manualmente.
   - Attendi che il periodo di pausa finisca. In questo caso, RDS Custom riprende automaticamente il monitoraggio e il ripristino delle istanze.
9. Verifica il framework di automazione di RDS Custom.

Se hai seguito correttamente i passaggi precedenti, RDS Custom avvia un backup automatico. Lo stato dell'istanza nella console è Disponibile.

Per best practice e step-by-step istruzioni, consulta i post del AWS blog Apportare modifiche alla configurazione di un'istanza Amazon RDS Custom for Oracle: parte 1 e Ricreare un database Amazon RDS personalizzato per Oracle: parte 2.

Sospensione e ripresa dell'istanza database RDS Custom

È possibile sospendere e riprendere l'automazione dell'istanza database tramite la console o la CLI.

Console

Per sospendere o riprendere l'automazione RDS Custom

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel pannello di navigazione, scegliere Databases (Database) e selezionare l'istanza database RDS Custom da modificare.
3. Scegliere Modify (Modifica). Viene visualizzata la pagina Modify DB Instance (Modifica istanza database).
4. Per Modalità di automazione RDS Custom, scegliere una delle seguenti opzioni:
   - Paused (In pausa) sospende il monitoraggio e il ripristino dell'istanza per l'istanza database RDS Custom. Inserire la durata di pausa desiderata (in minuti) Durata della modalità di automazione. Il valore minimo è 60 minuti (predefinito). Il valore massimo è 1.440 minuti.
   - Automazione completa riprende l'automazione.
5. Scegliere Continue (Continua) per controllare il riepilogo delle modifiche.

   Un messaggio indica che RDS Custom applicherà immediatamente le modifiche.
6. Se le modifiche sono corrette, selezionare Modify DB Instance (Modifica istanza database). Oppure scegliere Back (Indietro) per cambiare le modifiche o Cancel (Annulla) per annullare le modifiche.

   Nella console RDS vengono visualizzati i dettagli per la modifica. Se hai interrotto l'automazione, lo Stato della tua istanza database RDS Custom indica Automation paused (Sospensione dell'automazione).

7. (Opzionale) Nel pannello di navigazione, scegliere Databases (Database), quindi scegliere un'istanza database RDS Custom.

   Nel pannello Summary (Riepilogo), la Modalità di automazione RDS Custom indica lo stato dell'automazione. Se l'automazione è sospesa, il valore è In pausa. L'automazione riprende in ***num*** minuti.

AWS CLI

Per mettere in pausa o riprendere l'automazione RDS Custom, usa il comando. `modify-db-instance` AWS CLI Identificare l'istanza database utilizzando il parametro richiesto `--db-instance-identifier`. Controllare la modalità di automazione con i seguenti parametri:

- `--automation-mode` specifica lo stato di pausa dell'istanza database. I valori validi sono `all-paused`, che mette in pausa l'automazione e `full`, che la riprende.
- `--resume-full-automation-mode-minutes` specifica la durata della pausa. Il valore predefinito è di 60 minuti.

> **Note**
>
> Indipendentemente dal fatto che tu specifichi `--no-apply-immediately` o `--apply-immediately`, RDS Custom applica le modifiche in modo asincrono il prima possibile.

Nella risposta al comando, `ResumeFullAutomationModeTime` indica l'orario di ripristino come timestamp UTC. Quando la modalità di automazione è `all-paused`, è possibile utilizzare `modify-db-instance` per riprendere la modalità di automazione o prolungare il periodo di pausa. Non sono supportate altre opzioni `modify-db-instance`.

L'esempio seguente sospende l'automazione per `my-custom-instance` per 90 minuti.

Example

PerLinux, o: macOS Unix

```
aws rds modify-db-instance \
    --db-instance-identifier my-custom-instance \
    --automation-mode all-paused \
```

```
    --resume-full-automation-mode-minutes 90
```

Per Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier my-custom-instance ^
    --automation-mode all-paused ^
    --resume-full-automation-mode-minutes 90
```

L'esempio seguente estende la durata della pausa di altri 30 minuti. I 30 minuti vengono aggiunti all'orario di origine mostrato in `ResumeFullAutomationModeTime`.

Example

Per LinuxmacOS, oUnix:

```
aws rds modify-db-instance \
    --db-instance-identifier my-custom-instance \
    --automation-mode all-paused \
    --resume-full-automation-mode-minutes 30
```

Per Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier my-custom-instance ^
    --automation-mode all-paused ^
    --resume-full-automation-mode-minutes 30
```

L'esempio seguente riprende l'automazione completa per `my-custom-instance`.

Example

Per LinuxmacOS, oUnix:

```
aws rds modify-db-instance \
    --db-instance-identifier my-custom-instance \
    --automation-mode full \
```

Per Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier my-custom-instance ^
    --automation-mode full
```

Nel seguente output di esempio parziale, il valore `AutomationMode` in attesa è `full`.


```
{
    "DBInstance": {
        "PubliclyAccessible": true,
        "MasterUsername": "admin",
        "MonitoringInterval": 0,
        "LicenseModel": "bring-your-own-license",
        "VpcSecurityGroups": [
            {
                "Status": "active",
                "VpcSecurityGroupId": "0123456789abcdefg"
            }
        ],
        "InstanceCreateTime": "2020-11-07T19:50:06.193Z",
        "CopyTagsToSnapshot": false,
        "OptionGroupMemberships": [
            {
                "Status": "in-sync",
                "OptionGroupName": "default:custom-oracle-ee-19"
            }
        ],
        "PendingModifiedValues": {
            "AutomationMode": "full"
        },
        "Engine": "custom-oracle-ee",
        "MultiAZ": false,
        "DBSecurityGroups": [],
        "DBParameterGroups": [
            {
                "DBParameterGroupName": "default.custom-oracle-ee-19",
                "ParameterApplyStatus": "in-sync"
            }
        ],
        ...
        "ReadReplicaDBInstanceIdentifiers": [],
        "AllocatedStorage": 250,
        "DBInstanceArn": "arn:aws:rds:us-west-2:012345678912:db:my-custom-instance",
        "BackupRetentionPeriod": 3,
```

```
        "DBName": "ORCL",
        "PreferredMaintenanceWindow": "fri:10:56-fri:11:26",
        "Endpoint": {
            "HostedZoneId": "ABCDEFGHIJKLMNO",
            "Port": 8200,
            "Address": "my-custom-instance.abcdefghijk.us-west-2.rds.amazonaws.com"
        },
        "DBInstanceStatus": "automation-paused",
        "IAMDatabaseAuthenticationEnabled": false,
        "AutomationMode": "all-paused",
        "EngineVersion": "19.my_cev1",
        "DeletionProtection": false,
        "AvailabilityZone": "us-west-2a",
        "DomainMemberships": [],
        "StorageType": "gp2",
        "DbiResourceId": "db-ABCDEFGHIJKLMNOPQRSTUVW",
        "ResumeFullAutomationModeTime": "2020-11-07T20:56:50.565Z",
        "KmsKeyId": "arn:aws:kms:us-west-2:012345678912:key/
aa111a11-111a-11a1-1a11-1111a11a1a1a",
        "StorageEncrypted": false,
        "AssociatedRoles": [],
        "DBInstanceClass": "db.m5.xlarge",
        "DbInstancePort": 0,
        "DBInstanceIdentifier": "my-custom-instance",
        "TagList": []
    }
```


Modifica dell'istanza database RDS Custom per Oracle

La modifica di un'istanza DB RDS Custom for Oracle è simile alla modifica di un'istanza DB Amazon RDS. Puoi modificare impostazioni come le seguenti:

- DB instance class (Classe istanza database)
- Allocazione e tipo di archiviazione
- Backup retention period (Periodo di retention dei backup)
- Deletion protection (Protezione da eliminazione)
- Option group (Gruppo di opzioni)
- CEV (vedi) Aggiornamento di un'istanza database RDS Custom per Oracle
- Porta

Argomenti

Requisiti e limitazioni per la modifica dell'archiviazione dell'istanza database

Tieni presenti i seguenti requisiti e le limitazioni quando modifichi l'archiviazione di un'istanza database RDS Custom per Oracle:

- Lo storage minimo allocato per RDS Custom per Oracle è 40 GiB e il massimo è 64 TiB.
- Come per Amazon RDS, non è possibile ridurre lo storage allocato. Questa è una limitazione dei volumi Amazon EBS.
- La scalabilità automatica dello storage non è supportato per le istanze database RDS Custom.
- Tutti i volumi di archiviazione collegati manualmente all'istanza database RDS Custom si trovano al di fuori del perimetro di supporto.

  Per ulteriori informazioni, consulta Perimetro di supporto RDS Custom.
- Il tipo di archiviazione magnetico (standard) Amazon EBS non è supportato per RDS Custom. Puoi scegliere solo i tipi di archiviazione SSD io1, gp2 o gp3.

Per ulteriori informazioni sull'archiviazione Amazon EBS, consulta Storage delle istanze di database Amazon RDS. Per informazioni generali sulla modifica dello storage, consulta Uso dello storage per istanze database di Amazon RDS.

Requisiti e limitazioni durante la modifica della classe di istanza database

Tieni presenti i seguenti requisiti e le limitazioni quando modifichi la classe di istanza per un'istanza database RDS Custom per Oracle:

- L'istanza database deve essere nello stato `available`.
- L'istanza database deve disporre di almeno 100 MiB di spazio libero sul volume root, sul volume dei dati e sul volume binario.

- È possibile assegnare un solo IP elastico (EIP) all'istanza database RDS Custom per Oracle quando si utilizza l'interfaccia di rete elastica (ENI) predefinita. Se colleghi più ENI all'istanza database, l'operazione di modifica ha esito negativo.
- Tutti i tag RDS Custom per Oracle devono essere presenti.
- Tieni presenti i requisiti e le limitazioni seguenti se usi la replica di RDS Custom per Oracle:
  - Per le istanze database primarie e le repliche di lettura, è possibile modificare la classe di istanza per una sola istanza database alla volta.
  - Se l'istanza database RDS Custom per Oracle dispone di un database primario o di replica on-premise, assicurati di aggiornare manualmente gli indirizzi IP privati sull'istanza database on-premise al termine della modifica. Questa azione è necessaria per preservare la DataGuard funzionalità di Oracle. RDS Custom per Oracle pubblica un evento quando la modifica ha esito positivo.
  - Non è possibile modificare la classe di istanza database RDS Custom per Oracle quando le istanze database primarie o di replica di lettura hanno configurato FSFO (Fast-Start Failover).

In che modo RDS Custom crea l'istanza database quando si modifica la classe di istanza

Quando modifichi la classe di istanza, RDS Custom crea l'istanza database come segue:

- Crea l'istanza Amazon EC2.
- Crea il volume root dall'ultimo snapshot di database. RDS Custom per Oracle non mantiene le informazioni aggiunte al volume root dopo l'ultimo snapshot di database.
- Crea CloudWatch allarmi Amazon.
- Crea una coppia di chiavi SSH Amazon EC2 se hai eliminato la coppia di chiavi originale. Altrimenti, RDS Custom per Oracle mantiene la coppia di chiavi originale.
- Crea nuove risorse utilizzando i tag associati all'istanza database quando si avvia la modifica. RDS Custom non trasferisce i tag alle nuove risorse quando sono collegate direttamente alle risorse sottostanti.
- Trasferisce i volumi binari e di dati con le modifiche più recenti alla nuova istanza database.
- Trasferisce l'indirizzo IP elastico (EIP). Se l'istanza database è accessibile pubblicamente, RDS Custom associa temporaneamente un indirizzo IP pubblico alla nuova istanza database prima di trasferire l'indirizzo EIP. Se l'istanza database non è accessibile pubblicamente, RDS Custom non crea indirizzi IP pubblici.

Modifica dell'istanza database RDS Custom per Oracle

Puoi modificare la classe o lo storage dell'istanza DB utilizzando la console o AWS CLI l'API RDS.

Console

Per modificare un'istanza database RDS Custom per Oracle

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, scegliere Databases (Database).
3. Scegliere l'istanza database che si vuole modificare.
4. Scegliere Modify (Modifica).
5. (Facoltativo) Nella configurazione dell'istanza, scegli un valore per la classe dell'istanza DB. Per le classi supportate, consultare Supporto delle classi di istanza database per RDS Custom per Oracle.
6. (Facoltativo) In Storage, apporta le seguenti modifiche in base alle esigenze:
   a. Inserire un nuovo valore per Allocated Storage (Storage allocato). Questo valore deve essere maggiore di quello corrente e da 40 Gib—64 TiB.
   b. Modifica il valore nel campo Tipo di archiviazione impostandolo su SSD per scopo generico (gp2), SSD per scopo generico (gp3) o Capacità di IOPS allocata (io1).
   c. Se utilizzi Capacità di IOPS allocata (io1) o SSD per scopo generico (gp3), puoi modificare il valore del campo Capacità di IOPS allocata.
7. (Facoltativo) In Configurazione aggiuntiva, apporta le seguenti modifiche in base alle esigenze:
   - Per Gruppo di opzioni, scegliete un nuovo gruppo di opzioni. Per ulteriori informazioni, consulta Utilizzo dei gruppi di opzioni in RDS Custom for Oracle.
8. Scegli Continue (Continua).
9. Scegliere Apply immediately (Applica immediatamente) o Apply during the next scheduled maintenance window (Applica durante la prossima finestra di manutenzione pianificata).
10. Scegliere Modify DB Instance (Modifica istanza database).

AWS CLI

Per modificare lo storage per un'istanza DB RDS Custom for Oracle, utilizzare il modify-db-instance AWS CLI comando. Impostazione dei parametri seguenti in base alle esigenze:

- `--db-instance-class` – Una nuova classe di istanza. Per le classi supportate, consultare Supporto delle classi di istanza database per RDS Custom per Oracle.
- `--allocated-storage`: la quantità di spazio di archiviazione, in gibibyte, da allocare per l'istanza database. Questo valore deve essere maggiore di quello corrente e da 40–65,536 GiB.
- `--storage-type`: il tipo di archiviazione, ovvero gp2, gp3 o io1.
- `--iops`: Capacità di IOPS allocata per l'istanza database, se utilizzi i tipi di archiviazione io1 o gp3.
- `--apply-immediately`: utilizza `--apply-immediately` per applicare immediatamente le modifiche.

  Oppure utilizza `--no-apply-immediately` (impostazione di default) per applicare le modifiche durante la finestra di manutenzione successiva.

L'esempio seguente modifica la classe dell'istanza DB in my-cfo-instance db.m5.16xlarge. Il comando modifica anche la dimensione di archiviazione a 1 TiB, il tipo di archiviazione a io1, Provisioned IOPS a 3000 e il gruppo di opzioni in cfo-ee-19-mt.

Example

Per, o: Linux macOS Unix

```
aws rds modify-db-instance \
    --db-instance-identifier my-cfo-instance \
    --db-instance-class db.m5.16xlarge \
    --storage-type io1 \
    --iops 3000 \
    --allocated-storage 1024 \
    --option-group cfo-ee-19-mt \
    --apply-immediately
```

Per Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier my-cfo-instance ^
    --db-instance-class db.m5.16xlarge ^
    --storage-type io1 ^
    --iops 3000 ^
    --allocated-storage 1024 ^
    --option-group cfo-ee-19-mt ^
```

```
    --apply-immediately
```

Modifica del set di caratteri di un'istanza database di RDS Custom per Oracle

RDS Custom per Oracle è impostato per default sul set di caratteri US7ASCII. Potrebbe essere necessario specificare set di caratteri diversi per soddisfare i requisiti di caratteri lingua o multibyte. Quando si utilizza RDS Custom per Oracle, puoi sospendere l'automazione e quindi modificare manualmente il set di caratteri del database.

La modifica del set di caratteri di un'istanza database di RDS Custom per Oracle ha i seguenti requisiti:

- Puoi modificare il carattere solo su un'istanza di RDS Custom appena sottoposta a provisioning che dispone di un database vuoto o di avvio senza dati dell'applicazione. Per tutti gli altri scenari, modifica il set di caratteri utilizzando DMU (Database Migration Assistant for Unicode).
- Puoi passare solo a un set di caratteri supportato da RDS per Oracle. Per ulteriori informazioni, consulta Set di caratteri DB supportati.

Modificare il set di caratteri di un'istanza database di RDS Custom per Oracle

1. Sospendi l'automazione RDS Custom. Per ulteriori informazioni, consulta Sospensione e ripresa dell'istanza database RDS Custom.
2. Accedi al database come utente con privilegi SYSDBA.
3. Riavvia il database in modalità limitata, modifica il set di caratteri e quindi riavvia il database in modalità normale.

   Esegui lo script seguente nel client SQL:

   ```
   SHUTDOWN IMMEDIATE;
   STARTUP RESTRICT;
   ALTER DATABASE CHARACTER SET INTERNAL_CONVERT AL32UTF8;
   SHUTDOWN IMMEDIATE;
   STARTUP;
   SELECT VALUE FROM NLS_DATABASE_PARAMETERS WHERE PARAMETER = 'NLS_CHARACTERSET';
   ```

   Verifica che l'output contenga il set di caratteri corretto:

   ```
   VALUE
   --------
   ```

```
AL32UTF8
```

4. Riprendere l'automazione RDS Custom. Per ulteriori informazioni, consulta Sospensione e ripresa dell'istanza database RDS Custom.

Impostazione del valore NLS_LANG in RDS Custom per Oracle

Un locale è un insieme di informazioni che riguardano i requisiti linguistici e culturali che corrispondono a una determinata lingua e paese. Per specificare il comportamento delle impostazioni locali per il software Oracle, imposta la variabile di ambiente `NLS_LANG` sull'host client. Questo parametro imposta la lingua. il territorio e il set di caratteri utilizzati dall'applicazione client in una sessione di database.

Per RDS Custom per Oracle, nella variabile `NLS_LANG` è possibile impostare solo la lingua: il territorio e il set di caratteri utilizzano le impostazioni predefinite. La lingua viene utilizzata per i messaggi del database Oracle, l'ordinamento, i nomi dei giorni e i nomi dei mesi. Ogni lingua supportata ha un nome univoco, ad esempio americano, francese o tedesco. Se la lingua non è specificata, il valore predefinito è Americano.

Dopo aver creato il database RDS Custom per Oracle, è possibile impostare`NLS_LANG` sull'host client configurando una lingua diversa dall'inglese. Per visualizzare l'elenco delle lingue supportate da Oracle Database, accedi al database RDS Custom per Oracle ed esegui la seguente query:

```
SELECT VALUE FROM V$NLS_VALID_VALUES WHERE PARAMETER='LANGUAGE' ORDER BY VALUE;
```

Puoi impostare `NLS_LANG` sulla riga di comando dell'host. L'esempio seguente imposta la lingua su Tedesco per l'applicazione client utilizzando la shell (interprete di comandi) Z su Linux.

```
export NLS_LANG=German
```

L'applicazione legge il valore `NLS_LANG` all'avvio e quindi lo comunica al database quando si connette.

Per ulteriori informazioni, consulta la pagina relativa alla scelta di una lingua con la variabile di ambiente NLS_LANG nel manuale Oracle Database Globalization Support Guide.

Supporto per Transparent Data Encryption

RDS Custom supporta Transparent Data Encryption (TDE) per le istanze database RDS Custom for Oracle.

Tuttavia, non è possibile abilitare TDE utilizzando un'opzione in un gruppo di opzioni personalizzato come è possibile in RDS for Oracle. Attiva TDE manualmente. Per informazioni sull'uso di Oracle Transparent Data Encryption, consulta Garantire la sicurezza dei dati archiviati con Transparent Data Encryption nella documentazione di Oracle.

Assegnazione di tag a risorse RDS Custom for Oracle

Puoi taggare le risorse RDS Custom come con le risorse Amazon RDS, ma con alcune importanti differenze:

- Non creare o modificare la tag `AWSRDSCustom` richiesta per l'automazione RDS Custom. Se lo fai, potresti interrompere l'automazione.
- Il tag `Name` viene aggiunto alle risorse RDS Custom con il valore del prefisso `do-not-delete-rds-custom`. Qualsiasi valore passato dal cliente per la chiave viene sovrascritto.
- Le tag aggiunte alle istanze database RDS Custom durante la creazione vengono propagate a tutte le altre risorse RDS Custom correlate.
- Le tag non vengono propagate quando le aggiungi alle risorse RDS Custom dopo la creazione dell'istanza database.

Per informazioni sul tagging delle risorse, consulta Tagging delle risorse Amazon RDS.

Eliminazione di un'istanza database RDS Custom for Oracle

Per eliminare un'istanza database RDS Custom, occorre eseguire quanto segue:

- Fornire il nome dell'istanza database.
- Deselezionare l'opzione per acquisire uno snapshot DB finale dell'istanza database.
- Scegliere o deselezionare l'opzione per mantenere i backup automatici.

È possibile eliminare un'istanza database RDS Custom utilizzando la console o CLI. Il tempo necessario per eliminare un'istanza database può variare a seconda del periodo di conservazione del backup, ovvero del numero di backup da eliminare, dalla quantità di dati eliminati.

Console

Per eliminare un'istanza database RDS Custom

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, scegliere Databases (Database) e selezionare l'istanza database RDS Custom da eliminare. Le istanze database RDS Custom mostrano il ruolo Istanza (RDS Custom).
3. In Actions (Azioni), scegliere Delete (Elimina).
4. Per mantenere i backup automatici, scegliere Retain automated backups (Mantieni backup automatici).
5. Immettere **`delete me`** nella casella.
6. Scegliere Delete (Elimina).

AWS CLI

È possibile eliminare un'istanza DB personalizzata RDS utilizzando il delete-db-instance AWS CLI comando. Identificare l'istanza database utilizzando il parametro richiesto `--db-instance-identifier`. I parametri rimanenti sono gli stessi di un'istanza database Amazon RDS, con le seguenti eccezioni:

- `--skip-final-snapshot` è obbligatorio.
- `--no-skip-final-snapshot` non è supportata.
- `--final-db-snapshot-identifier` non è supportata.

L'esempio seguente elimina l'istanza database RDS Custom denominata `my-custom-instance` e mantiene backup automatici.

Example

Per LinuxmacOS, oUnix:

```
aws rds delete-db-instance \
    --db-instance-identifier my-custom-instance \
    --skip-final-snapshot \
    --no-delete-automated-backups
```

Per Windows:

```
aws rds delete-db-instance ^
    --db-instance-identifier my-custom-instance ^
    --skip-final-snapshot ^
    --no-delete-automated-backups
```

Utilizzo delle repliche Oracle per RDS Custom per Oracle

È possibile creare repliche Oracle per RDS Custom per istanze DB Oracle che eseguono Oracle Enterprise Edition. Sono supportati sia i database container (CDB) che quelli non CDB. La Standard Edition 2 non supporta Oracle Data Guard.

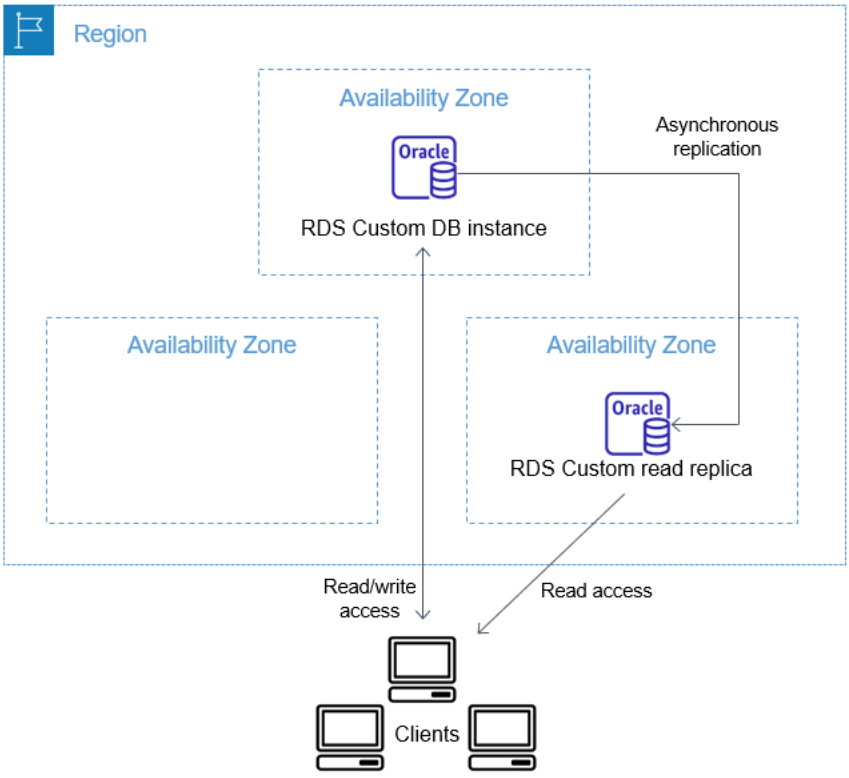
La creazione di una replica RDS Custom per Oracle è simile alla creazione di una replica RDS per Oracle, ma con alcune differenze importanti. Per informazioni generali sulla creazione e la gestione delle repliche Oracle, consulta Uso delle repliche di lettura dell'istanza database e Utilizzo di repliche di lettura per Amazon RDS per Oracle.

Argomenti

- Panoramica della replica RDS Custom per Oracle
- Linee guida e limitazioni per la replica RDS Custom per Oracle
- Promozione di una replica RDS Custom per Oracle a istanza database autonoma

Panoramica della replica RDS Custom per Oracle

L'architettura di replica RDS Custom per Oracle è analoga all'architettura di replica RDS per Oracle. Un'istanza database primaria si replica in modo asincrono su una o più repliche Oracle.



Numero massimo di repliche

Come con RDS per Oracle, è possibile creare fino a cinque repliche Oracle gestite dell'istanza database primaria RDS Custom per Oracle. È inoltre possibile creare repliche Oracle personalizzate (esterne) configurate manualmente. Le repliche esterne non contano ai fini del limite delle istanze database. Si trovano inoltre al di fuori del perimetro di supporto di RDS Custom. Per ulteriori informazioni sul perimetro di supporto, vedere Perimetro di supporto RDS Custom.

Convenzione di denominazione delle repliche

I nomi delle repliche Oracle si basano sul nome univoco del database. Il formato è *DB_UNIQUE_NAME*_*X*, con lettere aggiunte in sequenza. Ad esempio, se il nome univoco del database è `ORCL`, le prime due repliche sono denominate `ORCL_A` e `ORCL_B`. Le prime sei lettere, A—F, sono riservate per RDS Custom. RDS Custom copia i parametri del database dall'istanza database primaria alle repliche. Per ulteriori informazioni, consulta DB_UNIQUE_NAME nella documentazione di Oracle.

Conservazione dei backup delle repliche

Le repliche RDS Custom utilizzano lo stesso tempo di conservazione del backup dell'istanza database primaria per impostazione predefinita. È possibile modificare il tempo di conservazione del backup (1–35 giorni). RDS Custom supporta il backup, il ripristino e il point-in-time ripristino (PITR). Per ulteriori informazioni sul backup e il ripristino delle istanze database di RDS Custom, consulta Backup e ripristino di un'istanza database di Amazon RDS Custom per Oracle.

> **Note**
>
> Durante la creazione di una replica Oracle, RDS Custom interrompe temporaneamente la pulizia dei registri di ripristino. In questo modo, RDS Custom garantisce che questi registri vengano applicati alla nuova replica Oracle quando sarà disponibile.

Promozione delle repliche

È possibile promuovere le repliche Oracle gestite in RDS Custom for Oracle utilizzando la console, il comando o l'API. `promote-read-replica` AWS CLI `PromoteReadReplica` Se si elimina l'istanza database primaria e tutte le repliche sono integre, RDS Custom per Oracle promuove automaticamente le repliche gestite in istanze autonome. Se una replica ha sospeso l'automazione o si trova al di fuori del perimetro di supporto, è necessario correggerla prima che RDS Custom possa promuoverla automaticamente. È possibile promuovere le repliche Oracle esterne solo manualmente.

Linee guida e limitazioni per la replica RDS Custom per Oracle

Non tutte le opzioni di replica RDS Oracle sono supportate quando si creano repliche RDS Custom per Oracle.

Argomenti

Linee guida generali per la replica RDS Custom per Oracle

Quando si usa RDS Custom per Oracle, segui le linee guida riportate di seguito:

- È possibile utilizzare RDS Custom per la replica Oracle solo in Oracle Enterprise Edition. La Standard Edition 2 non è supportata.
- Non modificare l'utente `RDS_DATAGUARD`. Questo utente è riservato per l'automazione RDS Custom per Oracle. La modifica di questo utente può restituire risultati indesiderati, ad esempio l'impossibilità di creare repliche Oracle per l'istanza database RDS Custom per Oracle.
- Non modificare la password dell'utente di replica. Questa password è necessaria per amministrare la configurazione di Oracle Data Guard sull'host RDS Custom. Se si modifica la password, RDS Custom per Oracle potrebbe posizionare la replica Oracle al di fuori del perimetro di supporto. Per ulteriori informazioni, consulta Perimetro di supporto RDS Custom.

  La password è memorizzata in AWS Secrets Manager, contrassegnata con l'ID della risorsa DB. Ogni replica Oracle ha il suo segreto in Secrets Manager. Di seguito è riportato il formato per il segreto.

  ```
  do-not-delete-rds-custom-db-DB_resource_id-6-digit_UUID-dg
  ```
- Non modificare il valore `DB_UNIQUE_NAME` per un'istanza database primaria. La modifica del nome causa il blocco di qualsiasi operazione di ripristino.
- Non specificare la clausola `STANDBYS=NONE` in un comando `CREATE PLUGGABLE DATABASE` in un CDB RDS Custom. In questo modo, in caso di failover, il CDB in standby contiene tutti i PDB.

Limitazioni generali per la replica RDS Custom per Oracle

Le repliche RDS Custom per Oracle hanno le seguenti limitazioni:

- Non è possibile creare repliche RDS Custom per Oracle solo in modalità di sola lettura. Tuttavia, è possibile convertire manualmente le repliche dalla modalità montata a in sola lettura e da in sola lettura a montata. Per ulteriori informazioni, consulta la documentazione del comando create-db-instance-read-replica AWS CLI .
- Non è possibile creare repliche RDS Custom per Oracle tra regioni.
- Non è possibile modificare il valore del parametro Oracle Data Guard `CommunicationTimeout`. Questo parametro è impostato su 15 secondi per le istanze database RDS Custom per Oracle.

Requisiti e limitazioni delle reti per la replica RDS Custom per Oracle

Verifica che la configurazione di rete supporti le repliche RDS Custom per Oracle. Considera i seguenti aspetti:

- Assicurati di abilitare la porta 1140 per la comunicazione in entrata e in uscita all'interno del cloud privato virtuale (VPC) per l'istanza database primaria e tutte le relative repliche. Ciò è necessario per la comunicazione di Oracle Data Guard tra le repliche di lettura.
- RDS Custom per Oracle convalida la rete durante la creazione di una replica Oracle. Se l'istanza database primaria e la nuova replica non riescono a connettersi in rete, RDS Custom per Oracle non crea la replica e imposta il relativo stato su `INCOMPATIBLE_NETWORK`.
- Per le repliche Oracle esterne, ad esempio quelle create su Amazon EC2 o on-premise, utilizza un'altra porta e un listener per la replica Oracle Data Guard. Il tentativo di utilizzare la porta 1140 potrebbe causare conflitti con l'automazione RDS Custom.
- Il file `/rdsdbdata/config/tnsnames.ora` contiene i nomi dei servizi di rete mappati agli indirizzi del protocollo del listener. Prendi nota dei seguenti requisiti e raccomandazioni:
  - Le voci in `tnsnames.ora` con prefisso `rds_custom_` sono riservate a RDS Custom quando si gestiscono le operazioni di replica Oracle.

    Quando si creano voci manuali in `tnsnames.ora`, non usare questo prefisso.
  - In alcuni casi, si potrebbe voler passare o eseguire il failover manualmente o utilizzare tecnologie di failover come Fast-Start Failover (FSFO). In tal caso, assicurarsi di effettuare la sincronizzazione manuale delle voci `tnsnames.ora` dall'istanza database primaria a tutte le istanze in stand-by. Questo suggerimento si applica sia alle repliche Oracle gestite da RDS Custom che alle repliche Oracle esterne.

L'automazione di RDS Custom aggiorna le voci `tnsnames.ora` solo sull'istanza database primaria. Assicurati di eseguire la sincronizzazione anche quando aggiungi o rimuovi una replica Oracle.

Se non si sincronizzano i file `tnsnames.ora` e si esegue uno switchover o un failover manualmente, sull'istanza database primaria Oracle Data Guard potrebbe non essere in grado di comunicare con le repliche Oracle.

Limitazioni esterne per la replica RDS Custom per Oracle

Le repliche esterne RDS Custom per Oracle, che includono repliche on-premise, hanno le seguenti limitazioni:

- RDS Custom per Oracle non rileva le modifiche del ruolo dell'istanza in caso di failover manuale, come FSFO, per le repliche Oracle esterne.

  RDS Custom per Oracle non rileva le modifiche per le repliche gestite. La modifica del ruolo è annotata nel registro eventi. È inoltre possibile visualizzare il nuovo stato utilizzando il describe-db-instances AWS CLI comando.
- RDS Custom per Oracle non rileva un elevato ritardo di replica per le repliche Oracle esterne.

  RDS Custom per Oracle rileva il ritardo per le repliche gestite. L'elevato ritardo di replica produce l'evento `Replication has stopped`. È inoltre possibile visualizzare lo stato della replica utilizzando il describe-db-instances AWS CLI comando, ma potrebbe verificarsi un ritardo nell'aggiornamento.
- RDS Custom per Oracle non promuove automaticamente le repliche Oracle esterne dopo l'eliminazione dell'istanza database primaria.

  La funzione di promozione automatica è disponibile solo per le repliche Oracle gestite. Per informazioni sulla promozione manuale delle repliche Oracle, consulta il whitepaper Abilitazione dell'elevata disponibilità con Data Guard su Amazon RDS Custom per Oracle.

Limitazioni della promozione delle repliche per RDS Custom per Oracle

Promuovere le repliche Oracle gestite da RDS Custom per Oracle equivale a promuovere le repliche gestite da RDS, con alcune differenze. Per le repliche RDS Custom per Oracle sono valide le seguenti limitazioni:

- Non è possibile promuovere una replica mentre RDS Custom per Oracle ne esegue il backup.
- Quando promuovi la replica Oracle, non puoi modificare il periodo di conservazione del backup impostando `0`.
- Non puoi promuovere la tua replica quando il relativo stato non è integro.

  Se si esegue `delete-db-instance` sull'istanza database primaria, RDS Custom per Oracle verifica che ogni replica Oracle gestita sia integra e disponibile per la promozione. Una replica potrebbe non essere idonea per la promozione perché l'automazione è in pausa o si trova al di fuori del perimetro di supporto. In questi casi, RDS Custom per Oracle pubblica un evento che spiega il problema in modo da poter riparare manualmente la replica Oracle.

Linee guida relative alla promozione delle repliche per RDS Custom per Oracle

Quando promuovi una replica, tieni presente le seguenti linee guida:

- Non avviare un failover mentre RDS Custom per Oracle sta promuovendo la replica. In caso contrario, il flusso di lavoro di promozione potrebbe bloccarsi.
- Non eseguire lo switchover sull'istanza database primaria mentre RDS Custom per Oracle sta promuovendo la replica Oracle. In caso contrario, il flusso di lavoro di promozione potrebbe bloccarsi.
- Non arrestare l'istanza database primaria mentre RDS Custom per Oracle sta promuovendo la replica Oracle. In caso contrario, il flusso di lavoro di promozione potrebbe bloccarsi.
- Non tentare di riavviare la replica con l'istanza database appena promossa come destinazione. Dopo aver promosso la replica Oracle, tale replica diventa un'istanza database autonoma e non ha più il ruolo di replica.

Per ulteriori informazioni, consulta Risoluzione dei problemi di promozione delle repliche per RDS Custom per Oracle.

Promozione di una replica RDS Custom per Oracle a istanza database autonoma

Proprio come con RDS per Oracle, puoi promuovere una replica RDS Custom per Oracle a istanza database autonoma. Quando promuovi una replica Oracle, l'istanza database viene riavviata prima di diventare disponibile. Per ulteriori informazioni sulla promozione delle repliche Oracle, consulta Promozione di una replica di lettura a istanza database standalone.

Le fasi seguenti descrivono il processo generale per la promozione di una replica Oracle a istanza database:

1. Interrompi la scrittura di eventuali transazioni sull'istanza database primaria.
2. Attendi che RDS Custom per Oracle applichi tutti gli aggiornamenti alla replica Oracle.
3. Promuovi la tua replica Oracle scegliendo l'opzione Promote sulla console Amazon RDS, il AWS CLI comando o l'`promote-read-replica`operazione dell'API `PromoteReadReplica`Amazon RDS.

Per il completamento della promozione di una replica Oracle sono necessari alcuni minuti. Durante il processo, RDS Custom per Oracle interrompe e riavvia la replica. Al termine del riavvio, la replica Oracle è disponibile come nuova istanza database.

Console

Per promuovere una replica RDS Custom per Oracle a istanza database autonoma

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nella console Amazon RDS scegliere Databases (Database).

   Verrà visualizzato il riquadro Databases (Database). Ogni replica Oracle mostra la voce Replica nella colonna Role (Ruolo).
3. Scegli la replica RDS Custom per Oracle da promuovere.
4. In Actions (Operazioni), seleziona Promote (Promuovi).
5. Nella pagina Promote Oracle replica (Promuovi replica di lettura) immetti il periodo di conservazione dei backup e la finestra di backup per la nuova istanza database promossa. Non è possibile impostare questo valore su 0.
6. Dopo aver selezionato tutte le impostazioni desiderate, scegli Promote Oracle replica (Promuovi replica Oracle).

AWS CLI

Per promuovere la tua replica RDS Custom for Oracle a un'istanza DB autonoma, usa il comando. AWS CLI `promote-read-replica`

Example

PerLinux, o: macOS Unix

```
aws rds promote-read-replica \
--db-instance-identifier my-custom-read-replica \
--backup-retention-period 2 \
--preferred-backup-window 23:00-24:00
```

Per Windows:

```
aws rds promote-read-replica ^
--db-instance-identifier my-custom-read-replica ^
--backup-retention-period 2 ^
--preferred-backup-window 23:00-24:00
```

API RDS

Per promuovere una replica RDS Custom per Oracle a istanza database autonoma, richiama l'operazione `PromoteReadReplica` dell'API Amazon RDS con il parametro `DBInstanceIdentifier` richiesto.

Backup e ripristino di un'istanza database di Amazon RDS Custom per Oracle

Come per Amazon RDS, RDS Custom crea e salva backup automatici dell'istanza database RDS Custom per Oracle durante la finestra di backup dell'istanza database. Puoi inoltre eseguire il backup dell'istanza database manualmente.

La procedura è identica alla creazione di una snapshot di un'istanza database Amazon RDS. La prima snapshot di un'istanza database RDS Custom contiene i dati dell'intera istanza database. Le snapshot successive sono incrementali.

Ripristina le istantanee del DB utilizzando AWS Management Console o. AWS CLI

Argomenti

Creazione di una snapshot RDS Custom per Oracle

RDS Custom per Oracle crea uno snapshot dei volumi di storage dell'istanza database, eseguendo il backup dell'intera istanza database anziché dei singoli database. Quando l'istanza database contiene un database container (CDB), lo snapshot dell'istanza include il CDB root e tutti i PDB.

Quando crei una snapshot RDS Custom per Oracle , specifica di quale istanza database RDS Custom eseguire il backup. Dai un nome alla snapshot database in modo che tu possa ripristinarla in un secondo momento.

Quando crei una snapshot, RDS Custom per Oracle per Oracle crea uno snapshot Amazon EBS per ogni volume collegato all'istanza database. RDS Custom per Oracle utilizza lo snapshot EBS del volume root per registrare una nuova Amazon Machine Image (AMI). Per semplificare l'associazione delle snapshot a un'istanza database specifica, sono contrassegnate con `DBSnapshotIdentifier`, `DbiResourceId` e `VolumeType`.

La creazione di una snapshot DB si traduce in una breve interruzione delle operazioni di I/O. Questa sospensione può durare da pochi secondi a pochi minuti, a seconda delle dimensioni e della classe

dell'istanza database. Il tempo di creazione dello snapshot varia a seconda delle dimensioni del database. Poiché lo snapshot include l'intero volume d'archiviazione, la dimensione dei file, come i file temporanei, influisce sul tempo di creazione dello snapshot. Per ulteriori informazioni sulla creazione di snapshot, consulta Creazione di uno snapshot DB per un'istanza DB Single-AZ.

Creazione di una snapshot RDS Custom per Oracle utilizzando la console o la AWS CLI.

Console

Per creare una snapshot RDS Custom

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, scegli Databases (Database).
3. Nell'elenco di istanze database RDS Custom scegliere l'istanza database per cui si desidera acquisire uno snapshot.
4. Per Actions (Operazioni), selezionare Take snapshot (Acquisisci snapshot).

   Viene visualizzata la finestra Acquisizione di snapshot DB.
5. Per Nome snapshot, inserisci il nome dello snapshot.
6. Seleziona Acquisisci snapshot.

AWS CLI

Puoi creare uno snapshot di un'istanza DB personalizzata RDS utilizzando il comando. create-db-snapshot AWS CLI

Puoi specificare le seguenti opzioni:

- `--db-instance-identifier` – Identificare l'istanza database RDS Custom di cui effettuare il backup
- `--db-snapshot-identifier` – Assegna i nomi alla snapshot RDS Custom in modo che tu possa ripristinarla in un secondo momento

In questo esempio crei uno snapshot database denominata *`my-custom-snapshot`* per un'istanza database RDS Custom denominata *`my-custom-instance`*.

Example

PerLinux, omacOS: Unix

```
aws rds create-db-snapshot \
    --db-instance-identifier my-custom-instance \
    --db-snapshot-identifier my-custom-snapshot
```

Per Windows:

```
aws rds create-db-snapshot ^
    --db-instance-identifier my-custom-instance ^
    --db-snapshot-identifier my-custom-snapshot
```

Ripristino da una snapshot database RDS Custom per Oracle

Quando ripristini un'istanza database RDS Custom per Oracle, devi fornire il nome della snapshot database e il nome della nuova istanza. Non puoi eseguire il ripristino da una snapshot a un'istanza database RDS Custom esistente. Quando esegui il ripristino, viene creata una nuova istanza database RDS Custom per Oracle.

Il processo di ripristino differisce secondo le seguenti modalità dal ripristino in Amazon RDS:

- Prima di ripristinare uno snapshot, RDS Custom per Oracle esegue il backup dei file di configurazione esistenti. Questi file sono disponibili sull'istanza ripristinata nella directory `/rdsdbdata/config/backup`. RDS Custom per Oracle ripristina la snapshot database con i parametri predefiniti e sovrascrive i precedenti file di configurazione del database con quelli esistenti. Pertanto, l'istanza ripristinata non conserva i parametri personalizzati e le modifiche ai file di configurazione del database.
- Il database ripristinato ha lo stesso nome della snapshot. Non puoi specificare un nome diverso. (Per RDS Custom per Oracle, il valore predefinito è `ORCL`.)

Console

Per ripristinare un'istanza database RDS Custom da uno snapshot database

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, selezionare Snapshots (Snapshot).

3. Scegliere la snapshot DB dalla quale effettuare il ripristino.
4. Per Actions (Operazioni), selezionare Restore Snapshot (Ripristina snapshot).
5. Nella pagina Restore DB Instance (Ripristina istanza database), per DB Instance Identifier (Identificatore istanze DB), immettere il nome dell'istanza database RDS Custom ripristinata.
6. Selezionare Ripristina istanza database.

AWS CLI

È possibile ripristinare uno snapshot DB personalizzato RDS utilizzando il comando -db-snapshot. restore-db-instance-from AWS CLI

Se la snapshot da cui si sta ripristinando è per un'istanza database privata, assicurarsi di specificare entrambi i valori corretti `db-subnet-group-name` e `no-publicly-accessible`. In caso contrario, l'istanza database è accessibile pubblicamente per impostazione predefinita. Sono richieste le seguenti opzioni:

- `db-snapshot-identifier` – Identifica la snapshot da cui eseguire il ripristino
- `db-instance-identifier` – Specifica il nome dell'istanza database RDS Custom da creare dalla snapshot database
- `custom-iam-instance-profile`: specifica il profilo di istanza associato all'istanza Amazon EC2 sottostante di un'istanza database RDS Custom.

Il codice seguente ripristina la snapshot denominata `my-custom-snapshot` per `my-custom-instance`.

Example

Per, o: Linux macOS Unix

```
aws rds restore-db-instance-from-db-snapshot \
  --db-snapshot-identifier my-custom-snapshot \
  --db-instance-identifier my-custom-instance \
  --custom-iam-instance-profile AWSRDSCustomInstanceProfileForRdsCustomInstance \
  --no-publicly-accessible
```

Per Windows:

```
aws rds restore-db-instance-from-db-snapshot ^
```

```
  --db-snapshot-identifier my-custom-snapshot ^
  --db-instance-identifier my-custom-instance ^
  --custom-iam-instance-profile AWSRDSCustomInstanceProfileForRdsCustomInstance ^
  --no-publicly-accessible
```

Ripristino di un'istanza RDS Custom per Oracle in un determinato momento

Puoi ripristinare un'istanza database in un punto temporale specifico (PITR), creando una nuova istanza database. Per supportare PITR, le istanze DB devono avere la retention dei backup impostata su un valore diverso da zero.

L'ultimo orario ripristinabile di un'istanza database RDS Custom dipende da diversi fattori, ma generalmente entro 5 minuti dall'orario attuale. Per visualizzare l'ora di ripristino più recente per un'istanza DB, usa il AWS CLI describe-db-instancescomando e guarda il valore restituito nel `LatestRestorableTime` campo per l'istanza DB. Per visualizzare l'ora di ripristino più recente per ogni istanza del DB nella console Amazon RDS, scegliere Backup automatici.

Puoi eseguire il ripristino point-in-time durante il periodo di retention dei backup. Per visualizzare il tempo di ripristino più breve per ogni istanza del DB, scegliere Backup automatici nella console Amazon RDS.

Per informazioni generali su PITR, consulta Ripristino a un'ora specifica per un'istanza database.

Argomenti

- Considerazioni PITR per RDS Custom per Oracle

Considerazioni PITR per RDS Custom per Oracle

In RDS Custom per Oracle, PITR differisce secondo le seguenti importanti modalità da PITR in Amazon RDS:

- Il database ripristinato ha lo stesso nome dell'istanza database di origine. Non puoi specificare un nome diverso. Il valore predefinito è `ORCL`.
- `AWSRDSCustomIamRolePolicy` richiede nuove autorizzazioni. Per ulteriori informazioni, consulta Passaggio 2: aggiungere una politica di accesso a AWSRDSCustomInstanceRoleForRdsCustomInstance.
- Tutte le istanze database RDS Custom per Oracle devono avere la retention dei backup impostata su un valore diverso da zero.

- Se si modifica il fuso orario dell'istanza database o del sistema operativo, PITR potrebbe non funzionare. Per informazioni sulla modifica dei fusi orari, consulta Fuso orario Oracle.
- Se si imposta l'automazione su`ALL_PAUSED`, RDS Custom sospende il caricamento dei redo log file archiviati, inclusi i log creati prima dell'ultima data di ripristino (LRT). Si consiglia di sospendere l'automazione per un breve periodo.

  Per illustrare, supponiamo che la tua LRT sia 10 minuti fa. Metti in sospensione l'automazione. Durante la pausa, RDS Custom non carica i log di ripristino archiviati. Se l'istanza database si arresta in modo anomalo, è possibile eseguire il ripristino solo a un orario prima dell'LRT esistente al momento della pausa. Quando si riprende l'automazione, RDS Custom riprende il caricamento dei registri. L'LRT avanza. Si applicano regole PITR normali.
- In RDS Custom, è possibile specificare manualmente un numero arbitrario di ore per conservare i log di ripristino archiviati prima che RDS Custom li elimini dopo il caricamento. Specifica il numero di ore come segue:
  1. Crea un file di testo denominato `/opt/aws/rdscustomagent/config/redo_logs_custom_configuration.json`.
  2. Aggiungi un oggetto JSON con il formato seguente: `{"archivedLogRetentionHours" : "`*`num_of_hours`*`"}`. Il numero deve essere un numero intero compreso tra 1 e 840.
- Supponi di collegare un database non CDB a un database container (CDB) come PDB e quindi prova PITR. L'operazione ha esito positivo solo se in precedenza è stato eseguito il backup del PDB. Dopo aver creato o modificato un PDB, ti consigliamo di eseguire sempre il backup.
- Si consiglia di non personalizzare i parametri di inizializzazione del database. Ad esempio, la modifica dei seguenti parametri influisce su PITR:
  - `CONTROL_FILE_RECORD_KEEP_TIME` influisce sulle regole per il caricamento e l'eliminazione dei registri.
  - `LOG_ARCHIVE_DEST_`*`n`* non supporta più destinazioni.
  - `ARCHIVE_LAG_TARGET`influisce sull'ultima ora di ripristino. `ARCHIVE_LAG_TARGET`è impostato su `300` perché l'obiettivo del punto di ripristino (RPO) è di 5 minuti. Per raggiungere questo obiettivo, RDS cambia il redo log online ogni 5 minuti e lo archivia in un bucket Amazon S3. Se la frequenza del log switch causa un problema di prestazioni per il database RDS Custom for Oracle, puoi scalare l'istanza DB e lo storage su uno con IOPS e throughput più elevati. Se necessario per il piano di ripristino, è possibile regolare l'impostazione del parametro di `ARCHIVE_LAG_TARGET` inizializzazione su un valore compreso tra 60 e 7200.
- Se si personalizzano i parametri di inizializzazione del database, si consiglia vivamente di personalizzare solo quanto segue:

- `COMPATIBLE`
- `MAX_STRING_SIZE`
- `DB_FILES`
- `UNDO_TABLESPACE`
- `ENABLE_PLUGGABLE_DATABASE`
- `CONTROL_FILES`
- `AUDIT_TRAIL`
- `AUDIT_TRAIL_DEST`

Per tutti gli altri parametri di inizializzazione, RDS Custom ripristina i valori predefiniti. Se modifichi un parametro che non è presente nell'elenco precedente, potrebbe avere un effetto negativo sul PITR e portare a risultati imprevedibili. Ad esempio, `CONTROL_FILE_RECORD_KEEP_TIME` influisce sulle regole per il caricamento e l'eliminazione dei registri.

È possibile ripristinare un'istanza DB personalizzata RDS in un determinato momento utilizzando AWS Management Console, the o l' AWS CLI API RDS.

Console

Per ripristinare un'istanza database RDS Custom un punto temporale specifico

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, selezionare Automated backups (Backup automatici).
3. Scegli l'istanza database RDS Custom da ripristinare.
4. In Actions (Operazioni), scegli Restore to point in time (Ripristina a un istante temporale).

   Viene visualizzata la finestra Restore to point in time (Ripristina a un istante temporale).
5. Scegliere Latest restorable time (Ultimo orario di ripristino) per eseguire il ripristino in base al momento più recente oppure scegliere Custom (Personalizzato) per scegliere una data e un'ora.

   Se scegli Personalizzato, specifica la data e l'ora in cui desideri ripristinare l'istanza.

   Gli orari vengono visualizzati nel fuso orario locale, indicato come un offset dell'ora UTC (Coordinated Universal Time). Ad esempio, UTC-5 è l'orario standard degli Stati Uniti orientali/ora legale degli Stati Uniti centrali.

6. Per DB Instance Identifier (Identificatore istanze database), inserire il nome dell'istanza database RDS Custom di destinazione ripristinata. Il nome deve essere univoco.
7. Scegli altre opzioni in base alle esigenze, ad esempio la classe di istanza database.
8. Scegli Restore to point in time (Ripristina per punto nel tempo).

AWS CLI

Puoi ripristinare un'istanza DB a un'ora specificata utilizzando il point-in-time AWS CLI comando restore-db-instance-to- per creare una nuova istanza DB personalizzata RDS.

Utilizzare una delle opzioni seguenti per specificare il backup da cui effettuare il ripristino:

- `--source-db-instance-identifier` *`mysourcedbinstance`*
- `--source-dbi-resource-id` *`dbinstanceresourceID`*
- `--source-db-instance-automated-backups-arn` *`backupARN`*

L'opzione `custom-iam-instance-profile` è obbligatoria.

Il seguente esempio ripristina `my-custom-db-instance` a una nuova istanza database denominata `my-restored-custom-db-instance`, a partire dal tempo specificato.

Example

Per LinuxmacOS, oUnix:

```
aws rds restore-db-instance-to-point-in-time \
    --source-db-instance-identifier my-custom-db-instance\
    --target-db-instance-identifier my-restored-custom-db-instance \
    --custom-iam-instance-profile AWSRDSCustomInstanceProfileForRdsCustomInstance \
    --restore-time 2022-10-14T23:45:00.000Z
```

Per Windows:

```
aws rds restore-db-instance-to-point-in-time ^
    --source-db-instance-identifier my-custom-db-instance ^
    --target-db-instance-identifier my-restored-custom-db-instance ^
    --custom-iam-instance-profile AWSRDSCustomInstanceProfileForRdsCustomInstance ^
    --restore-time 2022-10-14T23:45:00.000Z
```

Eliminazione di una snapshot RDS Custom per Oracle

Puoi eliminare snapshot database da RDS Custom per Oracle quando non ti servono più. La procedura di eliminazione è la stessa per le istanze database Amazon RDS e RDS Custom.

Le snapshot Amazon EBS per i volumi binari e root rimangono nel tuo account per un periodo più lungo perché potrebbero essere collegate ad alcune istanze in esecuzione nel tuo account o ad altre istantanee RDS Custom per Oracle. Queste snapshot EBS vengono eliminate automaticamente dopo che non sono più correlate a risorse RDS Custom per Oracle esistenti (istanze database o backup).

Console

Per eliminare una snapshot di un'istanza database RDS Custom

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel pannello di navigazione, selezionare Snapshots (Snapshot).
3. Scegliere la snapshot DB da eliminare.
4. Per Actions (Operazioni), scegliere Delete Snapshot (Elimina snapshot).
5. Nella pagina di conferma, scegliere Delete (Elimina).

AWS CLI

Per eliminare uno snapshot RDS Custom, usa il comando. AWS CLI delete-db-snapshot

Si richiede la seguente opzione:

- `--db-snapshot-identifier` – La snapshot da eliminare

L'esempio seguente elimina la snapshot database `my-custom-snapshot`.

Example

PerLinux, omacOS: Unix

```
aws rds delete-db-snapshot \
  --db-snapshot-identifier my-custom-snapshot
```

Per Windows:

```
aws rds delete-db-snapshot ^
  --db-snapshot-identifier my-custom-snapshot
```

Eliminazione di backup automatici RDS Custom per Oracle

Puoi eliminare i backup automatici mantenuti per RDS Custom per Oracle quando non servono più. La procedura è la stessa della procedura per l'eliminazione dei backup Amazon RDS.

Console

Per eliminare i backup automatici mantenuti

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, selezionare Automated backups (Backup automatici).
3. ScegliereRetained (Mantenuti).
4. Scegliere il backup automatico mantenuto da eliminare.
5. In Actions (Azioni), selezionare Delete (Elimina).
6. Nella pagina di conferma, immetti **`delete me`** e seleziona Elimina.

AWS CLI

Puoi eliminare un backup automatico conservato utilizzando il AWS CLI comando delete-db-instance-automated -backup.

La seguente opzione viene utilizzata per eliminare un backup automatico mantenuto:

- `--dbi-resource-id` – L'identificatore della risorsa per l'istanza database RDS Custom di origine.

  È possibile trovare l'identificatore di risorsa per l'istanza DB di origine di un backup automatizzato mantenuto utilizzando il comando -backups. AWS CLI describe-db-instance-automated

Il seguente esempio elimina il backup automatico mantenuto con l'identificatore della risorsa di istanza DB source `custom-db-123ABCEXAMPLE`.

Example

PerLinux, o: macOS Unix

```
aws rds delete-db-instance-automated-backup \
    --dbi-resource-id custom-db-123ABCEXAMPLE
```

Per Windows:

```
aws rds delete-db-instance-automated-backup ^
    --dbi-resource-id custom-db-123ABCEXAMPLE
```

Utilizzo dei gruppi di opzioni in RDS Custom for Oracle

RDS Custom utilizza i gruppi di opzioni per abilitare e configurare funzionalità aggiuntive. Un gruppo di opzioni specifica le funzionalità, denominate opzioni, disponibili per un'istanza DB RDS Custom for Oracle. Le opzioni possono includere impostazioni che specificano il funzionamento delle opzioni stesse. Quando si associa un'istanza RDS Custom for Oracle DB a un gruppo di opzioni, le opzioni e le impostazioni delle opzioni specificate vengono abilitate per questa istanza. Per informazioni generali sui gruppi di opzioni in Amazon RDS, consultaUso di gruppi di opzioni.

Argomenti

Panoramica dei gruppi di opzioni in RDS Custom for Oracle

Per abilitare queste opzioni per database Oracle, dovrai aggiungerle a un gruppo di opzioni e quindi associare il gruppo di opzioni all'istanza database. Per ulteriori informazioni, consulta Uso di gruppi di opzioni.

Argomenti

Riepilogo delle opzioni di RDS Custom for Oracle

RDS Custom for Oracle supporta le seguenti opzioni per un'istanza DB.

Opzione	ID opzione	Descrizione
Fuso orario Oracle	`Timezone`	Il fuso orario utilizzato dall'istanza DB di RDS Custom for Oracle.

Passaggi di base per aggiungere un'opzione a un'istanza RDS Custom for Oracle DB

La procedura generale per aggiungere un'opzione all'istanza DB RDS Custom for Oracle è la seguente:

1. Creare un nuovo gruppo di opzioni oppure copiare o modificare un gruppo di opzioni esistente.
2. Aggiungere l'opzione al gruppo di opzioni.
3. Associa il gruppo di opzioni all'istanza DB quando la crei o la modifichi.

Creazione di un gruppo di opzioni per in RDS Custom for Oracle

Puoi creare un nuovo gruppo di opzioni che utilizzi le impostazioni del gruppo di opzioni predefinito. Quindi, aggiungi una o più opzioni al nuovo gruppo di opzioni. In alternativa, se esiste già un gruppo di opzioni, puoi copiarlo con tutte le opzioni in un nuovo gruppo di opzioni. Per informazioni su come copiare un gruppo di opzioni, vedereCopia di un gruppo di opzioni.

I gruppi di opzioni predefiniti per RDS Custom for Oracle sono i seguenti:

- `default:custom-oracle-ee`
- `default:custom-oracle-se2`
- `default:custom-oracle-ee-cdb`
- `default:custom-oracle-se2-cdb`

Quando si crea un gruppo di opzioni, le impostazioni derivano dal gruppo di opzioni predefinito. Dopo aver aggiunto l'`TIME_ZONE`opzione, è possibile associare il gruppo di opzioni all'istanza DB.

Console

Un metodo per creare un gruppo di opzioni consiste nell'usare la AWS Management Console.

Per creare un nuovo gruppo di opzioni tramite la console

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel pannello di navigazione scegliere Option groups (Gruppi di opzioni).
3. Seleziona Create group (Crea gruppo).
4. Nella finestra Create option group (Crea gruppo di opzioni) eseguire queste operazioni:

a. Per Nome, digita un nome per il gruppo di opzioni univoco all'interno del tuo AWS account. Il nome può includere solo lettere, cifre e trattini.

b. Per Description (Descrizione) digitare una breve descrizione del gruppo di opzioni. La descrizione viene usata per la visualizzazione.

c. Per Engine, scegli uno dei seguenti motori RDS Custom for Oracle DB:

- custom-oracle-ee
- custom-oracle-se2
- custom-oracle-ee-cdb
- custom-oracle-se2 cdb

d. Per la versione principale del motore, scegli una versione principale del motore supportata da RDS Custom for Oracle. Per ulteriori informazioni, consulta Regioni e motori DB supportati per RDS Custom for Oracle.

5. Per continuare, scegliere Create (Crea). Per annullare l'operazione, invece, scegliere Cancel (Annulla).

AWS CLI

Per creare un gruppo di opzioni, utilizzare il AWS CLI `create-option-group`comando con i seguenti parametri obbligatori.

- `--option-group-name`
- `--engine-name`
- `--major-engine-version`
- `--option-group-description`

Example

L'esempio seguente crea un gruppo di opzioni denominato `testoptiongroup`, associato al motore di database Oracle Enterprise Edition. La descrizione è racchiusa tra virgolette.

Per LinuxmacOS, oUnix:

```
aws rds create-option-group \
    --option-group-name testoptiongroup \
    --engine-name custom-oracle-ee-cdb \
```

```
    --major-engine-version 19 \
    --option-group-description "Test option group for a Custom Oracle CDB"
```

Per Windows:

```
aws rds create-option-group ^
    --option-group-name testoptiongroup ^
    --engine-name custom-oracle-ee-cdb ^
    --major-engine-version 19 ^
    --option-group-description "Test option group for a Custom Oracle CDB"
```

API RDS

Per creare un gruppo di opzioni, chiamare l'operazione API Amazon RDS `CreateOptionGroup`.

Associazione di un gruppo di opzioni a un'istanza DB RDS Custom for Oracle

È possibile associare il gruppo di opzioni a un'istanza database nuova o esistente:

- Per una nuova istanza DB, applica il gruppo di opzioni quando crei l'istanza. Per ulteriori informazioni, consulta Creazione di un'istanza database RDS Custom per Oracle.
- Per un'istanza database esistente, applicare il gruppo di opzioni modificando l'istanza e collegando il nuovo gruppo di opzioni. Per ulteriori informazioni, consulta Modifica dell'istanza database RDS Custom per Oracle.

Fuso orario Oracle

Per modificare il fuso orario del sistema utilizzato dall'istanza DB RDS Custom for Oracle, utilizza l'opzione del fuso orario. Ad esempio, potrebbe essere necessario modificare il fuso orario di un'istanza di database in modo che sia compatibile con un ambiente locale o con un'applicazione legacy. L'opzione del fuso orario modifica il fuso orario a livello di host. La modifica del fuso orario influisce su tutti i valori e su tutte le colonne della data, inclusi `SYSDATE` e `SYSTIMESTAMP`.

Argomenti

- Impostazioni delle opzioni di fuso orario in RDS Custom for Oracle
- Fusi orari disponibili in RDS Custom for Oracle
- Considerazioni sull'impostazione del fuso orario in RDS Custom for Oracle
- Limitazioni per l'impostazione del fuso orario in RDS Custom for Oracle
- Aggiungere l'opzione del fuso orario a un gruppo di opzioni

- Rimozione dell'opzione del fuso orario

Impostazioni delle opzioni di fuso orario in RDS Custom for Oracle

Amazon RDS supporta le seguenti impostazioni per l'opzione del fuso orario.

Impostazione opzioni	Valori validi	Descrizione
`TIME_ZONE`	Uno dei fusi orari disponibili. Per l'elenco completo, consulta Fusi orari disponibili in RDS Custom for Oracle.	Il nuovo fuso orario per l'istanza di database.

Fusi orari disponibili in RDS Custom for Oracle

È possibile utilizzare i seguenti valori per l'opzione del fuso orario.

Zona	Time zone (Fuso orario)
Africa	Africa/Cairo, Africa/Casablanca, Africa/Harare, Africa/Lagos, Africa/Luanda, Africa/Monrovia, Africa/Nairobi, Africa/Tripoli, Africa/Windhoek
America	America/Araguaina, America/Argentina/Buenos_Aires, America/Asuncion, America/Bogota, America/Caracas, America/Chicago, America/Chihuahua, America/Cuiaba, America/Denver, America/Detroit, America/Fortaleza, America/Godthab, America/Guatemala, America/Halifax, America/Lima, America/Los_Angeles, America/Manaus, America/Matamoros, America/Mexico_City, America/Monterrey, America/Montevideo, America/New_York, America/Phoenix, America/Santiago, America/Sao_Paulo, America/Tijuana, America/Toronto
Asia	Asia/Amman, Asia/Ashgabat, Asia/Baghdad, Asia/Baku, Asia/Bangkok, Asia/Beirut, Asia/Calcutta, Asia/Damasco, Asia/Dhaka, Asia/Hong_Kong, Asia/Irkutsk, Asia/Giacarta, Asia/Gerusalemme, Asia/Kabul, Asia/Karachi, Asia/Kathmandu, Asia/Kolkata, Asia/Krasnoyarsk, Asia/Magadan, Asia/Manila, Asia/Muscat, Asia/Novosibirsk, Asia/Rangoon, Asia/Riyadh, Asia/Seul, Asia/Shan

Zona	Time zone (Fuso orario)
	ghai, Asia/Singapore, Asia/Taipei, Asia/Tehran, Asia/Tokyo, Asia/Ulaanbaatar, Asia/Vladivostok, Asia/Yakutsk, Asia/Yerevan
Atlantico	Atlantico/Azzorre, Atlantico/Cape_Verde
Australia	Australia/Adelaide, Australia/Brisbane, Australia/Darwin, Australia/Eucla, Australia/Hobart, Australia/Lord_Howe, Australia/Perth, Australia/Sydney
Brasile	Brasile/, Brasile/Est DeNoronha
Canada	Canada/Newfoundland, Canada/Saskatchewan
ecc	Ecc./GMT-3
Europa	Europa/Amsterdam, Europa/Atene, Europa/Berlino, Europa/Dublino, Europa/Helsinki, Europa/Kaliningrad, Europa/Londra, Europa/Madrid, Europa/Mosca, Europa/Parigi, Europa/Praga, Europa/Roma, Europa/Sarajevo
Pacifico	Pacifico/Apia, Pacifico/Auckland, Pacifico/Chatham, Pacifico/Fiji, Pacifico/Guam, Pacifico/Honolulu, Pacifico/Kiritimati, Pacifico/Marquesas, Pacifico/Samoa, Pacifico/Tongatapu, Pacifico/Wake
USA	Stati Uniti/Alaska, Stati Uniti/Centrali, Stati Uniti/Est-Indiana, Stati Uniti/Orientali, Stati Uniti/Pacifico
UTC	UTC

Considerazioni sull'impostazione del fuso orario in RDS Custom for Oracle

Se scegli di impostare il fuso orario per la tua istanza DB, considera quanto segue:

- Quando aggiungi l'opzione del fuso orario, si verifica una breve interruzione mentre l'istanza di database viene automaticamente riavviata.
- Se imposti accidentalmente il fuso orario in modo errato, dovrai ripristinare l'istanza database alle impostazioni del fuso orario precedente. Per questo motivo, ti consigliamo vivamente di utilizzare una delle seguenti strategie prima di aggiungere l'opzione del fuso orario all'istanza:

  - Se l'istanza DB RDS Custom for Oracle utilizza il gruppo di opzioni predefinito, scatta un'istantanea dell'istanza DB. Per ulteriori informazioni, consulta Creazione di una snapshot RDS Custom per Oracle.
  - Se l'istanza DB utilizza attualmente un gruppo di opzioni non predefinito, scatta uno snapshot dell'istanza DB, quindi crea un nuovo gruppo di opzioni con l'opzione del fuso orario.
- Ti consigliamo vivamente di eseguire il backup dell'istanza DB manualmente dopo aver applicato l'`Timezone`opzione.
- Ti consigliamo vivamente di testare l'opzione del fuso orario su un'istanza DB di test prima di aggiungerla a un'istanza DB di produzione. L'aggiunta dell'opzione del fuso orario può causare problemi con le tabelle che utilizzano la data di sistema per aggiungere date o orari. Analizza i tuoi dati e le tue applicazioni per valutare l'impatto della modifica del fuso orario.

Limitazioni per l'impostazione del fuso orario in RDS Custom for Oracle

Nota i seguenti limiti:

- Non puoi modificare il fuso orario direttamente sull'host senza spostarlo al di fuori del perimetro di supporto. Per modificare il fuso orario del database, devi creare un gruppo di opzioni.
- Poiché l'opzione del fuso orario è un'opzione persistente (ma non un'opzione permanente), non è possibile effettuare le seguenti operazioni:
  - Non è possibile rimuovere l'opzione da un gruppo di opzioni dopo averla aggiunta.
  - Non è possibile sostituire l'impostazione del fuso orario dell'opzione con un altro fuso orario.
- Non è possibile associare più gruppi di opzioni alla propria istanza DB RDS Custom for Oracle.
- Non è possibile impostare il fuso orario per i singoli PDB all'interno di un CDB.

Aggiungere l'opzione del fuso orario a un gruppo di opzioni

I gruppi di opzioni predefiniti per RDS Custom for Oracle sono i seguenti:

- `default:custom-oracle-ee`
- `default:custom-oracle-se2`
- `default:custom-oracle-ee-cdb`
- `default:custom-oracle-se2-cdb`

Quando si crea un gruppo di opzioni, le impostazioni derivano dal gruppo di opzioni predefinito. Per informazioni generali sui gruppi di opzioni in Amazon RDS, consultaUso di gruppi di opzioni.

Console

Per aggiungere l'opzione del fuso orario a un gruppo di opzioni

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione scegliere Option groups (Gruppi di opzioni).
3. Selezionare il gruppo di opzioni che si vuole modificare, quindi scegliere Add Option (Aggiungi opzione).
4. Nella finestra Add option (Aggiungi opzione) eseguire queste operazioni:
   a. Scegli il fuso orario.
   b. Nelle impostazioni delle opzioni, scegli un fuso orario.
   c. Per abilitare l'opzione su tutte le istanze DB RDS Custom for Oracle associate non appena la aggiungi, per Applica immediatamente, scegli Sì. Se scegli No (impostazione predefinita), l'opzione viene abilitata per ogni istanza DB associata durante la successiva finestra di manutenzione.
   d.

   > **Important**
   >
   > Se aggiungi l'opzione del fuso orario a un gruppo di opzioni esistente già associato a una o più istanze di database, si verifica una breve interruzione mentre tutte le istanze di database vengono riavviate automaticamente.

5. Dopo aver selezionato le impostazioni desiderate, selezionare Add Option (Aggiungi opzione).
6. Esegui il backup delle istanze DB RDS Custom for Oracle i cui fusi orari sono stati aggiornati. Per ulteriori informazioni, consulta Creazione di una snapshot RDS Custom per Oracle.

AWS CLI

L'esempio seguente utilizza il comando AWS CLI add-option-to-option-group per aggiungere l'`Timezone`opzione e l'impostazione dell'`TIME_ZONE`opzione a un gruppo di opzioni chiamato. `testoptiongroup` Il fuso orario è impostato su `America/Los_Angeles`.

Per LinuxmacOS, oUnix:

```
aws rds add-option-to-option-group \
    --option-group-name "testoptiongroup" \
    --options "OptionName=Timezone,OptionSettings=[{Name=TIME_ZONE,Value=America/
Los_Angeles}]" \
    --apply-immediately
```

Per Windows:

```
aws rds add-option-to-option-group ^
    --option-group-name "testoptiongroup" ^
    --options "OptionName=Timezone,OptionSettings=[{Name=TIME_ZONE,Value=America/
Los_Angeles}]" ^
    --apply-immediately
```

Rimozione dell'opzione del fuso orario

L'opzione del fuso orario è un'opzione persistente, ma non permanente. Non è possibile rimuovere l'opzione da un gruppo di opzioni dopo averla aggiunta. Per dissociare il vecchio gruppo di opzioni dall'istanza DB:

1. Crea un nuovo gruppo di opzioni con un'`Timezone`opzione aggiornata.
2. Associate il nuovo gruppo di opzioni all'istanza DB quando modificate l'istanza.

Migrazione di un database on-premise a RDS Custom per Oracle

Prima di eseguire la migrazione di un database Oracle on premise a RDS Custom per Oracle, devi considerare i seguenti fattori:

- La durata del tempo di inattività che l'applicazione è in grado di gestire
- La dimensione del database di origine
- La connettività di rete
- Un requisito per un piano di fallback
- La versione del database Oracle di origine e di destinazione e i tipi di sistema operativo dell'istanza DB
- Strumenti di replica disponibili, come AWS Database Migration Service, Oracle GoldenGate o strumenti di replica di terze parti

In base a questi fattori, puoi scegliere la migrazione fisica, la migrazione logica o una combinazione di questi due tipi. Se scegli la migrazione fisica, puoi utilizzare le seguenti tecniche:

Duplicazione RMAN

La duplicazione del database attivo non richiede il backup del database di origine. Duplica il database di origine attivo sull'host di destinazione copiando i file del database in rete nell'istanza ausiliaria. Il comando RMAN `DUPLICATE` copia i file richiesti, ad esempio copie di immagini o set di backup. Per ulteriori informazioni su questa tecnica, consulta il post del blog AWS relativo alla migrazione fisica dei database Oracle su Amazon RDS Custom utilizzando la duplicazione RMAN.

Oracle Data Guard

Con questa tecnica, è possibile eseguire il backup di un database on-premise primario e copiare i backup in un bucket Amazon S3. Sarà quindi possibile copiare i backup nell'istanza DB in standby RDS Custom per Oracle. Dopo aver eseguito la configurazione necessaria, si passa manualmente dal database principale al database in standby RDS Custom per Oracle. Per ulteriori informazioni su questa tecnica, consulta il post del blog AWS relativo alla migrazione fisica dei database Oracle su Amazon RDS Custom utilizzando Data Guard.

Per informazioni generali sull'importazione logica dei dati in RDS per Oracle, consulta Importazione di dati in Oracle in Amazon RDS.

Aggiornamento di un'istanza database per Amazon RDS Custom for Oracle

Puoi aggiornare un'istanza database Amazon RDS Custom modificandola per utilizzare una nuova versione del motore personalizzato (CEV). Per informazioni generali sugli aggiornamenti, consulta Aggiornamento della versione del motore di un'istanza database.

Argomenti

Panoramica sugli aggiornamenti in RDS Custom per Oracle

Con RDS Custom per Oracle, puoi applicare le patch al database Oracle o al sistema operativo dell'istanza database creando nuovi CEV e quindi modificando l'istanza per utilizzare il nuovo CEV.

Argomenti

Opzioni di aggiornamento per CEV

Quando si crea un CEV per un aggiornamento, sono disponibili le seguenti opzioni che si escludono a vicenda:

Solo database

Riutilizza l'Amazon Machine Image (AMI) attualmente usata dall'istanza database, ma specifica file binari di database diversi. RDS Custom alloca un nuovo volume binario e quindi lo collega

all'istanza Amazon EC2 esistente. RDS Custom sostituisce l'intero volume del database con un nuovo volume che utilizza la versione del database di destinazione.

Solo sistema operativo

Riutilizza i file binari del database attualmente usati dall'istanza database, ma specifica un'AMI diversa. RDS Custom alloca una nuova istanza Amazon EC2, quindi collega il volume binario esistente alla nuova istanza. Il volume del database esistente viene mantenuto.

Per aggiornare sia il sistema operativo che il database, è necessario aggiornare il CEV due volte. È possibile aggiornare il sistema operativo e quindi il database oppure aggiornare il database e quindi il sistema operativo.

> **Warning**
>
> Quando applichi una patch al sistema operativo, i dati del volume root e qualsiasi personalizzazione esistente del sistema operativo andranno persi. Pertanto, si consiglia vivamente di non utilizzare il volume root per le installazioni o per l'archiviazione di dati o file permanenti. Si consiglia inoltre di eseguire il backup dei dati prima dell'aggiornamento.

Applicazione di patch senza CEV

Si consiglia vivamente di eseguire l'aggiornamento dell'istanza database RDS Custom per Oracle utilizzando i CEV. L'automazione RDS Custom per Oracle sincronizza i metadati delle patch con il file binario del database sull'istanza database.

In circostanze speciali, RDS Custom supporta l'applicazione di una singola patch del database direttamente all'istanza Amazon EC2 sottostante utilizzando l'utilità OPatch. Un caso d'uso valido potrebbe essere una patch del database da applicare immediatamente, ma il team di RDS Custom sta aggiornando la funzionalità CEV, causando un ritardo. Per applicare manualmente una patch del database, procedi nel seguente modo:

1. Sospendi l'automazione RDS Custom.
2. Applica la patch ai file binari del database sull'istanza Amazon EC2.
3. Riprendere l'automazione RDS Custom.

Uno svantaggio della tecnica precedente è che è necessario applicare la patch del database manualmente a ogni istanza da aggiornare. Al contrario, quando si crea un nuovo CEV, è possibile creare o aggiornare più istanze database utilizzando lo stesso CEV.

Procedure generali per applicare una patch all'istanza database con un CEV

Sia che applichi una patch al sistema operativo o al database, attieniti alla procedura di base riportata di seguito:

1. A seconda se la patch viene applicata al database o al sistema operativo, crea un CEV che contenga uno dei seguenti elementi:
   - L'aggiornamento della versione di Oracle Database che desideri applicare all'istanza database.
   - Un'AMI diversa, la più recente disponibile o quella specificata dall'utente, e un CEV esistente da utilizzare come origine.

   Seguire la procedura riportata in Creazione di un CEV.
2. (Facoltativo per l'applicazione della patch del database) Controlla gli aggiornamenti della versione del motore disponibili eseguendo `describe-db-engine-versions`.
3. Avvia il processo di applicazione della patch eseguendo `modify-db-instance`.

   Lo stato dell'istanza a cui viene applicata la patch varia come segue:
   - Mentre RDS applica la patch al database, lo stato dell'istanza database cambia in Aggiornamento in corso.
   - Mentre RDS applica la patch al sistema operativo, lo stato dell'istanza database cambia in Modifica in corso.

   Quando l'istanza database ha lo stato Disponibile, l'applicazione della patch è completata.
4. Verifica che l'istanza database utilizzi il nuovo CEV eseguendo `describe-db-instances`.

Requisiti per gli aggiornamenti di RDS Custom per Oracle

Quando si aggiorna un'istanza database RDS Custom per Oracle a una CEV di destinazione, accertati che siano soddisfatti i seguenti requisiti:

- La CEV di destinazione su cui si esegue l'aggiornamento deve esistere.

- È necessario aggiornare il sistema operativo o il database con un'unica operazione. L'aggiornamento sia del sistema operativo che del database in una singola chiamata API non è supportato.
- La CEV di destinazione deve utilizzare le impostazioni dei parametri di installazione presenti nel manifesto della CEV corrente. Ad esempio, non è possibile eseguire l'aggiornamento di un database che usa la home Oracle predefinita a una CEV che utilizza una home Oracle non predefinita.
- Per gli aggiornamenti del database, il CEV di destinazione deve utilizzare una nuova versione secondaria del database e non una nuova versione principale. Ad esempio, non è possibile eseguire l'aggiornamento da una CEV di Oracle Database 12c a una CEV di Oracle Database 19c. Puoi tuttavia eseguire l'aggiornamento dalla versione 21.0.0.0.ru-2023-04.rur-2023-04.r1 alla versione 21.0.0.0.ru-2023-07.rur-2023-07.r1.
- Per gli aggiornamenti del sistema operativo, il CEV di destinazione deve utilizzare un'AMI diversa ma avere la stessa versione principale.

Considerazioni per gli aggiornamenti del database RDS Custom for Oracle

Se intendi aggiornare il database, considera quanto segue:

- Quando si aggiornano i binari del database nell'istanza database primaria, RDS Custom per Oracle aggiorna automaticamente le repliche di lettura. Quando si aggiorna il sistema operativo, le repliche di lettura devono essere aggiornate manualmente.
- Quando si aggiorna un database contenitore (CDB) a una nuova versione del database, RDS Custom for Oracle verifica che tutti i PDB siano aperti o possano essere aperti. Se queste condizioni non sono soddisfatte, RDS Custom interrompe il controllo e riporta il database allo stato originale senza tentare l'aggiornamento. Se le condizioni sono soddisfatte, RDS Custom corregge prima la root del CDB e poi corregge tutti gli altri PDB (incluso `PDB$SEED`) in parallelo.

  Al termine dell'applicazione delle patch, RDS Custom tenta di aprire tutti i PDB. Se alcuni PDB non si aprono, viene generato il seguente evento: `The following PDBs failed to open: list-of-PDBs`. Se RDS Custom non riesce a applicare una patch alla root del CDB o a qualsiasi PDB, l'istanza viene messa nello stato `PATCH_DB_FAILED`.
- Potresti voler eseguire contemporaneamente un aggiornamento della versione principale del database e una conversione di una versione non CDB in CDB. In questo caso, ti consigliamo di procedere come segue:

  1. Crea una nuova istanza DB RDS Custom for Oracle che utilizza l'architettura multitenant Oracle.

2. Collega un non CDB alla root del CDB, creandolo come PDB. Assicurati che la versione non CDB sia uguale alla versione principale del tuo CDB.
3. Converti il tuo PDB eseguendo lo script Oracle SQL. `noncdb_to_pdb.sql`
4. Convalida l'istanza CDB.
5. Aggiorna l'istanza CDB.

Considerazioni sugli aggiornamenti di RDS Custom per Oracle OS

Quando pianifichi un aggiornamento del sistema operativo, tieni presente quanto segue:

- Non puoi fornire la tua AMI da utilizzare in un RDS Custom per Oracle CEV. È possibile specificare l'AMI predefinito o un AMI utilizzato in precedenza da un RDS Custom per Oracle CEV.

  > **Note**
  >
  > RDS Custom for Oracle rilascia una nuova AMI predefinita quando vengono scoperte vulnerabilità ed esposizioni comuni. Non è disponibile o garantito alcun programma fisso. RDS Custom for Oracle tende a pubblicare una nuova AMI predefinita ogni 30 giorni.

- Quando si aggiorna il sistema operativo nell'istanza DB principale, è necessario aggiornare manualmente le repliche di lettura associate.
- Riserva una capacità di calcolo di Amazon EC2 sufficiente per il tipo di istanza nella tua AZ prima di iniziare ad applicare le patch al sistema operativo.

  Quando crei una prenotazione della capacità, specifichi la zona di disponibilità, il numero di istanze e gli attributi delle istanza (incluso il tipo di istanza). Ad esempio, se l'istanza database utilizza il tipo di istanza EC2 sottostante r5.large, è consigliabile prenotare la capacità EC2 per r5.large nella zona di disponibilità. Durante l'applicazione della patch al sistema operativo, RDS Custom crea un nuovo host di tipo db.r5.large, che può bloccarsi se la zona di disponibilità non dispone della capacità EC2 per questo tipo di istanza. Se si prenota la capacità EC2, si riduce il rischio di blocco della patch causato da vincoli di capacità. Per ulteriori informazioni, consulta Prenotazione di capacità on demand nella Guida per l'utente per istanze Linux di Amazon EC2.
- Esegui il backup dell'istanza DB prima di aggiornarne il sistema operativo. L'aggiornamento rimuove i dati del volume root e tutte le personalizzazioni esistenti del sistema operativo.
- Nel modello di responsabilità condivisa, sei responsabile di mantenere aggiornato il tuo sistema operativo. RDS Custom for Oracle non impone quali patch applicare al sistema operativo. Se

il tuo RDS Custom for Oracle funziona, puoi utilizzare l'AMI associata a questo CEV a tempo indeterminato.

Visualizzazione di destinazioni di aggiornamento CEV valide per le istanze database RDS Custom per Oracle

È possibile visualizzare i CEV esistenti sulla pagina Versioni motore personalizzate in AWS Management Console.

È inoltre possibile utilizzare il describe-db-engine-versions AWS CLI comando per trovare CEV validi da utilizzare quando si aggiornano le istanze DB, come illustrato nell'esempio seguente. Questo esempio presuppone che sia stata creata un'istanza database utilizzando la versione del motore `19.my_cev1` e che le versioni di aggiornamento `19.my_cev2` e `19.my_cev` siano presenti.

```
aws rds describe-db-engine-versions --engine custom-oracle-ee --engine-version
 19.my_cev1
```

L'output è simile a quello riportato di seguito. Il campo `ImageId` mostra l'ID AMI.

```
{
    "DBEngineVersions": [
        {
            "Engine": "custom-oracle-ee",
            "EngineVersion": "19.my_cev1",
            ...
            "Image": {
                "ImageId": "ami-2345",
                "Status": "active"
            },
            "DBEngineVersionArn": "arn:aws:rds:us-west-2:123456789012:cev:custom-
oracle-ee/19.my_cev1/12a34b5c-67d8-90e1-2f34-gh56ijk78lm9"
            "ValidUpgradeTarget": [
                {
                    "Engine": "custom-oracle-ee",
                    "EngineVersion": "19.my_cev2",
                    "Description": "19.my_cev2 description",
                    "AutoUpgrade": false,
                    "IsMajorVersionUpgrade": false
                },
                {
                    "Engine": "custom-oracle-ee",
```

```
                "EngineVersion": "19.my_cev3",
                "Description": "19.my_cev3 description",
                "AutoUpgrade": false,
                "IsMajorVersionUpgrade": false
            }
        ]
        ...
```

Aggiornamento di un'istanza database RDS Custom per Oracle

Per aggiornare l'istanza database RDS Custom per Oracle, è necessario modificarla per utilizzare un nuovo CEV. Questo CEV può contenere nuovi file binari di database o una nuova AMI. Per aggiornare il database e il sistema operativo, è necessario eseguire due aggiornamenti separati.

> **Note**
>
> Se si aggiorna il database, RDS Custom aggiorna automaticamente le repliche di lettura dopo aver aggiornato l'istanza database primaria. Se si aggiorna il sistema operativo, è necessario aggiornare le repliche manualmente.

Prima di iniziare, consulta Requisiti per gli aggiornamenti di RDS Custom per Oracle e Considerazioni per gli aggiornamenti del database RDS Custom for Oracle.

Console

Per aggiornare un'istanza database RDS Custom per Oracle

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione scegli Database e seleziona l'istanza database RDS Custom per Oracle da aggiornare.
3. Scegliere Modify (Modifica). Viene visualizzata la pagina Modify DB Instance (Modifica istanza database).
4. Per Versione motore database scegli un CEV diverso. Esegui questa operazione:

    - Se stai applicando una patch al database, assicurati che il CEV specifichi file binari del database diversi da quelli utilizzati dall'istanza database e non specifichi un'AMI diversa dall'AMI attualmente utilizzata dall'istanza database.

- Se stai applicando una patch al sistema operativo, assicurati che il CEV specifichi un'AMI diversa dall'AMI attualmente utilizzata dall'istanza database e non specifichi binari del database diversi.

> **Warning**
>
> Quando applichi una patch al sistema operativo, i dati del volume root e qualsiasi personalizzazione esistente del sistema operativo andranno persi.

5. Scegliere Continue (Continua) per controllare il riepilogo delle modifiche.

   Per applicare immediatamente le modifiche, scegliere Apply immediately (Applica immediatamente).

6. Se le modifiche sono corrette, scegliere Modify DB instance (Modifica istanza database). Oppure scegliere Back (Indietro) per cambiare le modifiche o Cancel (Annulla) per annullare le modifiche.

AWS CLI

I seguenti esempi illustrano i possibili scenari di aggiornamento. Gli esempi presuppongono che sia stata creata un'istanza database RDS Custom per Oracle con le seguenti caratteristiche:

- Istanza database denominata `my-custom-instance`
- CEV denominato `19.my_cev1`
- Oracle Database 19c che utilizza l'architettura non CDB
- Oracle Linux 7.9 con AMI `ami-1234`

L'ultima AMI fornita dal servizio è `ami-2345`. È possibile trovare le AMI eseguendo il comando CLI `describe-db-engine-versions`.

Argomenti

- Aggiornamento del sistema operativo
- Aggiornamento del database

Aggiornamento del sistema operativo

In questo esempio, si esegue l'aggiornamento di `ami-1234` a `ami-2345`, che è l'AMI più recente fornita dal servizio. Poiché si tratta di un aggiornamento del sistema operativo, i file binari del database per `ami-1234` e `ami-2345` devono essere uguali. Crea un nuovo CEV denominato `19.my_cev2` basato su `19.my_cev1`.

Example

Per LinuxmacOS, oUnix:

```
aws rds create-custom-db-engine-version \
    --engine custom-oracle-ee \
    --engine-version 19.my_cev2 \
    --description "Non-CDB CEV based on ami-2345" \
    --kms-key-id key-name \
    --source-custom-db-engine-version-identifer arn:aws:rds:us-
west-2:123456789012:cev:custom-oracle-ee/19.my_cev1/12345678-ab12-1234-cde1-
abcde123456789 \
    --image-id ami-2345
```

Per Windows:

```
aws rds create-custom-db-engine-version ^
    --engine custom-oracle-ee ^
    --engine-version 19.my_cev2 ^
    --description "Non-CDB CEV based on ami-2345" ^
    --kms-key-id key-name ^
    --source-custom-db-engine-version-identifer arn:aws:rds:us-
west-2:123456789012:cev:custom-oracle-ee/19.my_cev1/12345678-ab12-1234-cde1-
abcde123456789 ^
    --image-id ami-2345
```

Per aggiornare un'istanza DB personalizzata RDS, utilizzate il modify-db-instance AWS CLI comando con i seguenti parametri:

- `--db-instance-identifier`: specifica l'istanza database RDS Custom per Oracle da aggiornare.
- `--engine-version`: specifica il CEV con la nuova AMI.
- `--no-apply-immediately` | `--apply-immediately`: specifica se eseguire immediatamente l'aggiornamento o attendere fino alla finestra di manutenzione programmata.

Il seguente esempio mostra l'aggiornamento di `my-custom-instance` alla versione `19.my_cev2`. Viene aggiornato solo il sistema operativo.

Example

Per LinuxmacOS, oUnix:

```
aws rds modify-db-instance \
    --db-instance-identifier my-custom-instance \
    --engine-version 19.my_cev2 \
    --apply-immediately
```

Per Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier my-custom-instance ^
    --engine-version 19.my_cev2 ^
    --apply-immediately
```

Aggiornamento del database

In questo esempio si applica la patch Oracle p35042068 all'istanza database RDS per Oracle. Poiché hai aggiornato il sistema operativo in Aggiornamento del sistema operativo, l'istanza database attualmente utilizza `19.my_cev2`, che si basa su `ami-2345`. Crea un nuovo CEV denominato `19.my_cev3` che utilizza `ami-2345`, ma specifica un nuovo manifesto JSON nella variabile di ambiente `$MANIFEST`. Pertanto, solo i file binari del database sono diversi nel nuovo CEV e nel CEV attualmente utilizzato dall'istanza.

Example

Per LinuxmacOS, oUnix:

```
aws rds create-custom-db-engine-version \
    --engine custom-oracle-ee \
    --engine-version 19.my_cev3 \
    --description "Non-CDB CEV with p35042068 based on ami-2345" \
    --kms-key-id key-name \
    --image-id ami-2345 \
    --manifest $MANIFEST
```

Per Windows:

```
aws rds create-custom-db-engine-version ^
    --engine custom-oracle-ee ^
    --engine-version 19.my_cev3 ^
    --description "Non-CDB CEV with p35042068 based on ami-2345" ^
    --kms-key-id key-name ^
    --image-id ami-2345 ^
    --manifest $MANIFEST
```

Il seguente esempio mostra l'aggiornamento di `my-custom-instance` alla versione del motore `19.my_cev3`. Viene aggiornato solo il database.

Example

Per LinuxmacOS, oUnix:

```
aws rds modify-db-instance \
    --db-instance-identifier my-custom-instance \
    --engine-version 19.my_cev3 \
    --apply-immediately
```

Per Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier my-custom-instance ^
    --engine-version 19.my_cev3 ^
    --apply-immediately
```

Visualizzazione di aggiornamenti del database in sospeso per le istanze database RDS Custom

Puoi visualizzare gli aggiornamenti del database in sospeso per le tue istanze database personalizzate di Amazon RDS utilizzando il comando or. describe-db-instancesdescribe-pending-maintenance-actions AWS CLI

Tuttavia, questo approccio non funziona se si utilizza l'opzione `--apply-immediately` o se l'aggiornamento è in corso.

Il comando seguente `describe-db-instances` mostra gli aggiornamenti del database in sospeso per `my-custom-instance`.

```
aws rds describe-db-instances --db-instance-identifier my-custom-instance
```

L'output è simile a quello riportato di seguito.

```
{
    "DBInstances": [
        {
           "DBInstanceIdentifier": "my-custom-instance",
            "EngineVersion": "19.my_cev1",
            ...
            "PendingModifiedValues": {
                "EngineVersion": "19.my_cev3"
            ...
            }
        }
    ]
}
```

Risoluzione dei problemi di aggiornamento per un'istanza database RDS Custom per Oracle

Se l'aggiornamento di un'istanza database RDS Custom non riesce, viene generato un evento RDS e lo stato dell'istanza database diventa `upgrade-failed`.

Puoi visualizzare questo stato utilizzando il describe-db-instances AWS CLI comando, come mostrato nell'esempio seguente.

```
aws rds describe-db-instances --db-instance-identifier my-custom-instance
```

L'output è simile a quello riportato di seguito.

```
{
    "DBInstances": [
        {
           "DBInstanceIdentifier": "my-custom-instance",
            "EngineVersion": "19.my_cev1",
            ...
            "PendingModifiedValues": {
                "EngineVersion": "19.my_cev3"
            ...
            }
```

```
            "DBInstanceStatus": "upgrade-failed"
        }
    ]
}
```

Dopo un errore di aggiornamento, tutte le azioni del database vengono bloccate tranne che per la modifica dell'istanza database per eseguire le seguenti attività:

- Riprovare lo stesso aggiornamento
- Sospensione e ripristino dell'automazione RDS Custom
- oint-in-time Ripristino P (PITR)
- Eliminazione di un'istanza database

> **Note**
>
> Se l'automazione è stata sospesa per l'istanza database RDS Custom, non è possibile riprovare l'aggiornamento fino a quando non l'automazione non viene ripresa.
> Le stesse azioni si applicano a un errore di aggiornamento per una replica di lettura gestita da RDS come per la primaria.

Per ulteriori informazioni, consultare Risoluzione dei problemi di aggiornamento per RDS Custom per Oracle.

Risoluzione dei problemi relativi ai database di Amazon RDS Custom per Oracle

Il modello di responsabilità condivisa di RDS Custom fornisce l'accesso a livello di shell al sistema operativo e l'accesso come amministratore al database. RDS Custom esegue risorse nel proprio account, a differenza di Amazon RDS, che esegue le risorse in un account di sistema. Con un maggiore accesso si ottiene una maggiore responsabilità. Nelle sezioni seguenti sono descritte le procedure di risoluzione dei problemi relativi alle istanze database Amazon RDS Custom.

> **Note**
>
> Questa sezione spiega come risolvere i problemi relativi a RDS Custom per Oracle. Per informazioni sulla risoluzione dei problemi relativi a RDS Custom per SQL Server, consulta Risoluzione dei problemi relativi ai database di Amazon RDS Custom per SQL Server.

Argomenti

Visualizzazione di eventi RDS Custom

La procedura per visualizzare gli eventi è la stessa per le istanze database Amazon RDS e RDS Custom. Per ulteriori informazioni, consulta Visualizzazione di eventi Amazon RDS.

Per visualizzare la notifica degli eventi RDS Custom utilizzando il AWS CLI, utilizzare il `describe-events` comando. RDS Custom presenta diversi nuovi eventi. Le categorie di eventi sono le stesse di Amazon RDS. Per l'elenco di eventi, consultare Categorie di eventi Amazon RDS e messaggi di evento.

Nell'esempio seguente vengono recuperati i dettagli per gli eventi verificati per l'istanza database RDS Custom specificata.

```
aws rds describe-events \
    --source-identifier my-custom-instance \
    --source-type db-instance
```

Iscrizione agli eventi RDS Custom

La procedura per visualizzare gli eventi è la stessa per le istanze database Amazon RDS e RDS Custom. Per ulteriori informazioni, consulta Sottoscrizione alle notifiche eventi di Amazon RDS.

Per abbonarsi alle notifiche degli eventi RDS Custom utilizzando la CLI, utilizza il comando `create-event-subscription`. Includi i parametri obbligatori seguenti:

- `--subscription-name`
- `--sns-topic-arn`

Nell'esempio seguente viene creata una sottoscrizione per gli eventi di backup e ripristino per un'istanza database RDS Custom nell'account AWS attuale. Le notifiche sono inviate a un argomento Amazon Simple Notification Service (Amazon SNS) specificato da `--sns-topic-arn`.

```
aws rds create-event-subscription \
    --subscription-name my-instance-events \
    --source-type db-instance \
    --event-categories '["backup","recovery"]' \
    --sns-topic-arn arn:aws:sns:us-east-1:123456789012:interesting-events
```

Risoluzione dei problemi relativi alla creazione di versioni personalizzate del motore per RDS Custom per Oracle

Se la creazione di CEV non riesce, RDS Custom emette `RDS-EVENT-0198` con il messaggio `Creation failed for custom engine version` *`major-engine-version.cev_name`* e include i dettagli sull'errore. Ad esempio, l'evento stampa i file mancanti.

La creazione di CEV potrebbe non riuscire a causa dei seguenti problemi:

- Il bucket Amazon S3 contenente i file di installazione non si trova nella stessa AWS regione del CEV.
- Quando richiedi la creazione di CEV Regione AWS per la prima volta, RDS Custom crea un bucket S3 per archiviare risorse RDS Custom (come artefatti CEV, log e log delle transazioni). AWS CloudTrail

La creazione di CEV non riesce se RDS Custom non è in grado di creare il bucket S3. O il chiamante non dispone delle autorizzazioni S3 come descritto in Passaggio 5: concedi le autorizzazioni necessarie al tuo utente o ruolo IAM o il numero di bucket S3 ha raggiunto il limite.

- Il chiamante non dispone delle autorizzazioni per ottenere i file dal bucket S3 che contiene i file multimediali di installazione. Queste autorizzazioni sono descritte in Fase 7: aggiunta delle autorizzazioni IAM necessarie.
- La tua policy IAM ha una condizione `aws:SourceIp`. Assicurati di seguire i consigli in AWS nega l'accesso ad AWS in base all'IP di origine nella Guida per l'utente di AWS Identity and Access Management . Assicurati inoltre che il chiamante disponga delle autorizzazioni S3 descritte in Passaggio 5: concedi le autorizzazioni necessarie al tuo utente o ruolo IAM.
- I file multimediali di installazione elencati nel manifest CEV non si trovano nel bucket S3.
- Le checksum SHA-256 dei file di installazione sono sconosciute a RDS Custom.

  Confermare che i checksum SHA-256 dei file forniti corrispondano al checksum SHA-256 sul sito Web Oracle. Se i checksum corrispondono, contattare AWS Supporto e fornire il nome CEV, il nome del file e il checksum non riusciti.
- La versione di OPatch non è compatibile con i file di patch. È possibile che venga visualizzato il seguente messaggio: `OPatch is lower than minimum required version. Check that the version meets the requirements for all patches, and try again`. Per applicare una patch Oracle, è necessario utilizzare una versione compatibile dell'utilità OPatch. La versione richiesta dell'utilità Opatch è disponibile nel file readme della patch. Scarica l'utilità OPatch più recente da My Oracle Support e prova a creare nuovamente il tuo CEV.
- Le patch specificate nel manifesto CEV sono nell'ordine sbagliato.

È possibile visualizzare gli eventi RDS sulla console RDS (nel pannello di navigazione, scegliere Eventi) o utilizzando il comando. `describe-events` AWS CLI La durata predefinita è di 60 minuti. Se non vengono restituiti eventi, specificare una durata più lunga, come indicato nell'esempio seguente.

```
aws rds describe-events --duration 360
```

Attualmente, il MediaImport servizio che importa file da Amazon S3 per creare CEV non è integrato con. AWS CloudTrail Pertanto, se attivi la registrazione dei dati per Amazon RDS CloudTrail, le chiamate al MediaImport servizio come l'`CreateCustomDbEngineVersion`evento non vengono registrate.

Tuttavia, si potrebbero vedere chiamate dal gateway API che accede al bucket Amazon S3. Queste chiamate provengono dal MediaImport servizio dell'evento. `CreateCustomDbEngineVersion`

Correzione delle configurazioni non supportate in RDS Custom per Oracle

In base al modello di responsabilità condivisa, è tua responsabilità risolvere i problemi di configurazione che comportano il passaggio dell'istanza database RDS Custom per Oracle allo stato `unsupported-configuration`. Se il problema riguarda l' AWS infrastruttura, puoi utilizzare la console o AWS CLI risolverlo. Se il problema riguarda il sistema operativo o la configurazione del database, è possibile accedere all'host per risolverlo.

> **Note**
>
> Questa sezione spiega come correggere le configurazioni non supportate in RDS Custom per Oracle. Per ulteriori informazioni su RDS Custom per SQL Server, consulta Correzione delle configurazioni non supportate in RDS Custom per SQL Server.

Nella tabella seguente puoi trovare le descrizioni delle notifiche e degli eventi inviati dal perimetro di supporto e come risolverli. Queste notifiche e il perimetro di supporto sono soggetti a modifiche. Per informazioni sul perimetro del supporto, consulta Perimetro di supporto RDS Custom. Per le descrizioni degli eventi, consulta Categorie di eventi Amazon RDS e messaggi di evento.

ID evento	Configurazione	Messaggio dell'evento RDS	Azione
SP-00000	Configurazione manuale non supportata	*Lo stato dell'istanza RDS Custom DB è impostato su [Configurazione non supportata] per: motivo.*	Per risolvere questo problema, crea un AWS Support caso.
AWS risorse (infrastruttura)			

ID evento	Configurazione	Messaggio dell'evento RDS	Azione
SP-O1001	Volumi Amazon Elastic Block Store (Amazon EBS)	*I seguenti volumi EBS sono stati aggiunti all'istanza EC2 ec2_id: volume_id.* Per risolvere il problema, scollega i volumi specificati dall'istanza.	RDS Custom crea due tipi di volume EBS, oltre al volume root creato da Amazon Machine Image (AMI), e li associa all'istanza EC2: • Il volume binario in cui si trovano i file binari del software di database • I volumi di dati in cui si trovano i file del database Quando crei l'istanza DB, le configurazioni di storage che specifichi configurano i volumi di dati. Il perimetro di supporto monitora quanto segue: • I volumi EBS iniziali creati con l'istanza DB sono ancora associati all'istanza. • I volumi EBS iniziali hanno ancora le stesse configurazioni impostate inizialmente: tipo di archiviazione, dimensioni, IOPS con provisioning e throughput di archiviazione. • Nessun volume EBS aggiuntivo è collegato all'istanza database. Utilizza il seguente comando CLI per confrontare il tipo di volume dei dettagli del volume EBS e i dettagli dell'istanza DB RDS Custom for Oracle: `aws rds describe-db-instances \ --db-instance-identifier db-instance-name | grep StorageType`

ID evento	Configurazione	Messaggio dell'evento RDS	Azione
SP-O1002	Volumi Amazon Elastic Block Store (Amazon EBS)	*Il volume EBS volume_id è stato scollegato dall'istanza EC2 [ec2_id].* Non puoi scollegare il volume originale da questa istanza. Per risolvere il problema, ricollega *volume_id a ec2_id.*	RDS Custom crea due tipi di volume EBS, oltre al volume root creato da Amazon Machine Image (AMI), e li associa all'istanza EC2: • Il volume binario in cui si trovano i file binari del software di database • I volumi di dati in cui si trovano i file del database Quando crei l'istanza DB, le configurazioni di storage che specifichi configurano i volumi di dati. Il perimetro di supporto monitora quanto segue: • I volumi EBS iniziali creati con l'istanza DB sono ancora associati all'istanza. • I volumi EBS iniziali hanno ancora le stesse configurazioni impostate inizialmente: tipo di archiviazione, dimensioni, IOPS con provisioning e throughput di archiviazione. • Nessun volume EBS aggiuntivo è collegato all'istanza database. Utilizza il seguente comando CLI per confrontare il tipo di volume dei dettagli del volume EBS e i dettagli dell'istanza DB RDS Custom for Oracle: `aws rds describe-db-instances \ --db-instance-identifier db-instance-name | grep StorageType`

ID evento	Configurazione	Messaggio dell'evento RDS	Azione
SP-O1003	Volumi Amazon Elastic Block Store (Amazon EBS)	*Il volume EBS volume_id originale collegato all'istanza EC2 ec2_id è stato modificato come segue: dimensione da [X] a [Y], tipo da [N] a [M] o IOPS [J] a [K].* Per risolvere il problema, annulla la modifica.	RDS Custom crea due tipi di volume EBS, oltre al volume root creato da Amazon Machine Image (AMI), e li associa all'istanza EC2: • Il volume binario in cui si trovano i file binari del software di database • I volumi di dati in cui si trovano i file del database Quando crei l'istanza DB, le configurazioni di storage che specifichi configurano i volumi di dati. Il perimetro di supporto monitora quanto segue: • I volumi EBS iniziali creati con l'istanza DB sono ancora associati all'istanza. • I volumi EBS iniziali hanno ancora le stesse configurazioni impostate inizialmente: tipo di archiviazione, dimensioni, IOPS con provisioning e throughput di archiviazione. • Nessun volume EBS aggiuntivo è collegato all'istanza database. Utilizza il seguente comando CLI per confrontare il tipo di volume dei dettagli del volume EBS e i dettagli dell'istanza DB RDS Custom for Oracle: `aws rds describe-db-instances \ --db-instance-identifier db-instance-name | grep StorageType`

ID evento	Configurazione	Messaggio dell'evento RDS	Azione
SP-O1004	Stato dell'istanza Amazon EC2	Il ripristino automatico ha lasciato l'istanza EC2 [*ec2_id*] in uno stato compromesso. Per risolvere il problema, consulta Risoluzione dei problemi di ripristino delle istanze.	Per verificare lo stato di un'istanza DB, usa la console o esegui il AWS CLI comando seguente: `aws rds describe-db-instances \ --db-instance-identifier *db-instance-name* |grep DBInstanceStatus`
SP-O1005	Attributi dell'istanza Amazon EC2	*L'istanza EC2 [ec2_id] è stata modificata come segue: l'attributo [att1] è stato modificato da [val-old] a [val-new], l'attributo [att2] è stato modificato da [val-old] a [val-new].* Per risolvere il problema, ripristina il valore originale.	

ID evento	Configurazione	Messaggio dell'evento RDS	Azione
SP-O1006	Stato dell'istanza Amazon EC2	L'istanza EC2 [*ec2_id*] è stata terminata o non può essere trovata. Per risolvere il problema, elimina l'istanza DB personalizzata RDS.	Il perimetro di supporto monitora le notifiche di modifica dello stato dell'istanza EC2. L'istanza EC2 deve essere sempre in esecuzione. Per eliminare l'istanza DB 1. Per verificare lo stato di un'istanza DB, usa la console o esegui il seguente AWS CLI comando: `aws rds describe-db-instances \ --db-instance-identifier db-instance-name |grep DBInstanceStatus` 2. Elimina la tua istanza DB RDS Custom for Oracle.
SP-O1007	Stato dell'istanza Amazon EC2	*L'istanza EC2 [ec2_id] è stata interrotta.* Per risolvere il problema, avvia l'istanza.	Il perimetro di supporto monitora le notifiche di modifica dello stato dell'istanza EC2. L'istanza EC2 deve essere sempre in esecuzione. Per riavviare l'istanza DB 1. Per verificare lo stato di un'istanza DB, usa la console o esegui il seguente AWS CLI comando: `aws rds describe-db-instances \ --db-instance-identifier db-instance-name |grep DBInstanceStatus` 2. Avvia la tua istanza DB. 3. Rimontare i volumi binari e di dati.
Sistema operativo			

ID evento	Configurazione	Messaggio dell'evento RDS	Azione
SP-O2001	Stato dell'agente RDS Custom	*L'agente RDS Custom non è in esecuzione sull'istanza EC2 [ec2_id]. Assicurati che l'agente sia in esecuzione su [ec2_id].*	In RDS Custom per Oracle, l'istanza database esce dal perimetro di supporto se l'agente RDS Custom si arresta. L'agente pubblica la `IamAlive` metrica su Amazon CloudWatch ogni 30 secondi. Viene attivato un allarme se il parametro non è stato pubblicato per 30 secondi. Il perimetro di supporto controlla inoltre lo stato del processo dell'agente RDS Custom sull'host ogni 30 minuti. Per riavviare l'agente RDS Custom 1. Accedi all'host e verifica che l'agente RDS Custom sia in esecuzione. 2. Esegui il comando seguente per trovare lo stato dell'agente. `service rdscustomagent status` 3. Utilizzate il seguente comando per avviare l'agente. `service rdscustomagent start` Quando l'agente RDS Custom è nuovamente in esecuzione, la `IamAlive` metrica viene pubblicata su Amazon CloudWatch e l'allarme passa allo stato. `OK` Questo switch notifica al perimetro di supporto che l'agente è in esecuzione.

ID evento	Configurazione	Messaggio dell'evento RDS	Azione
SP-02002	AWS Systems Manager stato dell'agente (agente SSM)	L'agente Systems Manager sull'istanza EC2 [*ec2_id*] non è raggiungibile. Assicurati di aver configurato correttamente la rete, l'agente e le autorizzazioni IAM.	L'agente SSM deve essere sempre in esecuzione. L'agente RDS Custom è responsabile di assicurarsi che Systems Manager agent sia in esecuzione. Se SSM Agent è stato terminato e poi riavviato, l'agente RDS Custom pubblica una metrica su. CloudWatch L'agente RDS Custom ha un allarme sul parametro impostato per attivarsi quando è stato effettuato un riavvio in ciascuno dei tre minuti precedenti. Il perimetro di supporto monitora inoltre lo stato del processo di SSM Agent sull'host ogni 30 minuti. Per ulteriori informazioni, consulta Risoluzione dei problemi relativi all'SSM Agent.
SP-02003	AWS Systems Manager stato dell'agente (agente SSM)	L'agente Systems Manager sull'istanza EC2 [*ec2_id*] si è bloccato più volte. Per ulteriori informazioni, consulta la documentazione sulla risoluzione dei problemi di SSM Agent.	Per ulteriori informazioni, consulta Risoluzione dei problemi relativi all'SSM Agent.

ID evento	Configurazione	Messaggio dell'evento RDS	Azione
SP-O2004	Fuso orario del sistema operativo	Il fuso orario sull'istanza EC2 [*ec2_id*] è stato modificato. Per risolvere questo problema, ripristina il fuso orario all'impostazione precedente di []. *previous-time-zone* Quindi utilizza un gruppo di opzioni RDS per modificare il fuso orario.	L'automazione RDS ha rilevato che il fuso orario sull'host è stato modificato senza l'uso di un gruppo di opzioni. Questa modifica a livello di host può causare errori di automazione RDS, pertanto l'istanza EC2 viene collocata nello stato. `unsupported-configuration` Per correggere l'impostazione del fuso orario 1. Accedi al tuo host EC2 e controlla il fuso orario del sistema operativo come segue: `timedatectl` 2. Sospendi l'automazione RDS Custom. Per ulteriori informazioni, consulta Sospensione e ripresa dell'istanza database RDS Custom. 3. Arresta l'istanza DB. 4. Ripristina la modifica del fuso orario nel sistema operativo. 5. Avviare l'istanza database. 6. Riprendere l'automazione RDS Custom. L'istanza database diventa disponibile entro 30 minuti. Per evitare di uscire dal perimetro in futuro, modifica il fuso orario tramite un gruppo di opzioni. Per ulteriori informazioni, consulta Fuso orario Oracle.

ID evento	Configurazione	Messaggio dell'evento RDS	Azione
SPO-2	Configurazioni sudo	*Le configurazioni sudo sull'istanza EC2 [ec2_id] non dispongono delle autorizzazioni necessarie.* Per risolvere questo problema, ripristina le modifiche recenti alle configurazioni sudo.	Il perimetro di supporto monitora che alcuni utenti del sistema operativo possano eseguire determinati comandi sulla scatola. Monitora le configurazioni sudo rispetto allo stato supportato. Quando le configurazioni sudo non sono supportate, RDS Custom tenta di sovrascriverle riportandole al precedente stato supportato. In caso di esito positivo, viene inviata la seguente notifica: RDS Custom ha sovrascritto con successo la configurazione. Per esaminare le modifiche alle configurazioni sudo 1. Accedi al tuo host. 2. Esegui il comando seguente. `visudo -c -f /etc/sudoers.d/ individual_sudo_files` 3. Modifica le sudo configurazioni secondo necessità. Dopo che il perimetro di supporto ha stabilito che le sudo configurazioni sono supportate, l'istanza DB RDS Custom for Oracle diventa disponibile entro 30 minuti.

ID evento	Configurazione	Messaggio dell'evento RDS	Azione
SPO-2	Accessibilità del bucket S3	*L'automazione RDS Custom non può scaricare file dal bucket S3 sull'istanza EC2 [ec2_id].* Controlla la configurazione di rete e assicurati che l'istanza consenta le connessioni da e verso S3.	
Database			

ID evento	Configurazione	Messaggio dell'evento RDS	Azione
SP-O3001	Target di ritardo dell'archivio del database	*Il parametro ARCHIVE_LAG_TARGET sull'istanza EC2 [ec2_id] non rientra nell'intervallo consigliato value_range.* Per risolvere il problema, imposta il parametro su un valore all'interno di value_range.	Il perimetro di supporto monitora il parametro del `ARCHIVE_LAG_TARGET` database per verificare che l'ultimo periodo di ripristino dell'istanza DB rientri nei limiti ragionevoli. Per modificare l'obiettivo di ritardo per i redo log archiviati 1. Accedi al tuo host EC2 2. Connect alla tua istanza DB RDS Custom for Oracle 3. Modificate il `ARCHIVE_LAG_TARGET` parametro impostando un valore compreso tra 60 e 7200. Ad esempio, utilizzate la seguente istruzione SQL. `ALTER SYSTEM SET ARCHIVE_LAG_TARGET=300 SCOPE=BOTH;` L'istanza database diventa disponibile entro 30 minuti.

ID evento	Configurazione	Messaggio dell'evento RDS	Azione
SP-O3002	Ruolo di Oracle Data Guard	*Il ruolo del database [role_name] non è supportato per Oracle Data Guard sull'istanza EC2 [ec2_id].* Per risolvere il problema, imposta il parametro DATABASE_ROLE su PRIMARY o PHYSICAL STANDBY.	Il perimetro di supporto monitora il ruolo corrente del database ogni 15 secondi e invia una CloudWatch notifica se il ruolo del database è cambiato. Il parametro `DATABASE_ROLE` Oracle Data Guard deve essere o `PRIMARY` o `PHYSICAL STANDBY`. Per ripristinare il ruolo del database Oracle Data Guard su un valore supportato 1. Verifica il ruolo di Oracle Data Guard eseguendo la seguente istruzione: `SELECT DATABASE_ROLE FROM V$DATABASE;` 2. Se l'istanza DB è autonoma, utilizza una delle seguenti istruzioni per riportarla al `PRIMARY` ruolo: `ALTER DATABASE COMMIT TO SWITCHOVER PRIMARY;` `ALTER DATABASE ACTIVATE STANDBY DATABASE;` Se l'istanza DB è una replica, usa la seguente istruzione per riportarla al `PHYSICAL STANDBY` ruolo: `ALTER DATABASE CONVERT TO PHYSICAL STANDBY;` Dopo che il perimetro di supporto determina che il ruolo del database è supportato, l'istanza database RDS Custom per Oracle diventa disponibile entro 15 secondi.

ID evento	Configurazione	Messaggio dell'evento RDS	Azione
SP-O3003	Integrità del database	Il processo SMON del database Oracle è in uno stato zombie. Per risolvere il problema, ripristina manualmente il database sull'istanza EC2 [*ec2_id*], apri il database ed esegui immediatamente il backup. Per ulteriore assistenza, contatta. AWS Support	Il perimetro di supporto monitora lo stato dell'istanza database. Monitora inoltre il numero di riavvii avvenuti durante l'ora e il giorno precedenti. Viene notificato quando l'istanza si trova in uno stato in cui esiste ancora, ma non è possibile interagire con essa. Per fare in modo che il perimetro di supporto valuti lo stato dell'istanza 1. Accedi al tuo host e determina lo stato del database. `ps -eo pid,state,command | grep smon` 2. Se necessario, riavvia l'istanza DB. Se il riavvio fallisce, procedi al passaggio successivo. 3. Se necessario, riavvia l'host EC2. Dopo il riavvio dell'istanza DB, l'agente RDS Custom rileva che l'istanza DB non è più in uno stato di non risposta. Quindi invia una notifica al perimetro di supporto affinché rivaluti lo stato dell'istanza database.

ID evento	Configurazione	Messaggio dell'evento RDS	Azione
SP-O3004	Modalità di log del database	*La modalità di registro del database sull'istanza EC2 [ec2_id] è stata modificata in [value_b]. Per risolvere il problema, imposta la modalità di registro su [value_a].*	Per modificare la modalità di registro dell'istanza DB su **ARCHIVELOG** 1. Accedi al tuo host EC2. 2. Connect al database ed esegui la seguente istruzione: `SELECT LOG_MODE FROM V$DATABASE;` Oppure è possibile eseguire il seguente comando in SQL*Plus: `ARCHIVE LOG LIST` 3. Esegui il seguente comando SQL*Plus per avviare un arresto coerente. `SHUTDOWN IMMEDIATE` L'agente RDS Custom riavvia automaticamente l'istanza DB e imposta la modalità di registro su. `ARCHIVELOG` L'istanza database diventa disponibile entro 30 minuti.

ID evento	Configurazione	Messaggio dell'evento RDS	Azione
SP-O3005	Percorso principale di Oracle	*La home Oracle sull'istanza EC2 [ec2_id] è stata modificata in new_path. Per risolvere il problema, ripristina l'impostazione su old_path.*	

ID evento	Configurazione	Messaggio dell'evento RDS	Azione
SP-O3006	Nome univoco del database	*Il nome univoco del database sull'istanza EC2 [ec2_id] è stato modificato in new_value . Per risolvere il problema, ripristina il nome in old_value.*	Per modificare il nome univoco del database per l'istanza DB 1. Accedi al tuo host EC2. 2. Connect al database ed esegui la seguente istruzione: `SELECT DB_UNIQUE_NAME FROM V$DATABASE;` 3. Specificare il nome univoco del database originale utilizzando il comando`ALTER SYSTEM SET DB_UNIQUE_NAME .` 4. Esegui la seguente istruzione SQL per avviare un arresto coerente. `SHUTDOWN IMMEDIATE;` L'agente RDS Custom riavvia automaticamente l'istanza DB e imposta la modalità di registro su. `ARCHIVELOG` L'istanza database diventa disponibile entro 30 minuti.

Risoluzione dei problemi di aggiornamento per RDS Custom per Oracle

L'aggiornamento di un'istanza RDS Custom per Oracle potrebbe non riuscire. Di seguito sono riportate alcune tecniche che è possibile utilizzare durante gli aggiornamenti delle istanze database RDS Custom per Oracle:

- Analizza i file di log dell'output degli aggiornamenti nella directory `/tmp` dell'istanza database. I nomi dei log dipendono dalla versione del motore DB. Ad esempio, potrebbero venire visualizzati i log contenenti le stringhe `catupgrd` o `catup`.
- Analizza il file `alert.log` disponibile nella directory `/rdsdbdata/log/trace`.

- Eseguire il seguente comando `grep` nella directory `root` per monitorare il processo di aggiornamento del sistema operativo. Questo comando mostra dove vengono scritti i file di log e determina lo stato del processo di aggiornamento.

```
ps -aux | grep upg
```

Di seguito viene mostrato l'output di esempio.

```
root     18884  0.0  0.0 235428  8172 ?        S<   17:03   0:00 /usr/bin/
sudo -u rdsdb /rdsdbbin/scripts/oracle-control ORCL op_apply_upgrade_sh RDS-
UPGRADE/2.upgrade.sh
rdsdb    18886  0.0  0.0 153968 12164 ?        S<   17:03   0:00 /usr/bin/perl -T -
w /rdsdbbin/scripts/oracle-control ORCL op_apply_upgrade_sh RDS-UPGRADE/2.upgrade.sh
rdsdb    18887  0.0  0.0 113196  3032 ?        S<   17:03   0:00 /bin/sh /rdsdbbin/
oracle/rdbms/admin/RDS-UPGRADE/2.upgrade.sh
rdsdb    18900  0.0  0.0 113196  1812 ?        S<   17:03   0:00 /bin/sh /rdsdbbin/
oracle/rdbms/admin/RDS-UPGRADE/2.upgrade.sh
rdsdb    18901  0.1  0.0 167652 20620 ?        S<   17:03   0:07 /rdsdbbin/oracle/
perl/bin/perl catctl.pl -n 4 -d /rdsdbbin/oracle/rdbms/admin -l /tmp catupgrd.sql
root     29944  0.0  0.0 112724  2316 pts/0    S+   18:43   0:00 grep --color=auto
 upg
```

- Eseguire la seguente query SQL per verificare lo stato corrente dei componenti per trovare la versione del database e le opzioni installate nell'istanza database.

```
SET LINESIZE 180
COLUMN COMP_ID FORMAT A15
COLUMN COMP_NAME FORMAT A40 TRUNC
COLUMN STATUS FORMAT A15 TRUNC
SELECT COMP_ID, COMP_NAME, VERSION, STATUS FROM DBA_REGISTRY ORDER BY 1;
```

L'output è simile a quello riportato di seguito.

```
COMP_NAME                                STATUS               PROCEDURE
---------------------------------------- --------------------
 ------------------------------------------------
Oracle Database Catalog Views            VALID
 DBMS_REGISTRY_SYS.VALIDATE_CATALOG
Oracle Database Packages and Types       VALID
 DBMS_REGISTRY_SYS.VALIDATE_CATPROC
Oracle Text                              VALID                VALIDATE_CONTEXT
```

```
Oracle XML Database                    VALID                DBMS_REGXDB.VALIDATEXDB

4 rows selected.
```

- Eseguire la seguente query SQL per verificare la presenza di oggetti non validi che potrebbero interferire con il processo di aggiornamento.

```
SET PAGES 1000 LINES 2000
COL OBJECT FOR A40
SELECT SUBSTR(OWNER,1,12) OWNER,
       SUBSTR(OBJECT_NAME,1,30) OBJECT,
       SUBSTR(OBJECT_TYPE,1,30) TYPE, STATUS,
       CREATED
FROM   DBA_OBJECTS
WHERE  STATUS <>'VALID'
AND    OWNER IN ('SYS','SYSTEM','RDSADMIN','XDB');
```

Risoluzione dei problemi di promozione delle repliche per RDS Custom per Oracle

È possibile promuovere le repliche Oracle gestite in RDS Custom for Oracle utilizzando la console, il `promote-read-replica` AWS CLI comando o l'API. `PromoteReadReplica` Se si elimina l'istanza database primaria e tutte le repliche sono integre, RDS Custom per Oracle promuove automaticamente le repliche gestite in istanze autonome. Se una replica ha sospeso l'automazione o si trova al di fuori del perimetro di supporto, è necessario correggerla prima che RDS Custom possa promuoverla automaticamente. Per ulteriori informazioni, consulta Limitazioni della promozione delle repliche per RDS Custom per Oracle.

Il flusso di lavoro di promozione delle repliche potrebbe bloccarsi nella seguente situazione:

- Lo stato dell'istanza database primaria è `STORAGE_FULL`.
- Il database primario non è in grado di archiviare tutti i redo log online.
- Si è verificata una mancata sincronizzazione tra i file dei registri di ripristino archiviati nella replica Oracle e il database primario.

Per rispondere al flusso di lavoro bloccato

1. Sincronizza il registro di ripristino con l'istanza database di replica Oracle.

2. Forza la promozione della replica di lettura in base all'ultimo registro di ripristino applicato. Esegui i seguenti comandi in SQL*Plus:

```
ALTER DATABASE ACTIVATE STANDBY DATABASE;
SHUTDOWN IMMEDIATE
STARTUP
```

3. Contatta AWS Support e chiedi loro di spostare la tua istanza DB allo `available` stato.

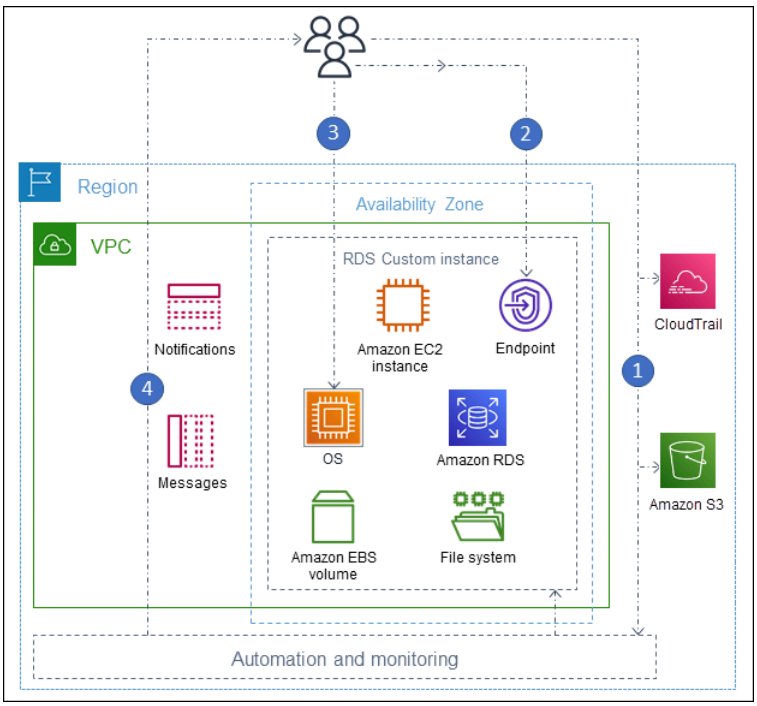
Utilizzo di RDS Custom for SQL Server

Di seguito puoi trovare le istruzioni per creare, gestire e mantenere le tue istanze database RDS Custom for SQL Server.

Argomenti

Flusso di lavoro RDS Custom per SQL Server

Il diagramma seguente mostra il flusso di lavoro tipico per RDS Custom for SQL Server.



I passaggi sono i seguenti:

1. Crea un'istanza database RDS Custom for SQL Server da una versione del motore offerta da RDS Custom.

   Per ulteriori informazioni, consultare Creazione di un'istanza database RDS Custom per SQL Server.
2. Connect l'applicazione all'endpoint dell'istanza RDS Custom DB.

   Per ulteriori informazioni, consulta Connessione alla tua istanza DB personalizzata RDS tramite AWS Systems Manager e Connessione all'istanza database RDS Custom tramite RDP.
3. (Facoltativo) Accedi all'host per personalizzare il software.
4. Monitora le notifiche e i messaggi generati dall'automazione RDS Custom.

Creazione di un'istanza database RDS Custom

Create la vostra istanza RDS Custom DB utilizzando il comando `create-db-instance`. La procedura è simile a quella per la creazione di un'istanza Amazon RDS. Tuttavia, alcuni dei parametri sono diversi. Per ulteriori informazioni, consultare Creazione e connessione a un'istanza database per Amazon RDS Custom per SQL Server.

Connessioni database

Come un'istanza Amazon RDS DB, l'istanza DB RDS Custom per SQL Server risiede in un VPC. L'applicazione si connette all'istanza RDS Custom utilizzando un client come SQL Server Management Suite (SSMS), proprio come in RDS per SQL Server.

Personalizzazione RDS Personalizza

È possibile accedere all'host RDS Custom per installare o personalizzare il software. Per evitare conflitti tra le modifiche e l'automazione personalizzata di RDS, è possibile sospendere l'automazione per un periodo specificato. Durante questo periodo, RDS Custom non esegue il monitoraggio o il ripristino dell'istanza. Al termine del periodo, RDS Custom riprende l'automazione completa. Per ulteriori informazioni, consultare Sospensione e ripristino dell'automazione RDS Custom.

Requisiti e limitazioni per Amazon RDS Custom for SQL Server

Di seguito puoi trovare un riepilogo dei requisiti e delle limitazioni di Amazon RDS Custom for SQL Server per una rapida consultazione. I requisiti e le limitazioni appaiono anche nelle sezioni pertinenti.

Argomenti

Disponibilità di regioni e versioni

Il supporto varia a seconda delle versioni specifiche di ciascun motore di database e a seconda delle Regioni AWS. Per ulteriori informazioni sulla disponibilità di versioni e regioni di Amazon RDS con Amazon RDS Custom per SQL Server, consulta Regioni e motori DB supportati per RDS Custom per SQL Server.

Requisiti generali per RDS Custom per SQL Server

Assicurati di seguire questi requisiti per Amazon RDS Custom per SQL Server:

- Usa le classi di istanza mostrate in Supporto delle classi di istanza database per RDS Custom for SQL Server. Gli unici tipi di storage supportati sono le unità a stato solido (SSD) di tipo gp2, gp3, io1 e io2 Block Express. Il limite massimo di archiviazione è di 16 TiB.
- Assicurati di disporre di una chiave di crittografia simmetrica per creare un'istanza DB personalizzata RDS AWS KMS . Per ulteriori informazioni, consulta Assicurati di disporre di una chiave di crittografia simmetrica AWS KMS.
- Assicurati di creare un ruolo AWS Identity and Access Management (IAM) e un profilo di istanza. Per ulteriori informazioni, consulta Creazione manuale del ruolo IAM e del profilo dell'istanza e Creazione automatica del profilo di istanza utilizzando il AWS Management Console.

- Assicurati di fornire una configurazione di rete che RDS Custom possa utilizzare per accedere ad altre Servizi AWS. Per requisiti specifici, consulta Passaggio 2: configurare la rete, il profilo dell'istanza e la crittografia.
- Il numero combinato di istanze RDS Custom e Amazon RDS DB non può superare il limite di quota. Ad esempio, se la tua quota è di 40 istanze DB, puoi avere 20 istanze RDS Custom per SQL Server DB e 20 istanze Amazon RDS DB.
- RDS Custom crea automaticamente un AWS CloudTrail percorso il cui nome inizia con. `do-not-delete-rds-custom-` Il perimetro di supporto RDS Custom si basa sugli eventi di CloudTrail per determinare se le azioni dell'utente influiscono sull'automazione RDS Custom. RDS Custom genera il trail quando crei la prima istanza database. Per utilizzarne uno già esistente CloudTrail, contatta l' AWS assistenza. Per ulteriori informazioni, consulta AWS CloudTrail.

Supporto delle classi di istanza database per RDS Custom for SQL Server

Verifica se la classe di istanza DB è supportata nella tua regione utilizzando il comando describe-orderable-db-instance-options.

RDS Custom for SQL Server supporta le classi di istanze DB mostrate nella tabella seguente:

SQL Server Edition	supporto RDS Custom
Enterprise Edition	db.r5.large–db.r5.24xlarge db.r5b.xlarge—db.r5b.24xlarge db.m5.large–db.m5.24xlarge db.r6i.xlarge — db.r6i.32xlarge db.m6i.xlarge — db.m6i.32xlarge db.x2iedn.xlarge - db.x2iedn.32xlarge
Standard Edition	db.r5.large–db.r5.24xlarge db.r5b.large–db.r5b.8xlarge

SQL Server Edition	supporto RDS Custom
	db.m5.large–db.m5.24xlarge db.r6i.large — db.r6i.8xlarge db.m6i.large — db.m6i.8xlarge db.x2iedn.xlarge — db.x2iedn.8xlarge
Developer Edition	db.r5.large–db.r5.24xlarge db.r5b.xlarge — db.r5b.24xlarge db.m5.large–db.m5.24xlarge db.r6i.xlarge — db.r6i.32xlarge db.m6i.xlarge — db.m6i.32xlarge db.x2iedn.xlarge - db.x2iedn.32xlarge
Web Edition	db.r5.large–db.r5.4xlarge db.m5.large–db.m5.4xlarge db.r6i.large–db.r6i.4xlarge db.m6i.large — db.m6i.4xlarge db.r5b.large–db.r5b.4xlarge

I seguenti consigli si applicano ai tipi di classe db.x2iedn:

- Al momento della creazione, l'archiviazione locale è un dispositivo grezzo e non allocato. Prima di utilizzare un'istanza DB con questa classe di istanze, è necessario montare e formattare l'archiviazione locale. Successivamente, `tempdb` configuralo per garantire prestazioni ottimali. Per ulteriori informazioni, consulta Ottimizzazione delle prestazioni tempdb in Amazon RDS Custom for SQL Server utilizzando lo storage a istanze locali.

- Lo storage locale torna allo stato grezzo e non allocato quando si eseguono operazioni su istanze DB come scalabilità, sostituzione di istanze, ripristino di istantanee o ripristino (PITR). point-in-time In queste situazioni, è necessario rimontare, riformattare e riconfigurare l'unità e ripristinare la funzionalità. `tempdb`
- Per le istanze Multi-AZ, si consiglia di eseguire la configurazione su un'istanza DB in standby. In questo modo, se si verifica un failover, il sistema continua a funzionare senza problemi perché la configurazione è già attiva sull'istanza di standby.

Limitazioni di RDS Custom per SQL Server

Le seguenti limitazioni si applicano all'utilizzo di MSDTC su RDS per SQL Server:

- Non puoi creare repliche di lettura in istanze database Amazon RDS per RDS Custom for SQL Server. Tuttavia, è possibile configurare automaticamente la disponibilità elevata tramite l'implementazione Multi-AZ. Per ulteriori informazioni, consulta Gestione di un'implementazione Multi-AZ per RDS Custom per SQL Server.
- Non è possibile modificare l'identificatore di istanza DB di un'istanza DB RDS Custom for SQL Server esistente.
- Per un'istanza DB RDS Custom for SQL Server che non è stata creata con una versione personalizzata del motore (CEV), non è garantito che le modifiche al sistema operativo Microsoft Windows persistano. Ad esempio, queste modifiche si perdono quando si avvia un'istantanea o un'operazione di ripristino. point-in-time Se l'istanza database RDS Custom per SQL Server è stata creata con una CEV, tali modifiche vengono mantenute.
- Non tutte le opzioni sono supportate. Quando, ad esempio, crei un'istanza database di RDS Custom for SQL Server, non puoi eseguire le seguenti operazioni:
  - Modificare il numero di thread per core e di core CPU sulla classe di istanza database.
  - Attivare il calcolo automatico dello storage.
  - Configurazione dell'autenticazione Kerberos utilizzando la AWS Management Console. Tuttavia, è possibile configurare manualmente l'autenticazione Windows e utilizzare Kerberos.
  - Specificare il gruppo di parametri DB, il gruppo di opzioni o il set di caratteri.
  - Attivare Performance Insights.
  - Attivazione degli aggiornamenti a versioni secondarie automatiche.
- Lo storage massimo di istanze DB è 16 TiB.

Collazione e supporto dei caratteri per istanze database RDS Custom per SQL Server

RDS Custom per SQL Server supporta un'ampia gamma di regole di confronto dei server, con codifica tradizionale e UTF-8, per le versioni locali SQL_Latin, giapponese, tedesco e arabo. La regola di confronto predefinita del server è `SQL_Latin1_General_CP1_CI_AS`, tuttavia, puoi selezionare altre regole di confronto supportate da utilizzare. Puoi selezionare un regola confronto utilizzando la stessa procedura utilizzata da RDS per SQL Server. Per ulteriori informazioni, consulta Regole di confronto e set di caratteri per Microsoft SQL Server.

I requisiti e le limitazioni seguenti si applicano quando si utilizzano le regole di confronto del server su RDS Custom per SQL Server:

- Puoi impostare le regole di confronto del server quando crei un'istanza database RDS Custom per SQL Server. Non puoi modificare le regole di confronto a livello di server dopo la creazione dell'istanza database.
- Non puoi modificare le regole di confronto a livello di server durante il ripristino da uno snapshot del database o durante un ripristino point-in-time (PITR).
- Quando crei un'istanza database B da una CEV di RDS Custom per SQL Server, l'istanza database non eredita le regole di confronto del server dalla CEV. Viene invece utilizzata la regola di confronto predefinita del server `SQL_Latin1_General_CP1_CI_AS`. Se hai configurato una regola di confronto server non predefinita su una CEV RDS Custom per SQL Server e desideri utilizzare le stesse regole di confronto del server su una nuova istanza database, assicurati di selezionare le stesse regole di confronto quando crei l'istanza database dalla CEV.

> **Note**
>
> Se le regole di confronto selezionate durante la creazione dell'istanza database sono diverse dalle regole di confronto della CEV, i database di sistema Microsoft SQL Server sulla nuova istanza database di RDS Custom per SQL Server verranno ricompilati per utilizzare le regole di confronto aggiornate. Il processo di ricompilazione viene eseguito solo sulla nuova istanza database RDS Custom per SQL Server e non ha alcun impatto sulla CEV stessa. Eventuali modifiche precedenti apportate ai database di sistema sulla CEV non verranno mantenute nella nuova istanza database RDS Custom per SQL Server una volta ricompilati i database di sistema. Esempi di alcune modifiche includono oggetti definiti dall'utente nel database `master`, processi pianificati nel database o modifiche alle impostazioni predefinite del database `msdb` nel database `model` della CEV. Puoi ricreare

manualmente le modifiche una volta creata la nuova istanza database RDS Custom per SQL Server.

- Quando crei un'istanza database da una versione del motore personalizzato (CEV) di RDS Custom per SQL Server e selezioni una regola di confronto diversa da quella dell CEV, assicurati che l'immagine "gold" (AMI) utilizzata per la creazione della CEV soddisfi i seguenti requisiti in modo che i database di sistema Microsoft SQL Server sulla nuova istanza database possano essere ricompilati:
  - Per SQL Server 2022, assicurati che il `setup.exe` file si trovi nel seguente percorso: `C:\Program Files\Microsoft SQL Server\160\Setup Bootstrap \SQL2022\setup.exe`
  - Per SQL Server 2019, assicurati che il file `setup.exe` si trovi nel seguente percorso: `C:\Program Files\Microsoft SQL Server\150\Setup Bootstrap \SQL2019\setup.exe`
  - Le copie dei modelli di dati e log per i database `master`, `model` e `msdb` devono esistere nelle rispettive posizioni predefinite. Per ulteriori informazioni, consulta Ricompilare i database di sistema nella documentazione Microsoft pubblica.
  - Assicurati che il motore di database SQL Server utilizzi `NT Service\MSSQLSERVER` o `NT AUTHORITY\NETWORK SERVICE` come account di servizio. Qualsiasi altro account non disporrà delle autorizzazioni necessarie sull'unità `C:\` durante la configurazione di una regola di confronto non predefinita del server per l'istanza database.
- Se le regole di confronto del server selezionate per una nuova istanza database sono le stesse configurate sulla CEV, i database di sistema Microsoft SQL Server sulla nuova istanza database RDS Custom per SQL Server non vengono sottoposti al processo di ricompilazione. Qualsiasi modifica precedente apportata ai database di sistema sulla CEV verrà automaticamente mantenuta nella nuova istanza database di RDS Custom per SQL Server.

Puoi impostare la regola di confronto su uno dei valori elencati nella tabella qui di seguito.

Collazione dei server	Descrizione
Arabic_100_bin	Arabic-100, ordinamento binario
Arabico_100_bin2	Arabic-100, tipo di confronto dei punti in codice binario
Arabic_100_CI_AI	Arabic-100, senza distinzione tra maiuscole e minuscole,

Arabico_100_CI_AI_KS	Arabic-100, senza distinzione tra maiuscole e minuscole,
Arabico_100_CI_AI_KS_SC	Caratteri supplementari in arabo 100, senza distinzione tr accenti, tipo Kana e larghezza
Arabico_100_CI_AI_KS_SC_UTF8	Arabico-100, senza distinzione tra maiuscole e minuscole supplementari, UTF8
Arabo_100_CI_AI_KS_WS	Arabic-100, senza distinzione tra maiuscole e minuscole,
Arabico_100_CI_AI_KS_WS_SC	Caratteri supplementari in arabo 100, senza distinzione tr accenti, tipo Kana e larghezza
Arabico_100_CI_AI_KS_WS_SC_UTF8	Arabico-100, senza distinzione tra maiuscole e minuscole tari, UTF8
Arabo_100_CI_AI_SC	Arabic-100, senza distinzione tra maiuscole e minuscole, tra maiuscole e minuscole, caratteri supplementari
Arabico_100_CI_AI_SC_UTF8	Arabico-100, senza distinzione tra maiuscole e minuscole supplementari, UTF8
Arabo_100_CI_AI_WS	Arabic-100, senza distinzione tra maiuscole e minuscole,
Arabico_100_CI_AI_WS_SC	Arabic-100, senza distinzione tra maiuscole e minuscole, tra maiuscole e minuscole, caratteri supplementari
Arabico_100_CI_AI_WS_SC_UTF8	Arabico-100, senza distinzione tra maiuscole e minuscole supplementari, UTF8
Arabico_100_CI_AS	Arabic-100, senza distinzione tra maiuscole e minuscole,
Arabico_100_CI_AS_KS	Arabic-100, senza distinzione tra maiuscole e minuscole,
Arabico_100_CI_AS_KS_SC	Arabic-100, senza distinzione tra maiuscole e minuscole, supplementari
Arabico_100_CI_AS_KS_SC_UTF8	Arabico-100, senza distinzione tra maiuscole e minuscole
Arabico_100_CI_AS_KS_WS	Arabic-100, senza distinzione tra maiuscole e minuscole,

Arabico_100_CI_AS_KS_WS_SC	Arabic-100, senza distinzione tra maiuscole e minuscole,
Arabico_100_CI_AS_KS_WS_SC_UTF8	Arabico-100, senza distinzione tra maiuscole e minuscole UTF8
Arabico_100_CI_AS_SC	Arabic-100, senza distinzione tra maiuscole e minuscole,
Arabico_100_CI_AS_SC_UTF8	Arabico-100, senza distinzione tra maiuscole e minuscole UTF8
Arabico_100_CI_AS_WS	Arabic-100, senza distinzione tra maiuscole e minuscole,
Arabico_100_CI_AS_WS_SC	Arabic-100, senza distinzione tra maiuscole e minuscole,
Arabico_100_CI_AS_WS_SC_UTF8	Arabico-100, senza distinzione tra maiuscole e minuscole UTF8
Arabico_100_CS_AI	Arabic-100, con distinzione tra maiuscole e minuscole, ins
Arabico_100_CS_AI_KS	Arabic-100, distinzione tra maiuscole e minuscole, insens
Arabico_100_CS_AI_KS_SC	Arabic-100, con distinzione tra maiuscole e minuscole, se
Arabico_100_CS_AI_KS_SC_UTF8	Arabico-100, con distinzione tra maiuscole e minuscole, s tari, UTF8
Arabico_100_CS_AI_KS_WS	Arabic-100, con distinzione tra maiuscole e minuscole, ins
Arabico_100_CS_AI_KS_WS_SC	Caratteri supplementari in arabo 100, con distinzione tra
Arabico_100_CS_AI_KS_WS_SC_UTF8	Arabico-100, con distinzione tra maiuscole e minuscole, s
Arabico_100_CS_AI_SC	Arabic-100, con distinzione tra maiuscole e minuscole, se supplementari
Arabico_100_CS_AI_SC_UTF8	Arabico-100, con distinzione tra maiuscole e minuscole, s tari, UTF8
Arabico_100_CS_AI_WS	Arabic-100, distinzione tra maiuscole e minuscole, insens

Arabico_100_CS_AI_WS_SC	Arabic-100, con distinzione tra maiuscole e minuscole, se supplementari
Arabico_100_CS_AI_WS_SC_UTF8	Arabico-100, con distinzione tra maiuscole e minuscole, s supplementari, UTF8
Arabico_100_CS_AS	Arabic-100, con distinzione tra maiuscole e minuscole, se
Arabico_100_CS_AS_KS	Arabic-100, con distinzione tra maiuscole e minuscole, se
Arabico_100_CS_AS_KS_SC	Arabic-100, con distinzione tra maiuscole e minuscole, se
Arabico_100_CS_AS_KS_SC_UTF8	Arabico-100, con distinzione tra maiuscole e minuscole, s
Arabico_100_CS_AS_KS_WS	Arabic-100, con distinzione tra maiuscole e minuscole, se
Arabico_100_CS_AS_KS_WS_SC	Arabic-100, con distinzione tra maiuscole e minuscole, se
Arabico_100_CS_AS_KS_WS_SC_UTF8	Arabico-100, con distinzione tra maiuscole e minuscole, s
Arabico_100_CS_AS_SC	Arabic-100, con distinzione tra maiuscole e minuscole, co
Arabico_100_CS_AS_SC_UTF8	Arabico-100, con distinzione tra maiuscole e minuscole, s supplementari, UTF8
Arabico_100_CS_AS_WS	Arabic-100, sensibile alle maiuscole e minuscole, agli acc
Arabico_100_CS_AS_WS_SC	Arabic-100, con distinzione tra maiuscole e minuscole, se
Arabico_100_CS_AS_WS_SC_UTF8	Arabico-100, con distinzione tra maiuscole e minuscole, s UTF8
Arabic_bin	Arabo, ordinamento binario
Arabic_bin2	Arabo, tipo di confronto dei punti in codice binario
Arabic_CI_AI	Arabo, senza distinzione tra maiuscole e minuscole, inser
Arabico_CI_AI_KS	Arabo, senza distinzione tra maiuscole e minuscole, senz
Arabico_CI_AI_KS_WS	Arabo, senza distinzione tra maiuscole e minuscole, senz

Arabico_CI_AI_WS	Arabo, senza distinzione tra maiuscole e minuscole, inser maiuscole e minuscole
Arabic_CI_AS	Arabo, senza distinzione tra maiuscole e minuscole, con d larghezza
Arabico_CI_AS_KS	Arabo, senza distinzione tra maiuscole e minuscole, sens
Arabico_CI_AS_KS_WS	Arabo, senza distinzione tra maiuscole e minuscole, sens
Arabico_CI_AS_WS	Arabo, senza distinzione tra maiuscole e minuscole, sens
Arabico_CS_AI	Arabo, con distinzione tra maiuscole e minuscole, insensi
Arabico_CS_AI_KS	Arabo, con distinzione tra maiuscole e minuscole, senza d
Arabico_CS_AI_KS_WS	Arabo, con distinzione tra maiuscole e minuscole, senza a
Arabico_CS_AI_WS	Arabo, con distinzione tra maiuscole e minuscole, insensi
Arabic_cs_as	Arabo, sensibile alle maiuscole e minuscole, agli accenti,
Arabico_CS_AS_KS	Arabo, con distinzione tra maiuscole e minuscole, sensibi
Arabico_CS_AS_KS_WS	Arabo, con distinzione tra maiuscole e minuscole, sensibi
Arabico_CS_AS_WS	Arabo, sensibile alle maiuscole e minuscole, agli accenti,
Chinese_PRC_BIN2	Ordinamento di confronto dei punti in codice binario tra C
Chinese_PRC_CI_AS	Chinese-PRC, case-insensitive, accent-sensitive, kanatyp
Chinese_Taiwan_Stroke_CI_AS	Chinese-Taiwan-Stroke, case-insensitive, accent-sensitiv
Danish_Norwegian_CI_AS	Danish-Norwegian, case-insensitive, accent-sensitive, ka
Finnish_Swedish_CI_AS	Finlandese-svedese, senza distinzione tra maiuscole e m
French_CI_AS	French, case-insensitive, accent-sensitive, kanatype-inse
Tedesco_ PhoneBook _100_BIN	Tedesco- PhoneBook -100, ordinamento binario

Tedesco_ PhoneBook _100_BIN2	German- PhoneBook -100, tipo di confronto dei punti in c
Tedesco_ PhoneBook _100_CI_AI	Tedesco- PhoneBook -100, senza distinzione tra maiusco
PhoneBookTedesco_ _100_CI_AI_KS	Tedesco- PhoneBook -100, senza distinzione tra maiusco
PhoneBookTedesco_ _100_CI_AI_KS_SC	Caratteri supplementari tedeschi PhoneBook -100, senza distinzione tra accenti, kana e larghezze
PhoneBookTedesco_ _100_CI_AI_KS_SC_UTF8	Tedesco- PhoneBook -100, senza distinzione tra maiusco supplementari, UTF8
PhoneBookTedesco_ _100_CI_AI_KS_WS	Tedesco- -100, senza distinzione tra maiuscole e minusc
PhoneBookTedesco_ _100_CI_AI_KS_WS_SC	Caratteri supplementari tedeschi PhoneBook -100, senza distinzione tra accenti, kana e larghezze
PhoneBookTedesco_ _100_CI_AI_KS_WS_SC_UTF8	Tedesco- -100, senza distinzione tra maiuscole e minusc UTF8 PhoneBook
PhoneBookTedesco_ _100_CI_AI_SC	Tedesco- PhoneBook -100, senza distinzione tra maiusco caratteri supplementari
PhoneBookTedesco_ _100_CI_AI_SC_UTF8	Tedesco- PhoneBook -100, senza distinzione tra maiusco caratteri supplementari, UTF8
PhoneBookTedesco_ _100_CI_AI_WS	Tedesco- PhoneBook -100, senza distinzione tra maiusco
PhoneBookTedesco_ _100_CI_AI_WS_SC	Caratteri supplementari tedeschi PhoneBook -100, senza agli accenti, senza distinzione tra maiuscole e minuscole,
PhoneBookTedesco_ _100_CI_AI_WS_SC_UTF8	Tedesco- PhoneBook -100, senza distinzione tra maiusco supplementari, UTF8
PhoneBookTedesco_ _100_CI_AS	Tedesco- PhoneBook -100, senza distinzione tra maiusco
PhoneBookTedesco_ _100_CI_AS_KS	Tedesco- PhoneBook -100, senza distinzione tra maiusco
PhoneBookTedesco_ _100_CI_AS_KS_SC	Caratteri supplementari tedeschi PhoneBook -100, senza

PhoneBookTedesco_ _100_CI_AS_KS_SC_UTF8	Tedesco- PhoneBook -100, senza distinzione tra maiusco supplementari, UTF8
PhoneBookTedesco_ _100_CI_AS_KS_WS	Tedesco- -100, senza distinzione tra maiuscole e minusco
PhoneBookTedesco_ _100_CI_AS_KS_WS_SC	Caratteri supplementari tedeschi PhoneBook -100, senza larghezza
PhoneBookTedesco_ _100_CI_AS_KS_WS_SC_UTF8	Tedesco- -100, senza distinzione tra maiuscole e minusco UTF8 PhoneBook
PhoneBookTedesco_ _100_CI_AS_SC	Tedesco- PhoneBook -100, senza distinzione tra maiusco caratteri supplementari
PhoneBookTedesco_ _100_CI_AS_SC_UTF8	Tedesco- PhoneBook -100, senza distinzione tra maiusco supplementari, UTF8
PhoneBookTedesco_ _100_CI_AS_WS	Tedesco- PhoneBook -100, senza distinzione tra maiusco
PhoneBookTedesco_ _100_CI_AS_WS_SC	Caratteri supplementari tedeschi PhoneBook -100, senza alla larghezza
PhoneBookTedesco_ _100_CI_AS_WS_SC_UTF8	Tedesco- PhoneBook -100, senza distinzione tra maiusco supplementari, UTF8
PhoneBookTedesco_ _100_CS_AI	Tedesco- PhoneBook -100, con distinzione tra maiuscole
PhoneBookTedesco_ _100_CS_AI_KS	Tedesco- PhoneBook -100, con distinzione tra maiuscole
PhoneBookTedesco_ _100_CS_AI_KS_SC	Caratteri supplementari tedeschi PhoneBook -100, con di
PhoneBookTedesco_ _100_CS_AI_KS_SC_UTF8	Tedesco- PhoneBook -100, con distinzione tra maiuscole UTF8
PhoneBookTedesco_ _100_CS_AI_KS_WS	Tedesco- -100, con distinzione tra maiuscole e minuscole
PhoneBookTedesco_ _100_CS_AI_KS_WS_SC	Caratteri supplementari tedeschi PhoneBook -100, con di
PhoneBookTedesco_ _100_CS_AI_KS_WS_SC_UTF8	Tedesco- -100, con distinzione tra maiuscole e minuscole PhoneBook

PhoneBookTedesco_ _100_CS_AI_SC	Tedesco- PhoneBook -100, con distinzione tra maiuscole supplementari
PhoneBookTedesco_ _100_CS_AI_SC_UTF8	Tedesco- PhoneBook -100, con distinzione tra maiuscole supplementari, UTF8
PhoneBookTedesco_ _100_CS_AI_WS	Tedesco- PhoneBook -100, con distinzione tra maiuscole
PhoneBookTedesco_ _100_CS_AI_WS_SC	Caratteri supplementari tedeschi PhoneBook -100, con di alla larghezza
PhoneBookTedesco_ _100_CS_AI_WS_SC_UTF8	Tedesco- PhoneBook -100, con distinzione tra maiuscole supplementari, UTF8
PhoneBookTedesco_ _100_CS_AS	Tedesco- PhoneBook -100, sensibile alle maiuscole e mi
PhoneBookTedesco_ _100_CS_AS_KS	Tedesco- PhoneBook -100, sensibile alle maiuscole e mi
PhoneBookTedesco_ _100_CS_AS_KS_SC	PhoneBookGermanio-100, con distinzione tra maiuscole caratteri supplementari
PhoneBookTedesco_ _100_CS_AS_KS_SC_UTF8	Tedesco- PhoneBook -100, con distinzione tra maiuscole UTF8
PhoneBookTedesco_ _100_CS_AS_KS_WS	Tedesco- -100, sensibile alle maiuscole e minuscole, agli
PhoneBookTedesco_ _100_CS_AS_KS_WS_SC	Caratteri supplementari tedeschi PhoneBook -100, con di larghezza
PhoneBookTedesco_ _100_CS_AS_KS_WS_SC_UTF8	Tedesco- -100, con distinzione tra maiuscole e minuscole UTF8 PhoneBook
PhoneBookTedesco_ _100_CS_AS_SC	Caratteri supplementari tedeschi PhoneBook -100, con di larghezza
PhoneBookTedesco_ _100_CS_AS_SC_UTF8	Tedesco- PhoneBook -100, con distinzione tra maiuscole supplementari, UTF8
PhoneBookTedesco_ _100_CS_AS_WS	Tedesco- PhoneBook -100, sensibile alle maiuscole e mi

PhoneBookTedesco_ _100_CS_AS_WS_SC	Caratteri supplementari tedeschi PhoneBook -100, con di alla larghezza
PhoneBookTedesco_ _100_CS_A S_WS_SC_UTF8	Tedesco- PhoneBook -100, con distinzione tra maiuscole supplementari, UTF8
PhoneBookTedesco_ _BIN	Tedesco-PhoneBook, ordinamento binario
Tedesco_ PhoneBook _BIN2	Tipo di confronto dei punti PhoneBook in codice binario te
Tedesco_ PhoneBook _CI_AI	Tedesco, senza distinzione tra maiuscole e PhoneBook m
PhoneBookTedesco_ _CI_AI_KS	Tedesco, senza distinzione tra maiuscole e minuscole, ins
PhoneBookTedesco_ _CI_AI_KS_WS	Tedesco, senza distinzione tra maiuscole e minuscole, ins
PhoneBookTedesco_ _CI_AI_WS	Tedesco, senza distinzione tra maiuscole e minuscole, ins
PhoneBookTedesco_ _CI_AS	Tedesco, senza distinzione tra maiuscole e PhoneBook m
PhoneBookTedesco_ _CI_AS_KS	Tedesco, senza distinzione tra maiuscole e minuscole, se
PhoneBookTedesco_ _CI_AS_KS_WS	Tedesco, senza distinzione tra maiuscole e minuscole, se
PhoneBookTedesco_ _CI_AS_WS	Tedesco, senza distinzione tra maiuscole e minuscole, se
PhoneBookTedesco_ _CS_AI	Tedesco, con distinzione tra maiuscole e minuscolePhone
PhoneBookTedesco_ _CS_AI_KS	Tedesco, con distinzione tra maiuscole e minuscole, inser
PhoneBookTedesco_ _CS_AI_KS_WS	Tedesco, con distinzione tra maiuscole e minuscole, inser
PhoneBookTedesco_ _CS_AI_WS	Tedesco, con distinzione tra maiuscole e minuscole, inser
PhoneBookTedesco_ _CS_AS	Tedesco, con distinzione tra maiuscole e minuscolePhone
PhoneBookTedesco_ _CS_AS_KS	Tedesco, con distinzione tra maiuscole e minuscole, sens
PhoneBookTedesco_ _CS_AS_KS_WS	Tedesco, con distinzione tra maiuscole e minuscole, sens
PhoneBookTedesco_ _CS_AS_WS	Tedesco, con distinzione tra maiuscole e minuscole, sens

Hebrew_BIN	Hebrew, binary sort
Hebrew_CI_AS	Ebraico, non sensibile al maiuscolo/minuscolo, sensibile a
Japanese_90_bin	Japanese-90, ordinamento binario
Japanese_90_bin2	Japanese-90, tipo di confronto di punti in codice binario
Japanese_90_CI_AI	Giapponese-90, senza distinzione tra maiuscole e minusc
Giapponese_90_CI_AI_KS	Giapponese-90, senza distinzione tra maiuscole e minusc
Giapponese_90_CI_AI_KS_SC	Japanese-90, senza distinzione tra maiuscole e minuscol supplementari
Giapponese_90_CI_AI_KS_SC_UTF8	Japanese-90, senza distinzione tra maiuscole e minuscol supplementari, UTF8
Giapponese_90_CI_AI_KS_WS	Giapponese-90, senza distinzione tra maiuscole e minusc
Giapponese_90_CI_AI_KS_WS_SC	Japanese-90, senza distinzione tra maiuscole e minuscol supplementari
Giapponese_90_CI_AI_KS_WS_SC_UTF8	Japanese-90, senza distinzione tra maiuscole e minuscol
Giapponese_90_CI_AI_SC	Japanese-90, senza distinzione tra maiuscole e minuscol caratteri supplementari
Giapponese_90_CI_AI_SC_UTF8	Japanese-90, senza distinzione tra maiuscole e minuscol supplementari, UTF8
Giapponese_90_CI_AI_WS	Giapponese-90, senza distinzione tra maiuscole e minusc
Giapponese_90_CI_AI_WS_SC	Japanese-90, senza distinzione tra maiuscole e minuscol maiuscole e minuscole, caratteri supplementari
Giapponese_90_CI_AI_WS_SC_UTF8	Japanese-90, senza distinzione tra maiuscole e minuscol UTF8
Giapponese_90_CI_AS	Giapponese-90, senza distinzione tra maiuscole e minusc

Giapponese_90_CI_AS_KS	Giapponese-90, senza distinzione tra maiuscole e minusc
Giapponese_90_CI_AS_KS_SC	Japanese-90, senza distinzione tra maiuscole e minuscol
Giapponese_90_CI_AS_KS_SC_UTF8	Japanese-90, senza distinzione tra maiuscole e minuscol
Giapponese_90_CI_as_KS_WS	Giapponese-90, senza distinzione tra maiuscole e minusc
Giapponese_90_CI_AS_KS_WS_SC	Japanese-90, senza distinzione tra maiuscole e minuscol
Giapponese_90_CI_AS_KS_WS_SC_UTF8	Japanese-90, senza distinzione tra maiuscole e minuscol UTF8
Giapponese_90_CI_AS_SC	Japanese-90, senza distinzione tra maiuscole e minuscol minuscole, caratteri supplementari
Giapponese_90_CI_AS_SC_UTF8	Japanese-90, senza distinzione tra maiuscole e minuscol maiuscole e minuscole, UTF8
Giapponese_90_CI_AS_WS	Giapponese-90, senza distinzione tra maiuscole e minusc
Giapponese_90_CI_AS_WS_SC	Japanese-90, senza distinzione tra maiuscole e minuscol maiuscole e minuscole, caratteri supplementari
Giapponese_90_CI_AS_WS_SC_UTF8	Japanese-90, senza distinzione tra maiuscole e minuscol UTF8
Giapponese_90_CS_AI	Giapponese-90, con distinzione tra maiuscole e minuscol
Giapponese_90_CS_AI_KS	Giapponese-90, con distinzione tra maiuscole e minuscol
Giapponese_90_CS_AI_KS_SC	Japanese-90, con distinzione tra maiuscole e minuscole,
Giapponese_90_CS_AI_KS_SC_UTF8	Japanese-90, con distinzione tra maiuscole e minuscole,
Giapponese_90_CS_AI_KS_WS	Giapponese-90, con distinzione tra maiuscole e minuscol
Giapponese_90_CS_AI_KS_WS_SC	Japanese-90, con distinzione tra maiuscole e minuscole,
Giapponese_90_CS_AI_KS_WS_SC_UTF8	Japanese-90, con distinzione tra maiuscole e minuscole,

Giapponese_90_CS_AI_SC	Japanese-90, con distinzione tra maiuscole e minuscole, supplementari
Giapponese_90_CS_AI_SC_UTF8	Japanese-90, con distinzione tra maiuscole e minuscole, supplementari, UTF8
Giapponese_90_CS_AI_WS	Giapponese-90, con distinzione tra maiuscole e minuscol
Giapponese_90_CS_AI_WS_SC	Japanese-90, con distinzione tra maiuscole e minuscole, minuscole, caratteri supplementari
Giapponese_90_CS_AI_WS_SC_UTF8	Japanese-90, con distinzione tra maiuscole e minuscole, UTF8
Giapponese_90_cs_AS	Giapponese-90, sensibile alle maiuscole e minuscole, agl
Giapponese_90_CS_AS_KS	Giapponese-90, con distinzione tra maiuscole e minuscol
Giapponese_90_CS_AS_KS_SC	Japanese-90, con distinzione tra maiuscole e minuscole,
Giapponese_90_CS_AS_KS_SC_UTF8	Japanese-90, con distinzione tra maiuscole e minuscole, UTF8
Giapponese_90_CS_AS_KS_WS	Giapponese-90, con distinzione tra maiuscole e minuscol
Giapponese_90_CS_AS_KS_WS_SC	Japanese-90, con distinzione tra maiuscole e minuscole,
Giapponese_90_CS_AS_KS_WS_SC_UTF8	Japanese-90, con distinzione tra maiuscole e minuscole, UTF8
Giapponese_90_CS_AS_SC	Japanese-90, con distinzione tra maiuscole e minuscole, minuscole, caratteri supplementari
Giapponese_90_CS_AS_SC_UTF8	Japanese-90, con distinzione tra maiuscole e minuscole, supplementari, UTF8
Giapponese_90_cs_as_WS	Giapponese-90, sensibile alle maiuscole e minuscole, agl
Giapponese_90_CS_AS_WS_SC	Japanese-90, con distinzione tra maiuscole e minuscole,

Giapponese_90_CS_AS_WS_SC_UTF8	Japanese-90, con distinzione tra maiuscole e minuscole, UTF8
Japanese_BIN	Giapponese, ordinamento binario
Japanese_bin2	Giapponese, tipo di confronto dei punti in codice binario
Japanese_Bushu_Kakusu_100_bin	Japanese-Bushu-Kakusu-100, ordinamento binario
Japanese_Bushu_Kakusu_100_bin2	Japanese-Bushu-Kakusu-100, tipo di confronto dei punti i
Japanese_Bushu_Kakusu_100_CI_AI	Japanese-Bushu-Kakusu-100, senza distinzione tra maiu
Giapponese_Bushu_Kakusu_100_CI_AI_KS	Japanese-Bushu-Kakusu-100, senza distinzione tra maiu
Giapponese_Bushu_Kakusu_100_CI_AI_KS_SC	Japanese-Bushu-Kakusu-100, senza distinzione tra maiu caratteri supplementari
Giapponese_Bushu_Kakusu_100_CI_AI_KS_SC_UTF8	Japanese-Bushu-Kakusu-100, caratteri supplementari se distinzione tra maiuscole e minuscole, UTF8
Giapponese_Bushu_Kakusu_100_CI_AI_KS_WS	Japanese-Bushu-Kakusu-100, senza distinzione tra maiu
Giapponese_Bushu_Kakusu_100_CI_AI_KS_WS_SC	Japanese-Bushu-Kakusu-100, senza distinzione tra maiu kana e caratteri supplementari
Giapponese_Bushu_Kakusu_100_CI_AI_KS_WS_SC_UTF8	Japanese-Bushu-Kakusu-100, caratteri supplementari se
Giapponese_Bushu_Kakusu_100_CI_AI_SC	Japanese-Bushu-Kakusu-100, senza distinzione tra maiu supplementari
Giapponese_Bushu_Kakusu_100_CI_AI_SC_UTF8	Japanese-Bushu-Kakusu-100, caratteri supplementari se distinzione tra maiuscole e minuscole, UTF8
Giapponese_Bushu_Kakusu_100_CI_AI_WS	Japanese-Bushu-Kakusu-100, senza distinzione tra maiu
Giapponese_Bushu_Kakusu_100_CI_AI_WS_SC	Japanese-Bushu-Kakusu-100, senza distinzione tra maiu insensibili ai tipi kana, caratteri supplementari

Giapponese_Bushu_Kakusu_100_CI_AI_WS _SC_UTF8	Japanese-Bushu-Kakusu-100, caratteri supplementari se distinzione tra maiuscole e minuscole, UTF8
Giapponese_Bushu_Kakusu_100_CI_AS	Japanese-Bushu-Kakusu-100, senza distinzione tra maiu
Giapponese_Bushu_Kakusu_100_CI_AS_KS	Japanese-Bushu-Kakusu-100, senza distinzione tra maiu
Giapponese_Bushu_Kakusu_100_CI_as_KS _SC	Japanese-Bushu-Kakusu-100, senza distinzione tra maiu supplementari
Giapponese_Bushu_Kakusu_100_CI_AS_KS _SC_UTF8	Japanese-Bushu-Kakusu-100, caratteri supplementari se
Bushu_Kakusu_100_CI_AS_KS_WS in giapponese	Japanese-Bushu-Kakusu-100, senza distinzione tra maiu
Giapponese_Bushu_Kakusu_100_CI_AS_KS _WS_SC	Japanese-Bushu-Kakusu-100, senza distinzione tra maiu supplementari
Giapponese_Bushu_Kakusu_100_CI_AS_KS _WS_SC_UTF8	Japanese-Bushu-Kakusu-100, caratteri supplementari se UTF8
Giapponese_Bushu_Kakusu_100_CI_AS_SC	Japanese-Bushu-Kakusu-100, senza distinzione tra maiu caratteri supplementari
Giapponese_Bushu_Kakusu_100_CI_AS_SC _UTF8	Japanese-Bushu-Kakusu-100, caratteri supplementari se larghezza, UTF8
Bushu_Kakusu_100_CI_AS_WS in giapponese	Japanese-Bushu-Kakusu-100, senza distinzione tra maiu
Giapponese_Bushu_Kakusu_100_CI_as_WS _SC	Japanese-Bushu-Kakusu-100, senza distinzione tra maiu supplementari
Giapponese_Bushu_Kakusu_100_CI_AS_WS _SC_UTF8	Japanese-Bushu-Kakusu-100, caratteri supplementari se , UTF8
Giapponese_Bushu_Kakusu_100_CS_AI	Japanese-Bushu-Kakusu-100, con distinzione tra maiusc
Giapponese_Bushu_Kakusu_100_CS_AI_KS	Japanese-Bushu-Kakusu-100, con distinzione tra maiusc

Giapponese_Bushu_Kakusu_100_CS_AI_KS_SC	Japanese-Bushu-Kakusu-100, con distinzione tra maiusc tari
Giapponese_Bushu_Kakusu_100_CS_AI_KS_SC_UTF8	Japanese-Bushu-Kakusu-100, caratteri supplementari co
Giapponese_Bushu_Kakusu_100_CS_AI_KS_WS	Japanese-Bushu-Kakusu-100, con distinzione tra maiusc
Giapponese_Bushu_Kakusu_100_CS_AI_KS_WS_SC	Japanese-Bushu-Kakusu-100, con distinzione tra maiusc supplementari
Giapponese_Bushu_Kakusu_100_CS_AI_KS_WS_SC_UTF8	Japanese-Bushu-Kakusu-100, caratteri supplementari co
Giapponese_Bushu_Kakusu_100_CS_AI_SC	Japanese-Bushu-Kakusu-100, con distinzione tra maiusc supplementari
Giapponese_Bushu_Kakusu_100_CS_AI_SC_UTF8	Japanese-Bushu-Kakusu-100, caratteri supplementari co larghezza, UTF8
Giapponese_Bushu_Kakusu_100_CS_AI_WS	Japanese-Bushu-Kakusu-100, con distinzione tra maiusc
Giapponese_Bushu_Kakusu_100_CS_AI_WS_SC	Japanese-Bushu-Kakusu-100, con distinzione tra maiusc supplementari
Giapponese_Bushu_Kakusu_100_CS_AI_WS_SC_UTF8	Japanese-Bushu-Kakusu-100, caratteri supplementari co larghezza, UTF8
Giapponese_Bushu_Kakusu_100_CS_AS	Japanese-Bushu-Kakusu-100, sensibile alle maiuscole e
Giapponese_Bushu_Kakusu_100_cs_as_KS	Japanese-Bushu-Kakusu-100, sensibile alle maiuscole e
Giapponese_Bushu_Kakusu_100_CS_AS_KS_SC	Japanese-Bushu-Kakusu-100, caratteri supplementari co
Giapponese_Bushu_Kakusu_100_CS_AS_KS_SC_UTF8	Japanese-Bushu-Kakusu-100, caratteri supplementari co

Giapponese_Bushu_Kakusu_100_CS_AS_KS_WS	Japanese-Bushu-Kakusu-100, con distinzione tra maiusc
Giapponese_Bushu_Kakusu_100_CS_AS_KS_WS_SC	Japanese-Bushu-Kakusu-100, caratteri supplementari co
Giapponese_Bushu_Kakusu_100_CS_AS_KS_WS_SC_UTF8	Japanese-Bushu-Kakusu-100, caratteri supplementari co
Giapponese_Bushu_Kakusu_100_CS_AS_SC	Japanese-Bushu-Kakusu-100, con distinzione tra maiusc supplementari
Giapponese_Bushu_Kakusu_100_CS_AS_SC_UTF8	Japanese-Bushu-Kakusu-100, caratteri supplementari co larghezza, UTF8
Bushu_Kakusu_100_CS_AS_WS in giapponese	Japanese-Bushu-Kakusu-100, sensibile alle maiuscole e
Giapponese_Bushu_Kakusu_100_CS_AS_WS_SC	Japanese-Bushu-Kakusu-100, con distinzione tra maiusc supplementari
Giapponese_Bushu_Kakusu_100_CS_AS_WS_SC_UTF8	Japanese-Bushu-Kakusu-100, caratteri supplementari co UTF8
Japanese_Bushu_Kakusu_140_bin	Japanese-Bushu-Kakusu-140, ordinamento binario
Japanese_Bushu_Kakusu_140_bin2	Japanese-Bushu-Kakusu-140, tipo di confronto dei punti i
Japanese_Bushu_Kakusu_140_CI_AI	Japanese-Bushu-Kakusu-140, senza distinzione tra maiu caratteri supplementari, selettore di variazione insensibile
Giapponese_Bushu_Kakusu_140_CI_AI_KS	Japanese-Bushu-Kakusu-140, senza distinzione tra maiu caratteri supplementari, selettore di variazione insensibile
Giapponese_Bushu_Kakusu_140_CI_AI_KS_UTF8	Japanese-Bushu-Kakusu-140, senza distinzione tra maiu maiuscole e minuscole, insensibile all'accento, UTF8
Bushu_Kakusu_140_CI_AI_KS_VSS in giapponese	Japanese-Bushu-Kakusu-140, senza distinzione tra maiu maiuscole e minuscole, senza distinzione tra accenti, sen

Japanese_Bushu_Kakusu_140_CI_AI_KS_VSS_UTF8	Japanese-Bushu-Kakusu-140, caratteri supplementari se distinzione tra maiuscole e minuscole, insensibile al selet
Bushu_Kakusu_140_CI_AI_KS_WS in giapponese	Japanese-Bushu-Kakusu-140, senza distinzione tra maiu supplementari, selettore di variazione insensibile
Japanese_Bushu_Kakusu_140_CI_AI_KS_WS_UTF8	Japanese-Bushu-Kakusu-140, caratteri supplementari se distinzione tra maiuscole e minuscole, senza distinzione t
Bushu_Kakusu_140_CI_AI_KS_WS_VSS in giapponese	Japanese-Bushu-Kakusu-140, caratteri supplementari se distinzione tra accenti, sensibile al selettore di variazione
Japanese_Bushu_Kakusu_140_CI_AI_KS_WS_VSS_UTF8	Japanese-Bushu-Kakusu-140, caratteri supplementari se distinzione tra maiuscole e minuscole, insensibile al selet
Giapponese_Bushu_Kakusu_140_CI_AI_UTF8	Japanese-Bushu-Kakusu-140, senza distinzione tra maiu caratteri supplementari, selettore di variazione insensibile
Bushu_Kakusu_140_CI_AI_VSS in giapponese	Japanese-Bushu-Kakusu-140, senza distinzione tra maiu supplementari, sensibile al selettore di variazione
Japanese_Bushu_Kakusu_140_CI_AI_VSS_UTF8	Japanese-Bushu-Kakusu-140, senza distinzione tra maiu maiuscole e minuscole, insensibile al selettore di variazio
Giapponese_Bushu_Kakusu_140_CI_AI_WS	Japanese-Bushu-Kakusu-140, senza distinzione tra maiu supplementari, selettore di variazione insensibile
Giapponese_Bushu_Kakusu_140_CI_AI_WS_UTF8	Japanese-Bushu-Kakusu-140, senza distinzione tra maiu maiuscole e minuscole, senza distinzione tra maiuscole e
Bushu_Kakusu_140_CI_AI_WS_VSS in giapponese	Japanese-Bushu-Kakusu-140, senza distinzione tra maiu maiuscole e minuscole, insensibile al selettore di variazio
Japanese_Bushu_Kakusu_140_CI_AI_WS_VSS_UTF8	Japanese-Bushu-Kakusu-140, caratteri supplementari se sensibili alla larghezza, supplementari, sensibile al selett
Giapponese_Bushu_Kakusu_140_CI_AS	Japanese-Bushu-Kakusu-140, senza distinzione tra maiu caratteri supplementari, selettore di variazione insensibile

Giapponese_Bushu_Kakusu_140_CI_as_KS	Japanese-Bushu-Kakusu-140, senza distinzione tra maiu caratteri supplementari, selettore di variazione insensibile
Japanese_Bushu_Kakusu_140_CI_as_KS_UTF8	Japanese-Bushu-Kakusu-140, caratteri supplementari se supplementari, selettore di variazione, UTF8
Bushu_Kakusu_140_CI_as_KS_VSS in giapponese	Japanese-Bushu-Kakusu-140, senza distinzione tra maiu supplementari, sensibile al selettore di variazione
Japanese_Bushu_Kakusu_140_CI_AS_KS_VSS_UTF8	Japanese-Bushu-Kakusu-140, caratteri supplementari se sensibili al selettore di variazione, UTF8
Bushu_Kakusu_140_CI_as_KS_WS in giapponese	Japanese-Bushu-Kakusu-140, senza distinzione tra maiu supplementari, selettore di variazione insensibile
Japanese_Bushu_Kakusu_140_CI_AS_KS_WS_UTF8	Japanese-Bushu-Kakusu-140, caratteri supplementari se supplementari, selettore di variazione insensibile, UTF8
Bushu_Kakusu_140_CI_AS_KS_WS_VSS in giapponese	Japanese-Bushu-Kakusu-140, senza distinzione tra maiu supplementari, sensibile al selettore di variazione
Bushu_Kakusu_140_CI_AS_KS_WS_VSS_UTF8	Japanese-Bushu-Kakusu-140, caratteri supplementari se sensibili al selettore di variazione, UTF8
Giapponese_Bushu_Kakusu_140_CI_AS_UTF8	Japanese-Bushu-Kakusu-140, senza distinzione tra maiu caratteri supplementari, selettore di variazione insensibile
Bushu_Kakusu_140_CI_as_VSS in giapponese	Japanese-Bushu-Kakusu-140, senza distinzione tra maiu supplementari, sensibile al selettore di variazione
Japanese_Bushu_Kakusu_140_CI_AS_VSS_UTF8	Japanese-Bushu-Kakusu-140, caratteri supplementari se insensibile alla larghezza, caratteri supplementari, sensib
Giapponese_Bushu_Kakusu_140_CI_AS_WS	Japanese-Bushu-Kakusu-140, senza distinzione tra maiu supplementari, selettore di variazione insensibile
Japanese_Bushu_Kakusu_140_CI_as_WS_UTF8	Japanese-Bushu-Kakusu-140, caratteri supplementari se larghezza, caratteri supplementari, selettore di variazione

Bushu_Kakusu_140_CI_as_WS_VSS in giapponese	Japanese-Bushu-Kakusu-140, senza distinzione tra maiu supplementari, sensibile al selettore di variazione
Japanese_Bushu_Kakusu_140_CI_AS_WS_VSS_UTF8	Japanese-Bushu-Kakusu-140, caratteri supplementari se sensibili alla larghezza, supplementari, sensibile al seletto
Giapponese_Bushu_Kakusu_140_CS_AI	Japanese-Bushu-Kakusu-140, con distinzione tra maiusc caratteri supplementari, selettore di variazione insensibile
Giapponese_Bushu_Kakusu_140_CS_AI_KS	Japanese-Bushu-Kakusu-140, con distinzione tra maiusc supplementari, selettore di variazione insensibile
Giapponese_Bushu_Kakusu_140_CS_AI_KS_UTF8	Japanese-Bushu-Kakusu-140, con distinzione tra maiusc minuscole, insensibile all'accento, UTF8
Giapponese_Bushu_Kakusu_140_CS_AI_KS_VSS	Japanese-Bushu-Kakusu-140, con distinzione tra maiusc supplementari, sensibile al selettore di variazione
Japanese_Bushu_Kakusu_140_CS_AI_KS_VSS_UTF8	Japanese-Bushu-Kakusu-140, caratteri supplementari co sensibili al selettore di variazione, UTF8
Giapponese_Bushu_Kakusu_140_CS_AI_KS_WS	Japanese-Bushu-Kakusu-140, con distinzione tra maiusc supplementari, selettore di variazione insensibile
Japanese_Bushu_Kakusu_140_CS_AI_KS_WS_UTF8	Japanese-Bushu-Kakusu-140, caratteri supplementari co supplementari, selettore di variazione insensibile, UTF8
Giapponese_Bushu_Kakusu_140_CS_AI_KS_WS_VSS	Japanese-Bushu-Kakusu-140, caratteri con distinzione tra caratteri supplementari, sensibile al selettore di variazione
Japanese_Bushu_Kakusu_140_CS_AI_KS_WS_VSS_UTF8	Japanese-Bushu-Kakusu-140, caratteri supplementari co al selettore di variazione, UTF8
Giapponese_Bushu_Kakusu_140_CS_AI_UTF8	Japanese-Bushu-Kakusu-140, con distinzione tra maiusc supplementari, selettore di variazione insensibile, UTF8
Giapponese_Bushu_Kakusu_140_CS_AI_VSS	Japanese-Bushu-Kakusu-140, con distinzione tra maiusc supplementari, sensibile al selettore di variazione

Japanese_Bushu_Kakusu_140_CS_AI_VSS_UTF8	Japanese-Bushu-Kakusu-140, caratteri supplementari co larghezza, caratteri supplementari, sensibile al selettore c
Giapponese_Bushu_Kakusu_140_CS_AI_WS	Japanese-Bushu-Kakusu-140, con distinzione tra maiusc supplementari, selettore di variazione insensibile
Giapponese_Bushu_Kakusu_140_CS_AI_WS_UTF8	Japanese-Bushu-Kakusu-140, con distinzione tra maiusc supplementari, selettore di variazione insensibile, UTF8
Giapponese_Bushu_Kakusu_140_CS_AI_WS_VSS	Japanese-Bushu-Kakusu-140, con distinzione tra maiusc supplementari, sensibile al selettore di variazione
Japanese_Bushu_Kakusu_140_CS_AI_WS_VSS_UTF8	Japanese-Bushu-Kakusu-140, caratteri supplementari co larghezza, supplementari, sensibile al selettore di variazio
Giapponese_Bushu_Kakusu_140_CS_AS	Japanese-Bushu-Kakusu-140, con distinzione tra maiusc supplementari, selettore di variazione insensibile al selett
Japanese_Bushu_Kakusu_140_CS_AS_KS	Japanese-Bushu-Kakusu-140, con distinzione tra maiusc selettore di variazione insensibile
Japanese_Bushu_Kakusu_140_CS_AS_KS_UTF8	Japanese-Bushu-Kakusu-140, caratteri supplementari co supplementari, selettore di variazione, UTF8
Giapponese_Bushu_Kakusu_140_CS_AS_KS_VSS	Japanese-Bushu-Kakusu-140, caratteri supplementari co supplementari, sensibile al selettore di variazione
Japanese_Bushu_Kakusu_140_CS_AS_KS_VSS_UTF8	Japanese-Bushu-Kakusu-140, caratteri supplementari co selettore di variazione, UTF8
Giapponese_Bushu_Kakusu_140_CS_AS_KS_WS	Japanese-Bushu-Kakusu-140, con distinzione tra maiusc supplementari, selettore di variazione insensibile
Japanese_Bushu_Kakusu_140_CS_AS_KS_WS_UTF8	Japanese-Bushu-Kakusu-140, caratteri supplementari co supplementari, selettore di variazione insensibile, UTF8
Giapponese_Bushu_Kakusu_140_CS_AS_KS_WS_VSS	Japanese-Bushu-Kakusu-140, con distinzione tra maiusc supplementari, sensibile al selettore di variazione

Japanese_Bushu_Kakusu_140_CS_AS_KS_WS_VSS_UTF8	Japanese-Bushu-Kakusu-140, caratteri supplementari co selettore di variazione, UTF8
Giapponese_Bushu_Kakusu_140_CS_AS_UTF8	Japanese-Bushu-Kakusu-140, con distinzione tra maiusc tari, selettore di variazione, UTF8
Giapponese_Bushu_Kakusu_140_CS_as_VSS	Japanese-Bushu-Kakusu-140, con distinzione tra maiusc supplementari, sensibile al selettore di variazione
Japanese_Bushu_Kakusu_140_CS_AS_VSS_UTF8	Japanese-Bushu-Kakusu-140, caratteri supplementari co larghezza, caratteri supplementari, sensibile al selettore d
Giapponese_Bushu_Kakusu_140_CS_AS_WS	Japanese-Bushu-Kakusu-140, con distinzione tra maiusc supplementari, selettore di variazione insensibile
Japanese_Bushu_Kakusu_140_CS_AS_WS_UTF8	Japanese-Bushu-Kakusu-140, caratteri supplementari co caratteri supplementari, selettore di variazione, UTF8
Giapponese_Bushu_Kakusu_140_CS_AS_WS_VSS	Japanese-Bushu-Kakusu-140, con distinzione tra maiusc supplementari, sensibile al selettore di variazione
Japanese_Bushu_Kakusu_140_CS_AS_WS_VSS_UTF8	Japanese-Bushu-Kakusu-140, caratteri supplementari co sensibili al selettore di variazione, UTF8
Japanese_CI_AI	Giapponese, senza distinzione tra maiuscole e minuscole
Giapponese_CI_AI_KS	Giapponese, senza distinzione tra maiuscole e minuscole
Giapponese_CI_AI_KS_WS	Giapponese, senza distinzione tra maiuscole e minuscole
Japanese_CI_AI_WS	Giapponese, senza distinzione tra maiuscole e minuscole maiuscole e minuscole
Japanese_CI_AS	Japanese, case-insensitive, accent-sensitive, kanatype-in
Giapponese_CI_AS_KS	Giapponese, senza distinzione tra maiuscole e minuscole
Giapponese_CI_AS_KS_WS	Giapponese, senza distinzione tra maiuscole e minuscole
Giapponese_CI_AS_WS	Giapponese, senza distinzione tra maiuscole e minuscole maiuscole e minuscole

Japanese_CS_AI	Giapponese, con distinzione tra maiuscole e minuscole, s
Giapponese_cs_AI_KS	Giapponese, con distinzione tra maiuscole e minuscole, s
Giapponese_cs_AI_KS_WS	Giapponese, con distinzione tra maiuscole e minuscole, s
Giapponese_cs_AI_WS	Giapponese, con distinzione tra maiuscole e minuscole, s
Japanese_CS_AS	Giapponese, non sensibile al maiuscolo/minuscolo, sensi
Giapponese_cs_as_KS	Giapponese, con distinzione tra maiuscole e minuscole, s
Giapponese_CS_AS_KS_WS	Giapponese, con distinzione tra maiuscole e minuscole, s
Japanese_cs_as_WS	Giapponese, con distinzione tra maiuscole e minuscole, s
Japanese_Unicode_bin	Japanese-Unicode, ordinamento binario
Japanese_Unicode_bin2	Japanese-Unicode, tipo di confronto dei punti in codice bi
Japanese_Unicode_CI_AI	Unicode giapponese, senza distinzione tra maiuscole e m
Unicode_ci_ai_ks in giapponese	Unicode giapponese, senza distinzione tra maiuscole e m
Unicode_giapponese_CI_AI_KS_WS	Unicode giapponese, senza distinzione tra maiuscole e m
Japanese_Unicode_CI_AI_WS	Unicode giapponese, senza distinzione tra maiuscole e m
Japanese_Unicode_CI_AS	Unicode giapponese, senza distinzione tra maiuscole e m
Unicode_ci_as_KS in giapponese	Unicode giapponese, senza distinzione tra maiuscole e m
Unicode_giapponese_CI_AS_KS_WS	Unicode giapponese, senza distinzione tra maiuscole e m
Unicode_ci_as_WS in giapponese	Unicode giapponese, senza distinzione tra maiuscole e m
Japanese_Unicode_CS_AI	Unicode giapponese, con distinzione tra maiuscole e min
Unicode_giapponese_cs_AI_KS	Unicode giapponese, con distinzione tra maiuscole e min
Unicode_giapponese_cs_AI_KS_WS	Unicode giapponese, con distinzione tra maiuscole e min
Unicode_giapponese_cs_AI_WS	Unicode giapponese, con distinzione tra maiuscole e min

Japanese_Unicode_cs_AS	Unicode giapponese, con distinzione tra maiuscole e min
Unicode_giapponese_cs_as_KS	Unicode giapponese, con distinzione tra maiuscole e min
Unicode_giapponese_cs_as_KS_WS	Unicode giapponese, con distinzione tra maiuscole e min
Unicode_giapponese_cs_as_WS	Unicode giapponese, con distinzione tra maiuscole e min
Japanese_XJIS_100_bin	Japanese-XJIS-100, ordinamento binario
Giapponese_XJIS_100_bin2	Japanese-XJIS-100, tipo di confronto dei punti in codice b
Giapponese_XJIS_100_CI_AI	Japanese-XJIS-100, senza distinzione tra maiuscole e mi
Giapponese_XJIS_100_CI_AI_KS	Japanese-XJIS-100, senza distinzione tra maiuscole e mi larghezza
Giapponese_XJIS_100_CI_AI_KS_SC	Japanese-XJIS-100, senza distinzione tra maiuscole e mi caratteri supplementari
Giapponese_XJIS_100_CI_AI_KS_SC_UTF8	Japanese-XJIS-100, senza distinzione tra maiuscole e mi
Giapponese_XJIS_100_CI_AI_KS_WS	Japanese-XJIS-100, senza distinzione tra maiuscole e mi
Giapponese_XJIS_100_CI_AI_KS_WS_SC	Japanese-XJIS-100, senza distinzione tra maiuscole e mi caratteri supplementari
Giapponese_XJIS_100_CI_AI_KS_WS_SC_UTF8	Japanese-XJIS-100, senza distinzione tra maiuscole e mi tari, UTF8
Giapponese_XJIS_100_CI_AI_SC	Japanese-XJIS-100, senza distinzione tra maiuscole e mi caratteri supplementari
Giapponese_XJIS_100_CI_AI_SC_UTF8	Japanese-XJIS-100, senza distinzione tra maiuscole e mi supplementari, UTF8
Giapponese_XJIS_100_CI_AI_WS	Japanese-XJIS-100, senza distinzione tra maiuscole e mi
Giapponese_XJIS_100_CI_AI_WS_SC	Japanese-XJIS-100, senza distinzione tra maiuscole e mi caratteri supplementari

Giapponese_XJIS_100_CI_AI_WS_SC_UTF8	Japanese-XJIS-100, senza distinzione tra maiuscole e mi supplementari, UTF8
Giapponese_XJIS_100_CI_AS	Japanese-XJIS-100, senza distinzione tra maiuscole e mi
Giapponese_XJIS_100_CI_AS_KS	Japanese-XJIS-100, senza distinzione tra maiuscole e mi
Giapponese_XJIS_100_CI_AS_KS_SC	Japanese-XJIS-100, senza distinzione tra maiuscole e mi
Giapponese_XJIS_100_CI_AS_KS_SC_UTF8	Japanese-XJIS-100, senza distinzione tra maiuscole e mi
Giapponese_XJIS_100_CI_AS_KS_WS	Japanese-XJIS-100, senza distinzione tra maiuscole e mi
Giapponese_XJIS_100_CI_AS_KS_WS_SC	Japanese-XJIS-100, senza distinzione tra maiuscole e mi
Giapponese_XJIS_100_CI_AS_KS_WS_SC_UTF8	Japanese-XJIS-100, senza distinzione tra maiuscole e mi UTF8
Giapponese_XJIS_100_CI_AS_SC	Japanese-XJIS-100, senza distinzione tra maiuscole e mi caratteri supplementari
Giapponese_XJIS_100_CI_AS_SC_UTF8	Japanese-XJIS-100, senza distinzione tra maiuscole e mi supplementari, UTF8
Giapponese_XJIS_100_CI_AS_WS	Japanese-XJIS-100, senza distinzione tra maiuscole e mi
Giapponese_XJIS_100_CI_AS_WS_SC	Japanese-XJIS-100, senza distinzione tra maiuscole e mi supplementari
Giapponese_XJIS_100_CI_AS_WS_SC_UTF8	Japanese-XJIS-100, senza distinzione tra maiuscole e mi supplementari, UTF8
Giapponese_XJIS_100_CS_AI	Japanese-XJIS-100, con distinzione tra maiuscole e minu
Giapponese_XJIS_100_CS_AI_KS	Japanese-XJIS-100, con distinzione tra maiuscole e minu
Giapponese_XJIS_100_CS_AI_KS_SC	Japanese-XJIS-100, con distinzione tra maiuscole e minu
Giapponese_XJIS_100_CS_AI_KS_SC_UTF8	Japanese-XJIS-100, con distinzione tra maiuscole e minu
Giapponese_XJIS_100_CS_AI_KS_WS	Japanese-XJIS-100, con distinzione tra maiuscole e minu

Giapponese_XJIS_100_CS_AI_KS_WS_SC	Japanese-XJIS-100, con distinzione tra maiuscole e minu
Giapponese_XJIS_100_CS_AI_KS_SC_UTF8	Japanese-XJIS-100, con distinzione tra maiuscole e minu
Giapponese_XJIS_100_CS_AI_SC	Japanese-XJIS-100, con distinzione tra maiuscole e minu caratteri supplementari
Giapponese_XJIS_100_CS_AI_SC_UTF8	Japanese-XJIS-100, con distinzione tra maiuscole e minu supplementari, UTF8
Giapponese_XJIS_100_CS_AI_WS	Japanese-XJIS-100, con distinzione tra maiuscole e minu
Giapponese_XJIS_100_CS_AI_WS_SC	Japanese-XJIS-100, con distinzione tra maiuscole e minu supplementari
Giapponese_XJIS_100_CS_AI_WS_SC_UTF8	Japanese-XJIS-100, con distinzione tra maiuscole e minu supplementari, UTF8
Giapponese_XJIS_100_CS_AS	Japanese-XJIS-100, con distinzione tra maiuscole e minu
Giapponese_XJIS_100_CS_AS_KS	Japanese-XJIS-100, con distinzione tra maiuscole e minu
Giapponese_XJIS_100_CS_AS_KS_SC	Japanese-XJIS-100, con distinzione tra maiuscole e minu
Giapponese_XJIS_100_CS_AS_KS_SC_UTF8	Japanese-XJIS-100, con distinzione tra maiuscole e minu
Giapponese_XJIS_100_CS_AS_KS_WS	Japanese-XJIS-100, con distinzione tra maiuscole e minu
Giapponese_XJIS_100_CS_AS_KS_WS_SC	Japanese-XJIS-100, con distinzione tra maiuscole e minu
Giapponese_XJIS_100_CS_AS_KS_SC_UTF8	Japanese-XJIS-100, con distinzione tra maiuscole e minu
Giapponese_XJIS_100_CS_AS_SC	Japanese-XJIS-100, con distinzione tra maiuscole e minu
Giapponese_XJIS_100_CS_AS_SC_UTF8	Japanese-XJIS-100, con distinzione tra maiuscole e minu supplementari, UTF8
Giapponese_XJIS_100_CS_AS_WS	Japanese-XJIS-100, con distinzione tra maiuscole e minu
Giapponese_XJIS_100_CS_AS_WS_SC	Japanese-XJIS-100, con distinzione tra maiuscole e minu supplementari

XJIS_100_CS_AS_WS_SC_UTF8 in giapponese_XJIS_100_CS_AS_WS_SC_UTF8	Japanese-XJIS-100, con distinzione tra maiuscole e minu supplementari, UTF8
Giapponese_XJIS_140_bin	Japanese-XJIS-140, ordinamento binario
Giapponese_XJIS_140_bin2	Japanese-XJIS-140, tipo di confronto dei punti in codice b
Giapponese_XJIS_140_CI_AI	Japanese-XJIS-140, senza distinzione tra maiuscole e mi minuscole, senza distinzione tra maiuscole e minuscole, i
Giapponese_XJIS_140_CI_AI_KS	Japanese-XJIS-140, senza distinzione tra maiuscole e mi minuscole, senza distinzione tra maiuscole e minuscole, s
Giapponese_XJIS_140_CI_AI_KS_UTF8	Japanese-XJIS-140, senza distinzione tra maiuscole e mi supplementari, selettore di variazione insensibile, UTF8
Giapponese_XJIS_140_CI_AI_KS_VSS	Japanese-XJIS-140, senza distinzione tra maiuscole e mi minuscole, senza distinzione tra accenti, sensibile al selet
Giapponese_XJIS_140_CI_AI_KS_VSS_UTF8	Japanese-XJIS-140, senza distinzione tra maiuscole e mi minuscole, insensibile al selettore di variazione, UTF8
Giapponese_XJIS_140_CI_AI_KS_WS	Japanese-XJIS-140, senza distinzione tra maiuscole e mi minuscole, senza distinzione tra maiuscole e minuscole, s
Giapponese_XJIS_140_CI_AI_KS_WS_UTF8	Japanese-XJIS-140, senza distinzione tra maiuscole e mi tari, selettore di variazione insensibile, UTF8
Giapponese_XJIS_140_CI_AI_KS_WS_VSS	Japanese-XJIS-140, senza distinzione tra maiuscole e mi supplementari, sensibile al selettore di variazione
Giapponese_XJIS_140_CI_AI_KS_WS_VSS_UTF8	Japanese-XJIS-140, senza distinzione tra maiuscole e mi minuscole, senza distinzione tra accenti, UTF8
Giapponese_XJIS_140_CI_AI_UTF8	Japanese-XJIS-140, senza distinzione tra maiuscole e mi supplementari, selettore di variazione insensibile, UTF8
Giapponese_XJIS_140_CI_AI_VSS	Japanese-XJIS-140, senza distinzione tra maiuscole e mi minuscole, senza distinzione tra accenti, insensibile al sel

Giapponese_XJIS_140_CI_AI_VSS_UTF8	Japanese-XJIS-140, senza distinzione tra maiuscole e mi supplementari, sensibile al selettore di variazione, UTF8
Giapponese_XJIS_140_CI_AI_WS	Japanese-XJIS-140, senza distinzione tra maiuscole e mi minuscole, senza distinzione tra maiuscole e minuscole, s
Giapponese_XJIS_140_CI_AI_WS_UTF8	Japanese-XJIS-140, senza distinzione tra maiuscole e mi supplementari, selettore di variazione insensibile, UTF8
Giapponese_XJIS_140_CI_AI_WS_VSS	Japanese-XJIS-140, senza distinzione tra maiuscole e mi supplementari, sensibile al selettore di variazione
Giapponese_XJIS_140_CI_AI_WS_VSS_UTF8	Japanese-XJIS-140, senza distinzione tra maiuscole e mi supplementari, sensibile al selettore di variazione, UTF8
Japanese_XJIS_140_CI_AS	Japanese-XJIS-140, senza distinzione tra maiuscole e mi distinzione della larghezza, caratteri supplementari, senza
Giapponese_XJIS_140_CI_AS_KS	Japanese-XJIS-140, senza distinzione tra maiuscole e mi selettore di variazione insensibile al selettore di variazione
Giapponese_XJIS_140_CI_AS_KS_UTF8	Japanese-XJIS-140, senza distinzione tra maiuscole e mi supplementari, selettore di variazione insensibile, UTF8
Japanese_XJIS_140_CI_AS_KS_VSS	Japanese-XJIS-140, senza distinzione tra maiuscole e mi ne della larghezza, caratteri supplementari, con distinzion
Giapponese_XJIS_140_CI_AS_KS_VSS_UTF 8	Japanese-XJIS-140, senza distinzione tra maiuscole e mi supplementari, sensibile al selettore di variazione, UTF8
Giapponese_XJIS_140_CI_AS_KS_WS	Japanese-XJIS-140, senza distinzione tra maiuscole e mi di variazione insensibile
XJIS_140_CI_AS_KS_WS_UTF8 in giappones e	Japanese-XJIS-140, senza distinzione tra maiuscole e mi tari, selettore di variazione insensibile, UTF8
Giapponese_XJIS_140_CI_AS_KS_WS_VSS	Japanese-XJIS-140, senza distinzione tra maiuscole e mi sensibile al selettore di variazione

Giapponese_XJIS_140_CI_AS_KS_WS_VSS_UTF8	Japanese-XJIS-140, senza distinzione tra maiuscole e mi tari, sensibile al selettore di variazione, UTF8
Giapponese_XJIS_140_CI_AS_UTF8	Japanese-XJIS-140, senza distinzione tra maiuscole e mi caratteri supplementari, selettore di variazione insensibile
Japanese_XJIS_140_CI_AS_VSS	Japanese-XJIS-140, senza distinzione tra maiuscole e mi distinzione della larghezza, caratteri supplementari, con d
Giapponese_XJIS_140_CI_AS_VSS_UTF8	Japanese-XJIS-140, senza distinzione tra maiuscole e mi supplementari, sensibile al selettore di variazione, UTF8
Giapponese_XJIS_140_CI_AS_WS	Japanese-XJIS-140, senza distinzione tra maiuscole e mi supplementari, selettore di variazione insensibile al selett
XJIS_140_CI_AS_WS_UTF8 in giapponese	Japanese-XJIS-140, senza distinzione tra maiuscole e mi supplementari, selettore di variazione insensibile, UTF8
Giapponese_XJIS_140_CI_AS_WS_VSS	Japanese-XJIS-140, senza distinzione tra maiuscole e mi supplementari, sensibile al selettore di variazione
Giapponese_XJIS_140_CI_AS_WS_VSS_UTF8	Japanese-XJIS-140, senza distinzione tra maiuscole e mi supplementari, sensibile al selettore di variazione, UTF8
Giapponese_XJIS_140_CS_AI	Japanese-XJIS-140, con distinzione tra maiuscole e minu supplementari, selettore di variazione insensibile al selett
Giapponese_XJIS_140_CS_AI_KS	Japanese-XJIS-140, con distinzione tra maiuscole e minu di variazione insensibile
Giapponese_XJIS_140_CS_AI_KS_UTF8	Japanese-XJIS-140, con distinzione tra maiuscole e minu di variazione insensibile, UTF8
Giapponese_XJIS_140_CS_AI_KS_VSS	Japanese-XJIS-140, con distinzione tra maiuscole e minu al selettore di variazione
Giapponese_XJIS_140_CS_AI_KS_VSS_UTF8	Japanese-XJIS-140, con distinzione tra maiuscole e minu tari, sensibile al selettore di variazione, UTF8

Giapponese_XJIS_140_CS_AI_KS_WS	Japanese-XJIS-140, con distinzione tra maiuscole e minu variazione insensibile
Giapponese_XJIS_140_CS_AI_KS_WS_UTF8	Japanese-XJIS-140, con distinzione tra maiuscole e minu variazione insensibile, UTF8
Giapponese_XJIS_140_CS_AI_KS_WS_VSS	Japanese-XJIS-140, con distinzione tra maiuscole e minu selettore di variazione
Giapponese_XJIS_140_CS_AI_KS_WS_VSS_UTF8	Japanese-XJIS-140, con distinzione tra maiuscole e minu sensibile al selettore di variazione, UTF8
Giapponese_XJIS_140_CS_AI_UTF8	Japanese-XJIS-140, con distinzione tra maiuscole e minu supplementari, selettore di variazione insensibile, UTF8
Giapponese_XJIS_140_CS_AI_VSS	Japanese-XJIS-140, con distinzione tra maiuscole e minu supplementari, sensibile al selettore di variazione
Giapponese_XJIS_140_CS_AI_VSS_UTF8	Japanese-XJIS-140, con distinzione tra maiuscole e minu supplementari, sensibile al selettore di variazione, UTF8
Giapponese_XJIS_140_CS_AI_WS	Japanese-XJIS-140, con distinzione tra maiuscole e minu tari, selettore di variazione insensibile al selettore di varia
Giapponese_XJIS_140_CS_AI_WS_UTF8	Japanese-XJIS-140, con distinzione tra maiuscole e minu supplementari, selettore di variazione insensibile, UTF8
Giapponese_XJIS_140_CS_AI_WS_VSS	Japanese-XJIS-140, con distinzione tra maiuscole e minu tari, sensibile al selettore di variazione
Giapponese_XJIS_140_CS_AI_WS_VSS_UTF8	Japanese-XJIS-140, con distinzione tra maiuscole e minu supplementari, sensibile al selettore di variazione, UTF8
Giapponese_XJIS_140_CS_AS	Japanese-XJIS-140, con distinzione tra maiuscole e minu supplementari, selettore di variazione insensibile al selett
Giapponese_XJIS_140_CS_AS_KS	Japanese-XJIS-140, con distinzione tra maiuscole e minu variazione insensibile al selettore di variazione

Giapponese_XJIS_140_CS_AS_KS_UTF8	Japanese-XJIS-140, con distinzione tra maiuscole e minu e insensibile, UTF8
Giapponese_XJIS_140_CS_AS_KS_VSS	Japanese-XJIS-140, con distinzione tra maiuscole e minu sensibile al selettore di variazione
Giapponese_XJIS_140_CS_AS_KS_VSS_UTF 8	Japanese-XJIS-140, con distinzione tra maiuscole e minu supplementari, sensibile al selettore di variazione, UTF8
Japanese_XJIS_140_CS_AS_KS_WS	XJIS-140 in giapponese, con distinzione tra maiuscole e r variazione insensibile
Giapponese_XJIS_140_CS_AS_K S_WS_UTF8	Japanese-XJIS-140, con distinzione tra maiuscole e minu variazione insensibile, UTF8
Giapponese_XJIS_140_CS_AS_KS_WS_VSS	Japanese-XJIS-140, con distinzione tra maiuscole e minu tari, sensibile al selettore di variazione
Giapponese_XJIS_140_CS_AS_K S_WS_VSS_UTF8	Japanese-XJIS-140, con distinzione tra maiuscole e minu tari, sensibile al selettore di variazione, UTF8
Giapponese_XJIS_140_CS_AS_UTF8	Japanese-XJIS-140, con distinzione tra maiuscole e minu supplementari, selettore di variazione insensibile, UTF8
Giapponese_XJIS_140_CS_AS_VSS	Japanese-XJIS-140, con distinzione tra maiuscole e minu supplementari, sensibile al selettore di variazione
Giapponese_XJIS_140_CS_AS_VSS_UTF8	Japanese-XJIS-140, con distinzione tra maiuscole e minu supplementari, sensibile al selettore di variazione, UTF8
Giapponese_XJIS_140_CS_AS_WS	Japanese-XJIS-140, con distinzione tra maiuscole e minu tari, selettore di variazione insensibile al selettore di varia
Giapponese_XJIS_140_CS_AS_WS_UTF8	Japanese-XJIS-140, con distinzione tra maiuscole e minu selettore di variazione insensibile, UTF8
Giapponese_XJIS_140_CS_AS_WS_VSS	Japanese-XJIS-140, con distinzione tra maiuscole e minu supplementari, sensibile al selettore di variazione

Giapponese_XJIS_140_CS_AS_WS_VSS_UTF8	Japanese-XJIS-140, con distinzione tra maiuscole e minu tari, sensibile al selettore di variazione, UTF8
Korean_Wansung_CI_AS	Korean-Wansung, case-insensitive, accent-sensitive, kan
Latin1_General_100_BIN	Latin1-General-100, ordinamento binario
Latin1_General_100_BIN2	Latin1-General-100, ordinamento binario basato sul confr
Latin1_General_100_BIN2_UTF8	Latin1-General-100, ordinamento di confronto dei punti in
Latin1_General_100_CI_AS	Latin1-General-100, case-insensitive, accent-sensitive, ka
Latin1_General_100_CI_AS_SC_UTF8	Latin1-General-100, senza distinzione tra maiuscole e mi supplementari, UTF8
Latin1_General_BIN	Latin1-General, binary sort
Latin1_General_BIN2	Latin1-General, ordinamento binario basato sul confronto
Latin1_General_CI_AI	Latin1-General, case-insensitive, accent-insensitive, kana
Latin1_General_CI_AS	Latin1-General, case-insensitive, accent-sensitive, kanaty
Latin1_General_CI_AS_KS	Latin1-General, non sensibile al maiuscolo/minuscolo, se
Latin1_General_CS_AS	Latin1-General, con distinzione tra maiuscole e minuscole
Modern_Spanish_CI_AS	Modern-Spanish, case-insensitive, accent-sensitive, kana
SQL_1xCompat_CP850_CI_AS	Latin1-General, non sensibile al maiuscolo/minuscolo, se Unicode, SQL Server Ordinamento 49 su codepage 850 p
SQL_Latin1_General_CP1_CI_AI	Latin1-General, case-insensitive, accent-insensitive, kana Page 1252 per dati non-Unicode
SQL_Latin1_General_CP1_CI_AS	Latin1-General, case-insensitive, accent-sensitive, kanaty 1252 per dati non-Unicode
SQL_Latin1_General_CP1_CS_AS	Latin1-General, case-sensitive, accent-sensitive, kanatyp per dati non-Unicode

SQL_Latin1_General_CP1250_CI_AS	Latin1-General, senza distinzione tra maiuscole e minusc ordinamento 82 di SQL Server nella pagina codici 1250 p
SQL_Latin1_General_CP1250_cs_AS	Latin1-General, sensibile alle maiuscole e minuscole, agli SQL Server nella pagina codici 1250 per dati non Unicode
SQL_Latin1_General_CP1251_CI_AS	Latin1-General, senza distinzione tra maiuscole e minusc ordinamento 106 di SQL Server nella pagina codici 1251
SQL_Latin1_General_CP1251_CS_AS	Latin1-General, sensibile alle maiuscole e minuscole, agli di SQL Server nella pagina codici 1251 per dati non Unico
SQL_Latin1_General_CP1253_CI_AI	Latin1-General, senza distinzione tra maiuscole e minusc ordinamento 124 di SQL Server nella pagina codici 1253
SQL_Latin1_General_CP1253_CI_AS	Latin1-General, senza distinzione tra maiuscole e minusc ordine di ordinamento 114 di SQL Server nella pagina co
SQL_Latin1_General_CP1253_cs_AS	Latin1-General, sensibile alle maiuscole e minuscole, agli di SQL Server nella pagina codici 1253 per dati non Unico
SQL_Latin1_General_CP1254_CI_AS	Turco, senza distinzione tra maiuscole e minuscole, sens Server Sort Order 130 nella pagina codici 1254 per dati n
SQL_Latin1_General_CP1254_CS_AS	Turco, con distinzione tra maiuscole e minuscole, sensibil Sort Order 129 nella pagina codici 1254 per dati non Unic
SQL_Latin1_General_CP1255_CI_AS	Latin1-General, senza distinzione tra maiuscole e minusc ordinamento 138 di SQL Server nella pagina codici 1255
SQL_Latin1_General_CP1255_CS_AS	Latin1-General, sensibile alle maiuscole e minuscole, agli di SQL Server nella pagina codici 1255 per dati non Unico
SQL_Latin1_General_CP1256_CI_AS	Latin1-General, non sensibile al maiuscolo/minuscolo, se Unicode, SQL Server Ordinamento 146 su codepage 125
SQL_Latin1_General_CP1256_CS_AS	Latin1-General, sensibile alle maiuscole e minuscole, agli Order 145 nella pagina codici 1256 per dati non Unicode

SQL_Latin1_General_CP1257_CI_AS	Latin1-General, senza distinzione tra maiuscole e minusc ordinamento 154 di SQL Server nella pagina codici 1257
SQL_Latin1_General_CP1257_CS_AS	Latin1-General, sensibile alle maiuscole e minuscole, agli di SQL Server nella pagina codici 1257 per dati non Unico
SQL_Latin1_General_CP437_bin	Latin1-General, ordinamento binario per dati Unicode, SQ
SQL_Latin1_General_CP437_bin2	Latin1-General, ordinamento di confronto dei punti di codi Unicode
SQL_Latin1_General_CP437_CI_AI	Latin1-General, case-insensitive, accent-insensitive, kana Page 437 per dati non-Unicode
SQL_Latin1_General_CP437_CI_AS	Latin1-General, senza distinzione tra maiuscole e minusc ordinamento 32 di SQL Server nella pagina codici 437 pe
SQL_Latin1_General_CP437_CS_AS	Latin1-General, sensibile alle maiuscole e minuscole, agli SQL Server nella pagina codici 437 per dati non Unicode
SQL_Latin1_General_CP850_BIN	Latin1-General, ordinamento binario per dati Unicode, SQ
SQL_Latin1_General_CP850_BIN2	Latin1-General, binary code point comparison sort per da
SQL_Latin1_General_CP850_CI_AI	Latin1-General, senza distinzione tra maiuscole e minusc della larghezza per dati Unicode, SQL Server Ordinamen
SQL_Latin1_General_CP850_CI_AS	Latin1-General, case-insensitive, accent-sensitive, kanaty 850 per dati non-Unicode
SQL_Latin1_General_CP850_CS_AS	Latin1-General, sensibile alle maiuscole e minuscole, agli ordinamento 41 di SQL Server nella pagina codici 850 pe
SQL_Latin1_General_Pref_CP1_CI_AS	Latin1-General, senza distinzione tra maiuscole e minusc ordinamento 53 di SQL Server nella pagina codici 1252 p
SQL_Latin1_General_Pref_CP437_CI_AS	Latin1-General, senza distinzione tra maiuscole e minusc ordinamento 33 di SQL Server nella pagina codici 437 pe

SQL_Latin1_General_Pref_CP850_CI_AS	Latin1-General, senza distinzione tra maiuscole e minusc ordine di ordinamento 43 di SQL Server nella pagina codi
Thai_CI_AS	Thai, case-insensitive, accent-sensitive, kanatype-insensi

Fuso orario locale per le istanze database di RDS Custom for SQL Server

Il fuso orario di un'istanza database RDS Custom for SQL Server è impostato per default. L'impostazione predefinita corrente è Universal Coordinated Time (UTC). Ora puoi invece impostare il fuso orario delle istanze database su un fuso orario locale, per farlo corrispondere a quello delle applicazioni.

Puoi impostare il fuso orario quando si crea prima l'istanza database. Puoi creare la tua istanza DB utilizzando l'azione AWS Management ConsoleCreateDBInstance dell'API Amazon RDS o il comando. AWS CLI create-db-instance

Se l'istanza database fa parte di un'implementazione Multi-AZ, quando si esegue il failover, il fuso orario rimane quello locale impostato.

Quando richiedi un point-in-time ripristino, specifichi l'ora in cui eseguire il ripristino. L'ora viene visualizzata nel fuso orario locale. Per ulteriori informazioni, consulta Ripristino a un'ora specifica per un'istanza database.

Le seguenti limitazioni riguardano l'impostazione del fuso orario locale sull'istanza database:

- È possibile configurare il fuso orario per un'istanza database durante la creazione dell'istanza, ma non è possibile modificare il fuso orario di un'istanza database RDS Custom for SQL Server esistente.
- Se il fuso orario viene modificato per un'istanza database RDS Custom for SQL Server esistente, RDS Custom modifica lo stato dell'istanza database in `unsupported-configuration` e invia notifiche di eventi.
- Non è possibile ripristinare uno snapshot da un'istanza database in un fuso orario a un'istanza database in un fuso orario diverso.
- Consigliamo vivamente di non ripristinare un file di backup da un fuso orario a un fuso orario diverso. Se ripristini un file di backup da un fuso orario in un fuso orario diverso, devi controllare le query e le applicazioni per verificare gli effetti del cambiamento di fuso orario. Per ulteriori informazioni, consulta Importazione ed esportazione di database SQL Server mediante backup e ripristino nativi.

Fusi orari supportati

Puoi impostare il fuso orario locale su uno dei valori elencati nella tabella di seguito.

Fusi orari supportati per RDS Custom for SQL Server

Time zone (Fuso orario)	Offset temporale standard	Descrizione	Note
Orario standard Afghanistan	(UTC+04:30)	Kabul	Questo fuso orario non osserva l'ora legale.
Orario standard Alaska	(UTC–09:00)	Alaska	
Orario standard delle Isole Aleutine	(UTC–10:00)	Isole Aleutine	
Orario standard Altai	(UTC+07:00)	Barnaul, Gorno-Altaysk	
Orario standard arabo	(UTC+03:00)	Kuwait, Riyad	Questo fuso orario non osserva l'ora legale.
Orario standard Arabia	(UTC+04:00)	Abu Dhabi, Mascate	
Orario standard arabo	(UTC+03:00)	Baghdad	Questo fuso orario non osserva l'ora legale.
Orario standard Argentina	(UTC–03:00)	Città di Buenos Aires	Questo fuso orario non osserva l'ora legale.
Orario standard di Astrakhan	(UTC+04:00)	Astrakhan, Ulyanovsk	
Orario standard Atlantico	(UTC–04:00)	Orario Atlantico (Canada)	

Time zone (Fuso orario)	Offset temporale standard	Descrizione	Note
Orario standard Australia centrale	(UTC+09:30)	Darwin	Questo fuso orario non osserva l'ora legale.
Orario standard Australia centrale	(UTC+ 08:45)	Eucla	
Orario standard Australia orientale	(UTC+10:00)	Canberra, Melbourne, Sydney	
Orario standard dell'Azerbaigian	(UTC+04:00)	Baku	
Orario standard delle Azzorre	(UTC–01:00)	Azzorre	
Orario standard di Bahia	(UTC–03:00)	Salvador	
Orario standard del Bangladesh	(UTC+06:00)	Dacca	Questo fuso orario non osserva l'ora legale.
Orario standard Bielorussia	(UTC+03:00)	Minsk	Questo fuso orario non osserva l'ora legale.
Orario standard di Bougainville	(UTC+11:00)	Isola di Bougainville	
Orario standard Canada centrale	(UTC–06:00)	Saskatchewan	Questo fuso orario non osserva l'ora legale.
Orario standard Capo Verde	(UTC–01:00)	Capo Verde II.	Questo fuso orario non osserva l'ora legale.

Time zone (Fuso orario)	Offset temporale standard	Descrizione	Note
Orario standard del Caucaso	(UTC+04:00)	Yerevan	
Cen. Ora standard Australia	(UTC+09:30)	Adelaide	
Orario standard America centrale	(UTC–06:00)	America centrale	Questo fuso orario non osserva l'ora legale.
Orario standard Asia centrale	(UTC+06:00)	Astana	Questo fuso orario non osserva l'ora legale.
Orario standard Brasile centrale	(UTC–04:00)	Cuiaba	
Orario standard Europa centrale	(UTC+01:00)	Belgrado, Bratislava, Budapest, Lubiana, Praga	
Orario standard Europeo centrale	(UTC+01:00)	Sarajevo, Skopje, Varsavia, Zagabria	
Orario standard Pacifico centrale	(UTC+11:00)	Isole Salomone, Nuova Caledonia	Questo fuso orario non osserva l'ora legale.
Orario standard centrale	(UTC–06:00)	Orario Centrale (Stati Uniti e Canada)	
Orario standard centrale (Messico)	(UTC–06:00)	Guadalajara, Città del Messico, Monterrey	
Orario standard Isole Chatham	(UTC+ 12:45)	Isole Chatham	

Time zone (Fuso orario)	Offset temporale standard	Descrizione	Note
Orario standard Cina	(UTC+08:00)	Pechino, Chongqing, Hong Kong, Urumqi	Questo fuso orario non osserva l'ora legale.
Orario standard Cuba	(UTC–05:00)	L'Avana	
Orario standard della Dateline	(UTC–12:00)	Linea di data internazionale Ovest	Questo fuso orario non osserva l'ora legale.
Ora standard Africa orientale	(UTC+03:00)	Nairobi	Questo fuso orario non osserva l'ora legale.
Ora standard Australia orientale	(UTC+10:00)	Brisbane	Questo fuso orario non osserva l'ora legale.
Ora standard Europa orientale	(UTC+02:00)	Chisinau	
Ora standard Sud America orientale	(UTC–03:00)	Brasilia	
Orario standard Isola di Pasqua	(UTC–06:00)	Isola di Pasqua	
Orario standard orientale	(UTC–05:00)	Orario orientale (Stati Uniti e Canada)	
Orario standard orientale (Messico)	(UTC–05:00)	Chetumal	
Orario standard Egitto	(UTC+02:00)	Il Cairo	
Orario standard Ekaterinburg	(UTC+ 05:00)	Ekaterinburg	

Time zone (Fuso orario)	Offset temporale standard	Descrizione	Note
Orario standard Fiji	(UTC+12:00)	Figi	
Orario standard FLE	(UTC+02:00)	Helsinki, Kiev, Riga, Sofia, Tallinn, Vilnius	
Orario standard Georgia	(UTC+04:00)	Tbilisi	Questo fuso orario non osserva l'ora legale.
Orario standard GMT	(UTC)	Dublino, Edimburgo, Lisbona, Londra	Questo fuso orario non è lo stesso di Greenwich Mean Time. Questo fuso orario osserva l'ora legale.
Orario standard Groenlandia	(UTC–03:00)	Groenlandia	
Orario standard Greenwich	(UTC)	Monrovia, Reykjavik	Questo fuso orario non osserva l'ora legale.
Orario standard GTB	(UTC+02:00)	Atene, Bucarest	
Orario standard di Haiti	(UTC–05:00)	Haiti	
Orario standard Hawaii	(UTC–10:00)	Hawaii	
Orario standard India	(UTC+05:30)	Chennai, Kolkata, Mumbai, Nuova Delhi	Questo fuso orario non osserva l'ora legale.
Orario standard Iran	(UTC+ 03:30)	Teheran	
Orario standard di Israele	(UTC+02:00)	Gerusalemme	

Time zone (Fuso orario)	Offset temporale standard	Descrizione	Note
Orario standard Giordania	(UTC+02:00)	Amman	
Orario standard di Kaliningrad	(UTC+02:00)	Kaliningrad	
Orario standard Kamchatka	(UTC+12:00)	Petropavlovsk-Kamchatsky – Vecchio	
Orario standard Corea	(UTC+09:00)	Seoul	Questo fuso orario non osserva l'ora legale.
Orario standard Libia	(UTC+02:00)	Tripoli	
Ora standard Isole Line	(UTC+ 14:00)	Isola di Kiritimati	
Orario standard Lord Howe	(UTC+ 10:30)	Isola di Lord Howe	
Orario standard Magadan	(UTC+11:00)	Magadan	Questo fuso orario non osserva l'ora legale.
Orario standard Magallanes	(UTC–03:00)	Punta Arenas	
Orario standard delle Marchesi	(UTC–09:30)	Isole Marchesi	
Orario standard delle Mauritius	(UTC+04:00)	Port Louis	Questo fuso orario non osserva l'ora legale.
Orario standard Medio Oriente	(UTC+02:00)	Beirut	
Orario standard di Montevideo	(UTC–03:00)	Montevideo	
Orario standard del Marocco	(UTC+01:00)	Casablanca	

Time zone (Fuso orario)	Offset temporale standard	Descrizione	Note
Orario standard di montagna	(UTC–07:00)	Orario di montagna (Stati Uniti e Canada)	
Orario standard di montagna (Messico)	(UTC–07:00)	Chihuahua, La Paz, Mazatlan	
Orario standard del Myanmar	(UTC+ 06:30)	Yangon (Rangoon)	Questo fuso orario non osserva l'ora legale.
Ora standard Asia centrale settentrionale	(UTC+07:00)	Novosibirsk	
Orario standard della Namibia	(UTC+02:00)	Windhoek	
Orario standard del Nepal	(UTC+ 05:45)	Katmandu	Questo fuso orario non osserva l'ora legale.
Orario standard Nuova Zelanda	(UTC+12:00)	Auckland, Wellington	
Orario standard Terranova	(UTC–03:30)	Terranova	
Orario standard di Norfolk	(UTC+11:00)	Isola di Norfolk	
Orario standard dell'Asia nord-orientale	(UTC+08:00)	Irkutsk	
Orario standard dell'Asia settentrionale	(UTC+07:00)	Krasnoyarsk	
Orario standard Corea del Nord	(UTC+09:00)	Pyongyang	
Orario standard di Omsk	(UTC+06:00)	Omsk	

Time zone (Fuso orario)	Offset temporale standard	Descrizione	Note
Orario standard Pacifico SA	(UTC–03:00)	Santiago	
Orario standard Pacifico	(UTC–08:00)	Orario del Pacifico (Stati Uniti e Canada)	
Orario standard Pacifico (Messico)	(UTC–08:00)	Bassa California	
Orario standard del Pakistan	(UTC+ 05:00)	Islamabad, Karachi	Questo fuso orario non osserva l'ora legale.
Orario standard Paraguay	(UTC–04:00)	Asuncion	
Orario standard Romance	(UTC+01:00)	Bruxelles, Copenaghen, Madrid, Parigi	
Russia Fuso orario 10	(UTC+11:00)	Chokurdakh	
Russia Fuso orario 11	(UTC+12:00)	Anadyr, Petropavl ovsk-Kamchatsky	
Russia Fuso orario 3	(UTC+04:00)	Izhevsk, Samara	
Orario standard Russia	(UTC+03:00)	Mosca, San Pietroburgo, Volgograd	Questo fuso orario non osserva l'ora legale.
Orario standard SA orientali	(UTC–03:00)	Cayenna, Fortaleza	Questo fuso orario non osserva l'ora legale.

Time zone (Fuso orario)	Offset temporale standard	Descrizione	Note
Orario standard Pacifico SA	(UTC–05:00)	Bogotà, Lima, Quito, Rio Branco	Questo fuso orario non osserva l'ora legale.
Orario standard SA occidentali	(UTC–04:00)	Georgetown, La Paz, Manaus, San Juan	Questo fuso orario non osserva l'ora legale.
Orario standard Saint Pierre	(UTC–03:00)	Saint Pierre e Miquelon	
Orario standard Sakhalin	(UTC+11:00)	Sakhalin	
Orario standard Samoa	(UTC+ 13:00)	Samoa	
Orario standard di Sao Tomé	(UTC+01:00)	São Tomé	
Orario standard di Saratov	(UTC+04:00)	Saratov	
Orario standard Asia sud-orientale	(UTC+07:00)	Bangkok, Hanoi, Giacarta	Questo fuso orario non osserva l'ora legale.
Orario standard Singapore	(UTC+08:00)	Kuala Lumpur, Singapore	Questo fuso orario non osserva l'ora legale.
Orario standard Africa meridionale	(UTC+02:00)	Harare, Pretoria	Questo fuso orario non osserva l'ora legale.
Ora standard dello Sri Lanka	(UTC+05:30)	Sri Jayawardenepura	Questo fuso orario non osserva l'ora legale.
Ora standard Sudan	(UTC+02:00)	Khartum	

Time zone (Fuso orario)	Offset temporale standard	Descrizione	Note
Orario standard Siria	(UTC+02:00)	Damasco	
Orario standard di Taipei	(UTC+08:00)	Taipei	Questo fuso orario non osserva l'ora legale.
Orario standard della Tasmania	(UTC+10:00)	Hobart	
Orario standard Tocantins	(UTC–03:00)	Araguaina	
Orario standard Tokyo	(UTC+09:00)	Osaka, Sapporo, Tokyo	Questo fuso orario non osserva l'ora legale.
Orario standard di Tomsk	(UTC+07:00)	Tomsk	
Orario standard di Tonga	(UTC+ 13:00)	Nuku'alofa	Questo fuso orario non osserva l'ora legale.
Orario standard di Transbaikal	(UTC+09:00)	Chita	
Orario standard della Turchia	(UTC+03:00)	Istanbul	
Orario standard di Turks e Caicos	(UTC–05:00)	Turks e Caicos	
Orario standard di Ulaanbaatar	(UTC+08:00)	Ulaanbaatar	Questo fuso orario non osserva l'ora legale.
Orario standard Stati Uniti orientali	(UTC–05:00)	Indiana (Est)	

Time zone (Fuso orario)	Offset temporale standard	Descrizione	Note
Orario standard di montagna Stati Uniti	(UTC–07:00)	Arizona	Questo fuso orario non osserva l'ora legale.
UTC	UTC	Tempo coordinato universale	Questo fuso orario non osserva l'ora legale.
UTC–02	(UTC–02:00)	Tempo coordinato universale–02	Questo fuso orario non osserva l'ora legale.
UTC–08	(UTC–08:00)	Tempo coordinato universale–08	
UTC–09	(UTC–09:00)	Tempo coordinato universale–09	
UTC–11	(UTC–11:00)	Tempo coordinato universale–11	Questo fuso orario non osserva l'ora legale.
UTC+12	(UTC+12:00)	Tempo coordinato universale+12	Questo fuso orario non osserva l'ora legale.
UTC+13	(UTC+ 13:00)	Tempo coordinato universale+13	
Orario standard del Venezuela	(UTC–04:00)	Caracas	Questo fuso orario non osserva l'ora legale.
Orario standard di Vladivostok	(UTC+10:00)	Vladivostok	
Orario standard di Volgograd	(UTC+04:00)	Volgograd	

Time zone (Fuso orario)	Offset temporale standard	Descrizione	Note
Ora standard Australia occidentale	(UTC+08:00)	Perth	Questo fuso orario non osserva l'ora legale.
Ora standard Africa centrale occidentale	(UTC+01:00)	Africa centro-oc cidentale	Questo fuso orario non osserva l'ora legale.
Ora standard Europa occidentale	(UTC+01:00)	Amsterdam, Berlino, Berna, Roma, Stoccolma, Vienna	
Ora standard Mongolia occidentale	(UTC+07:00)	Hovd	
Orario standard dell'Asia occidentale	(UTC+ 05:00)	Ashgabat, Tashkent	Questo fuso orario non osserva l'ora legale.
Orario standard della Cisgiordania	(UTC+02:00)	Gaza, Hebron	
Orario standard Pacifico occidentale	(UTC+10:00)	Guam, Port Moresby	Questo fuso orario non osserva l'ora legale.
Orario standard di Yakutsk	(UTC+09:00)	Yakutsk	

Utilizzo di una Service Master Key con RDS Custom per SQL Server

RDS Custom for SQL Server supporta l'utilizzo di una Service Master Key (SMK). RDS Custom mantiene lo stesso SMK per tutta la durata dell'istanza DB di RDS Custom for SQL Server. Mantenendo lo stesso SMK, l'istanza DB può utilizzare oggetti crittografati con SMK, come password e credenziali dei server collegati. Se si utilizza una distribuzione Multi-AZ, RDS Custom inoltre sincronizza e mantiene l'SMK tra le istanze DB primarie e secondarie.

Argomenti

Disponibilità di regioni e versioni

L'utilizzo di un SMK è supportato in tutte le regioni in cui è disponibile RDS Custom per SQL Server, per tutte le versioni di SQL Server disponibili su RDS Custom. Per ulteriori informazioni sulla disponibilità della versione e della regione di Amazon RDS con RDS Custom per SQL Server, consulta. Regioni e motori DB supportati per RDS Custom per SQL Server

Funzionalità supportate

Quando si utilizza un SMK con RDS Custom per SQL Server, sono supportate le seguenti funzionalità:

- Transparent Data Encryption (TDE)
- Crittografia a livello di colonna
- Posta elettronica database
- Server collegati
- Servizi di integrazione SQL Server (SSIS)

Uso di TDE

Un SMK consente di configurare Transparent Data Encryption (TDE), che crittografa i dati prima che vengano scritti sullo storage e decripta automaticamente i dati quando vengono letti dallo storage. A differenza di RDS per SQL Server, la configurazione di TDE su un'istanza DB RDS Custom for SQL Server non richiede l'utilizzo di gruppi di opzioni. Invece, dopo aver creato un certificato e una chiave di crittografia del database, puoi eseguire il seguente comando per attivare TDE a livello di database:

```
ALTER DATABASE [myDatabase] SET ENCRYPTION ON;
```

Per ulteriori informazioni sull'utilizzo di TDE con RDS per SQL Server, vedere. Supporto per Transparent Data Encryption in SQL Server

Per informazioni dettagliate su TDE in Microsoft SQL Server, vedere Transparent Data Encryption nella documentazione Microsoft.

Configurazione delle funzionalità

Per i passaggi dettagliati sulla configurazione delle funzionalità che utilizzano un SMK con RDS Custom for SQL Server, puoi utilizzare i seguenti post nel blog del database Amazon RDS:

- Server collegati: configurazione dei server collegati su RDS Custom for SQL Server.
- SSIS: migra i pacchetti SSIS su RDS Custom per SQL Server.
- TDE: proteggi i tuoi dati utilizzando TDE su RDS Custom for SQL Server.

Requisiti e limitazioni

Quando utilizzi un SMK con un'istanza DB RDS Custom for SQL Server, tieni presente i seguenti requisiti e limitazioni:

- Se rigeneri l'SMK sulla tua istanza DB, devi eseguire immediatamente uno snapshot del DB manuale. Se possibile, consigliamo di evitare di rigenerare l'SMK.
- È necessario conservare i backup dei certificati del server e delle password delle chiavi principali del database. Se non si eseguono i backup di questi dati, è possibile che si verifichi una perdita di dati.
- Se si configura SSIS, è necessario utilizzare un documento SSM per aggiungere l'istanza DB RDS Custom for SQL Server al dominio in caso di sostituzione di un computer o di un host di scala.
- Quando TDE o la crittografia a colonne sono abilitati, i backup del database vengono crittografati automaticamente. Quando si esegue un ripristino istantaneo o un ripristino point-in-time, l'SMK dall'istanza DB di origine verrà ripristinato per decrittografare i dati per il ripristino e verrà generato un nuovo SMK per crittografare nuovamente i dati sull'istanza ripristinata.

Per ulteriori informazioni sulle chiavi master dei servizi in Microsoft SQL Server, vedere SQL Server e le chiavi di crittografia del database nella documentazione Microsoft.

Configurazione dell'ambiente per Amazon RDS Custom per SQL Server

Prima di creare e gestire un'istanza database per l'istanza database di Amazon RDS Custom per SQL Server, assicurarsi di eseguire le seguenti attività.

Indice

> **Note**
>
> Per un step-by-step tutorial su come configurare i prerequisiti e avviare Amazon RDS Custom for SQL Server, consulta Introduzione ad Amazon RDS Custom for SQL Server utilizzando un

CloudFormation modello (configurazione di rete) ed Esplora i prerequisiti necessari per creare un'istanza Amazon RDS Custom for SQL Server.

Prerequisiti per la configurazione di RDS Custom for SQL Server

Prima di creare un'istanza database di RDS Custom per SQL Server, assicurati che l'ambiente soddisfi i requisiti descritti in questo argomento. Puoi anche utilizzare il CloudFormation modello per configurare i prerequisiti all'interno del tuo. Account AWS Per ulteriori informazioni, consulta Configurazione con AWS CloudFormation

RDS Custom for SQL Server richiede la configurazione dei seguenti prerequisiti:

- Configura le autorizzazioni AWS Identity and Access Management (IAM) richieste per la creazione dell'istanza. Si tratta dell'utente o del ruolo AWS Identity and Access Management (IAM) necessario per effettuare una `create-db-instance` richiesta a RDS.
- Configura le risorse prerequisite richieste dall'istanza DB RDS Custom for SQL Server:
  - Configura la AWS KMS chiave richiesta per la crittografia dell'istanza RDS Custom. RDS Custom richiede una chiave gestita dal cliente al momento della creazione dell'istanza per la crittografia. L'ARN, l'ID, l'alias ARN o il nome alias della chiave KMS viene passato `kms-key-id` come parametro nella richiesta di creazione dell'istanza DB personalizzata RDS.
  - Configura le autorizzazioni richieste all'interno dell'istanza DB di RDS Custom for SQL Server. RDS Custom associa un profilo di istanza all'istanza DB al momento della creazione e lo utilizza per l'automazione all'interno dell'istanza DB. Il nome del profilo dell'istanza è impostato su `custom-iam-instance-profile` nella richiesta di creazione personalizzata RDS. È possibile creare un profilo di istanza da AWS Management Console o creare manualmente il proprio profilo di istanza. Per ulteriori informazioni, consulta Creazione automatica del profilo di istanza utilizzando il AWS Management Console e Creazione manuale del ruolo IAM e del profilo dell'istanza.
  - Configurare la configurazione di rete in base ai requisiti di RDS Custom for SQL Server. Le istanze RDS Custom risiedono nelle sottoreti (configurate con il gruppo di sottoreti DB) fornite al momento della creazione dell'istanza. Queste sottoreti devono consentire alle istanze RDS Custom di comunicare con i servizi richiesti per l'automazione RDS.

> **Note**
>
> Per i requisiti sopra menzionati, assicurati che non esistano politiche di controllo del servizio (SCP) che limitino le autorizzazioni a livello di account.
> Se l'account che stai utilizzando fa parte di un'organizzazione AWS , potrebbe disporre di policy di controllo dei servizi che limitano le autorizzazioni a livello di account. Assicurati che le SCP non limitino le autorizzazioni per gli utenti e i ruoli creati utilizzando le seguenti procedure.
> Per ulteriori informazioni sulle SCP, consulta Policy di controllo dei servizi (Service Control Policies, SCP) nella Guida per l'utente di AWS Organizations . Usa il AWS CLI comando describe-organization per verificare se il tuo account fa parte di un'organizzazione. AWS
> Per ulteriori informazioni su AWS Organizations, consulta What is AWS Organizations nella AWS Organizations User Guide.

Per i requisiti generali applicabili a RDS Custom per SQL Server, vedere Requisiti generali per RDS Custom per SQL Server.

Creazione automatica del profilo di istanza utilizzando il AWS Management Console

RDS Custom richiede la creazione e la configurazione di un profilo di istanza per avviare qualsiasi istanza DB di RDS Custom for SQL Server. Utilizza il AWS Management Console per creare e allegare un nuovo profilo di istanza in un unico passaggio. Questa opzione è disponibile nella sezione RDS Custom security nelle pagine Create database, Restore snapshot e Restore to point-in-time console. Scegli Crea un nuovo profilo di istanza per fornire un suffisso per il nome del profilo di istanza. AWS Management Console crea un nuovo profilo di istanza con le autorizzazioni necessarie per le attività di automazione RDS Custom. Per creare automaticamente nuovi profili di istanza, l' AWS Management Console utente che ha effettuato l'accesso deve disporre di`iam:CreateInstanceProfile`,`iam:AddRoleToInstanceProfile`, `iam:CreateRole` e autorizzazioni. `iam:AttachRolePolicy`

> **Note**
>
> Questa opzione è disponibile solo in. AWS Management Console Se utilizzi la CLI o l'SDK, utilizza il CloudFormation modello RDS personalizzato o crea manualmente un profilo di istanza. Per ulteriori informazioni, consulta Creazione manuale del ruolo IAM e del profilo dell'istanza.

Fase 1: concedere le autorizzazioni necessarie al responsabile IAM

Assicurati di avere accesso sufficiente per creare un'istanza RDS Custom. Il ruolo IAM o l'utente IAM (denominato principale IAM) per la creazione di un'istanza DB RDS Custom for SQL Server utilizzando la console o la CLI deve disporre di una delle seguenti politiche per una corretta creazione dell'istanza DB:

- La policy `AdministratorAccess`
- La policy `AmazonRDSFullAccess` con le seguenti autorizzazioni aggiuntive:

```
iam:SimulatePrincipalPolicy
cloudtrail:CreateTrail
cloudtrail:StartLogging
s3:CreateBucket
s3:PutBucketPolicy
s3:PutBucketObjectLockConfiguration
s3:PutBucketVersioning
kms:CreateGrant
kms:DescribeKey
```

RDS Custom utilizza queste autorizzazioni durante la creazione dell'istanza. Queste autorizzazioni configurano le risorse dell'account necessarie per le operazioni RDS Custom.

Per ulteriori informazioni sull'autorizzazione `kms:CreateGrant`, consulta Gestione di AWS KMS key.

La policy JSON di esempio seguente fornisce le autorizzazioni necessarie.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ValidateIamRole",
            "Effect": "Allow",
            "Action": "iam:SimulatePrincipalPolicy",
            "Resource": "*"
        },
        {
            "Sid": "CreateCloudTrail",
            "Effect": "Allow",
            "Action": [
```

```
                "cloudtrail:CreateTrail",
                "cloudtrail:StartLogging"
            ],
            "Resource": "arn:aws:cloudtrail:*:*:trail/do-not-delete-rds-custom-*"
        },
        {
            "Sid": "CreateS3Bucket",
            "Effect": "Allow",
            "Action": [
                "s3:CreateBucket",
                "s3:PutBucketPolicy",
                "s3:PutBucketObjectLockConfiguration",
                "s3:PutBucketVersioning"
            ],
            "Resource": "arn:aws:s3:::do-not-delete-rds-custom-*"
        },
        {
            "Sid": "CreateKmsGrant",
            "Effect": "Allow",
            "Action": [
                "kms:CreateGrant",
                "kms:DescribeKey"
            ],
            "Resource": "*"
        }
    ]
}
```

Inoltre, il principale IAM richiede l'autorizzazione `iam:PassRole` sul ruolo IAM. Deve essere collegata al profilo dell'istanza passato nel parametro `custom-iam-instance-profile` nella richiesta di creazione dell'istanza database di RDS Custom. Il profilo dell'istanza e il suo ruolo collegato vengono creati in seguito in Passaggio 2: configurare la rete, il profilo dell'istanza e la crittografia.

> **Note**
>
> Assicurati che le autorizzazioni elencate in precedenza non siano limitate da policy di controllo dei servizi, da limiti delle autorizzazioni o da policy di sessione associati al principale IAM.

Passaggio 2: configurare la rete, il profilo dell'istanza e la crittografia

Puoi configurare il ruolo del profilo dell'istanza IAM, il cloud privato virtuale (VPC) e la chiave di crittografia AWS KMS simmetrica utilizzando uno dei seguenti processi:

- Configurazione con AWS CloudFormation (consigliato)
- Configurazione manuale

> **Note**
>
> Se il tuo account fa parte di uno di essi AWS Organizations, assicurati che le autorizzazioni richieste dal ruolo del profilo dell'istanza non siano limitate dalle policy di controllo del servizio (SCP).
> Le configurazioni di rete riportate in questo argomento funzionano meglio con istanze DB che non sono accessibili pubblicamente. Non è possibile connettersi direttamente a tali istanze DB dall'esterno del VPC.

Configurazione con AWS CloudFormation

Per semplificare la configurazione, puoi utilizzare un file AWS CloudFormation modello per creare uno CloudFormation stack. Un CloudFormation modello crea tutte le reti, i profili di istanza e le risorse di crittografia in base ai requisiti di RDS Custom.

Per informazioni su come creare pile, consulta Creazione di uno stack sulla AWS CloudFormation console nella Guida per l'utente.AWS CloudFormation

Per un tutorial su come avviare Amazon RDS Custom for SQL Server utilizzando un AWS CloudFormation modello, consulta Get started with Amazon RDS Custom for SQL Server using an AWS CloudFormation template nel AWS Database Blog.

Argomenti

- Parametri richiesti da CloudFormation
- Scarica il file modello AWS CloudFormation
- Configurazione delle risorse tramite CloudFormation

Parametri richiesti da CloudFormation

I seguenti parametri sono necessari per configurare le risorse dei prerequisiti RDS Custom con: CloudFormation

Gruppo di parametri	Nome del parametro	Valore predefinito	Descrizione
Configurazione della disponibilità	Seleziona una configurazione di disponibilità per l'impostazione dei prerequisiti	Multi-AZ	Specificare se impostare i prerequisiti nella configurazione Single-AZ o Multi-AZ per le istanze RDS Custom. È necessario utilizzare la configurazione Multi-AZ se è necessaria almeno un'istanza DB Multi-AZ in questa configurazione
Configurazione di rete	Blocco CIDR IPv4 per VPC	10.0.0.0/16	Specificate un blocco CIDR IPv4 (o intervallo di indirizzi IP) per il VPC. Questo VPC è configurato per creare e utilizzare un'istanza DB personalizzata RDS.
	Blocco CIDR IPv4 per 1 delle 2 sottoreti private	10.0.128.0/20	Specificate un blocco CIDR IPv4 (o intervallo di indirizzi IP) per la prima sottorete privata. Questa è una delle due sottoreti in cui è possibile creare l'istanza DB personalizzata RDS. Si tratta di

Gruppo di parametri	Nome del parametro	Valore predefinito	Descrizione
			una sottorete privata senza accesso a Internet.
	Blocco CIDR IPv4 per 2 sottoreti private su 2	10.0.144.0/20	Specificate un blocco CIDR IPv4 (o intervallo di indirizzi IP) per la seconda sottorete privata. Questa è una delle due sottoreti in cui è possibile creare l'istanza DB personalizzata RDS. Si tratta di una sottorete privata senza accesso a Internet.
	Blocco CIDR IPv4 per sottorete pubblica	10.0.0.0/20	Specificate un blocco CIDR IPv4 (o intervallo di indirizzi IP) per la sottorete pubblica. Questa è una delle sottoreti in cui l'istanza EC2 può connettersi con l'istanza DB personalizzata RDS. Si tratta di una sottorete pubblica con accesso a Internet.

Gruppo di parametri	Nome del parametro	Valore predefinito	Descrizione
Configurazione dell'accesso RDP	Blocco CIDR IPv4 del codice sorgente	-	Specificate un blocco CIDR IPv4 (o intervallo di indirizzi IP) della fonte. Questo è l'intervallo IP da cui si effettua la connessione RDP all'istanza EC2 nella sottorete pubblica. Se non è impostata, la connessione RDP all'istanza EC2 non è configurata.
	Imposta l'accesso RDP all'istanza RDS Custom for SQL Server	No	Specificare se abilitare la connessione RDP dall'istanza EC2 all'istanza RDS Custom for SQL Server. Per impostazione predefinita, la connessione RDP dall'istanza EC2 all'istanza DB non è configurata.

Risorse create da CloudFormation

La corretta creazione dello CloudFormation stack utilizzando le impostazioni predefinite crea le seguenti risorse nel tuo Account AWS:

- Chiave KMS di crittografia simmetrica per la crittografia dei dati gestiti da RDS Custom.
- Il profilo dell'istanza è associato a un ruolo IAM `AmazonRDSCustomInstanceProfileRolePolicy` per fornire le autorizzazioni richieste da

RDS Custom. Per ulteriori informazioni, consulta Amazon RDS CustomServiceRolePolicy nella Managed Policy Reference Guide.AWS

- VPC con l'intervallo CIDR specificato come parametro. CloudFormation Il valore predefinito è `10.0.0.0/16`.
- Due sottoreti private con l'intervallo CIDR specificato nei parametri e due diverse zone di disponibilità nella Regione AWS. I valori predefiniti per i CIDR sottorete sono `10.0.128.0/20` e `10.0.144.0/20`.
- Una sottorete pubblica con l'intervallo CIDR specificato nei parametri. Il valore predefinito per la sottorete CIDR è 10.0.0.0/20. L'istanza EC2 risiede in questa sottorete e può essere utilizzata per connettersi all'istanza RDS Custom.
- Opzione DHCP impostata per il VPC con risoluzione dei nomi di dominio su un server di sistema dei nomi di dominio (DNS) Amazon.
- Tabella di instradamento da associare a due sottoreti private e nessun accesso a Internet.
- Tabella di routing da associare alla sottorete pubblica e con accesso a Internet.
- Gateway Internet associato al VPC per consentire l'accesso a Internet alla sottorete pubblica.
- Elenco di controllo degli accessi alla rete (ACL) da associare a due sottoreti private e accesso limitato a HTTPS e alla porta DB all'interno di VPC.
- Gruppo di sicurezza VPC da associare all'istanza di RDS Custom. L'accesso per l'HTTPS in uscita è limitato agli Servizio AWS endpoint richiesti da RDS Custom e alla porta DB in entrata dal gruppo di sicurezza delle istanze EC2.
- Gruppo di sicurezza VPC da associare all'istanza EC2 nella sottorete pubblica. L'accesso è limitato per la porta DB in uscita al gruppo di sicurezza dell'istanza RDS Custom.
- Gruppo di sicurezza VPC da associare agli endpoint VPC creati per gli endpoint richiesti da Servizio AWS RDS Custom.
- Gruppo di sottoreti DB in cui vengono create istanze di RDS Custom. Due sottoreti private create da questo modello vengono aggiunte al gruppo di sottoreti DB.
- Endpoint VPC per ciascuno degli Servizio AWS endpoint richiesti da RDS Custom.

L'impostazione della configurazione della disponibilità su multi-az creerà le seguenti risorse oltre all'elenco precedente:

- Regole ACL di rete che consentono la comunicazione tra sottoreti private.
- Accesso in entrata e in uscita alla porta Multi-AZ all'interno del gruppo di sicurezza VPC associato all'istanza RDS Custom.

- Da endpoint VPC a endpoint AWS di servizio necessari per la comunicazione Multi-AZ.

Inoltre, l'impostazione della configurazione di accesso RDP crea le seguenti risorse:

- Configurazione dell'accesso RDP alla sottorete pubblica dall'indirizzo IP di origine:
  - Regole ACL di rete che consentono la connessione RDP dall'IP di origine alla sottorete pubblica.
  - Accedi alla porta RDP dall'IP di origine al gruppo di sicurezza VPC associato all'istanza EC2.
- Configurazione dell'accesso RDP dall'istanza EC2 nella sottorete pubblica all'istanza personalizzata RDS nelle sottoreti private:
  - Regole ACL di rete che consentono la connessione RDP dalla sottorete pubblica alle sottoreti private.
  - Accesso in entrata alla porta RDP dal gruppo di sicurezza VPC associato all'istanza EC2 al gruppo di sicurezza VPC associato all'istanza personalizzata RDS.

Utilizza le seguenti procedure per creare lo CloudFormation stack per RDS Custom for SQL Server.

Scarica il file modello AWS CloudFormation

Scaricare il file di modello

1. Apri il menu contestuale (fai clic con il pulsante destro del mouse) per il link custom-sqlserver-onboard.zip e scegli Salva collegamento con nome.
2. Salva ed estrai il file sul computer.

Configurazione delle risorse tramite CloudFormation

Per configurare le risorse utilizzando CloudFormation

1. Apri la CloudFormation console all'indirizzo https://console.aws.amazon.com/cloudformation.
2. Per avviare la creazione guidata dello stack, scegli Create Stack (Crea stack).

   Viene visualizzata la pagina Create stack (Crea stack).
3. Per Prerequisito - Prepara modello, scegliere Il modello è pronto.
4. Per Specify template (Specifica modello), procedi come segue:

a. Come Template source (Origine modello), scegliere Upload a template file (Carica un file di modello).

b. Per Scegli file, individua e quindi scegli il file corretto.

5. Seleziona Successivo.

   Viene visualizzata la pagina Specify stack details (Specifica dettagli stack).

6. In Nome stack, immetti **`rds-custom-sqlserver`**.

7. Per Parameters (Parametri), effettua le seguenti operazioni:

   a. Per mantenere le opzioni predefinite, scegli Next (Avanti).

   b. Per modificare le opzioni, scegliete la configurazione di disponibilità appropriata, la configurazione di rete e la configurazione di accesso RDP, quindi scegliete Avanti.

      Leggi attentamente la descrizione di ciascun parametro prima di modificare i parametri.

> **Note**
>
> Se scegli di creare almeno un'istanza Multi-AZ in questo CloudFormation stack, assicurati che il parametro CloudFormation stack Seleziona una configurazione di disponibilità per la configurazione dei prerequisiti sia impostato su. `Multi-AZ` Se crei lo CloudFormation stack come Single-AZ, aggiorna lo stack alla configurazione Multi-AZ prima di creare la CloudFormation prima istanza Multi-AZ.

8. Nella pagina Configure stack options (Configura opzioni pila), scegliere Next (Successivo).

9. Nella rds-custom-sqlserver pagina Revisione, procedi come segue:

   a. In Capabilities (Capacità), selezionare la casella di spunta I acknowledge that AWS CloudFormation might create IAM resources with custom names (Conferma che potrebbe creare risorse IAM con nomi personalizzati).

   b. Seleziona Crea stack.

> **Note**
>
> Non aggiornate le risorse create da questo AWS CloudFormation stack direttamente dalle pagine delle risorse. Ciò impedisce di applicare aggiornamenti futuri a queste risorse utilizzando un AWS CloudFormation modello.

CloudFormation crea le risorse richieste da RDS Custom for SQL Server. Se la creazione dello stack non va a buon fine, leggi la scheda Events (Eventi) per vedere quale creazione risorsa non è andata a buon fine e il motivo dello stato.

La scheda Output per questo CloudFormation stack nella console dovrebbe contenere informazioni su tutte le risorse da passare come parametri per la creazione di un'istanza DB RDS Custom for SQL Server. Assicurati di utilizzare il gruppo di sicurezza VPC e il gruppo di sottoreti DB creati da CloudFormation per le istanze DB personalizzate RDS. Per impostazione predefinita, RDS tenta di collegare il gruppo di sicurezza VPC predefinito, che potrebbe non disporre dell'accesso necessario.

Se prima creavi CloudFormation risorse, puoi saltare. Configurazione manuale

Aggiornamento dello stack CloudFormation

Puoi anche aggiornare alcune configurazioni dello CloudFormation stack dopo la creazione. Le configurazioni che possono essere aggiornate sono:

- Configurazione della disponibilità per RDS Custom per SQL Server
  - Seleziona una configurazione di disponibilità per l'impostazione dei prerequisiti: aggiorna questo parametro per passare dalla configurazione Single-AZ a quella Multi-AZ. Se si utilizza questo CloudFormation stack per almeno un'istanza Multi-AZ, è necessario aggiornare lo stack per scegliere la configurazione Multi-AZ.
- Configurazione di accesso RDP per RDS Custom per SQL Server
  - Blocco CIDR IPv4 dell'origine: è possibile aggiornare il blocco CIDR IPv4 (o intervallo di indirizzi IP) dell'origine aggiornando questo parametro. L'impostazione di questo parametro su vuoto rimuove la configurazione di accesso RDP dal blocco CIDR di origine alla sottorete pubblica.
  - Imposta l'accesso RDP a RDS Custom for SQL Server: abilita o disabilita la connessione RDP dall'istanza EC2 all'istanza RDS Custom for SQL Server.

Eliminazione dello stack CloudFormation

È possibile eliminare lo CloudFormation stack dopo aver eliminato tutte le istanze RDS Custom che utilizzano risorse dallo stack. RDS Custom non tiene traccia dello CloudFormation stack, quindi non blocca l'eliminazione dello stack quando ci sono istanze DB che utilizzano risorse dello stack. Assicurati che non ci siano istanze DB RDS Custom che utilizzano le risorse dello stack durante l'eliminazione dello stack.

> **Note**
>
> Quando elimini uno CloudFormation stack, tutte le risorse create dallo stack vengono eliminate tranne la chiave KMS. La chiave KMS entra in uno stato di eliminazione in sospeso e viene eliminata dopo 30 giorni. Per conservare la chiave KMS, esegui un'CancelKeyDeletionoperazione durante il periodo di prova di 30 giorni.

Configurazione manuale

Se scegli di configurare le risorse manualmente, esegui le seguenti operazioni.

> **Note**
>
> Per semplificare la configurazione, puoi utilizzare il file AWS CloudFormation modello per creare uno CloudFormation stack anziché una configurazione manuale. Per ulteriori informazioni, consulta Configurazione con AWS CloudFormation.
> È inoltre possibile utilizzare il AWS CLI per completare questa sezione. In tal caso, scarica e installa la CLI più recente.

Argomenti

- Assicurati di disporre di una chiave di crittografia simmetrica AWS KMS
- Creazione manuale del ruolo IAM e del profilo dell'istanza
- Configurazione manuale del VPC

Assicurati di disporre di una chiave di crittografia simmetrica AWS KMS

È richiesta una crittografia simmetrica per RDS AWS KMS key Custom. Quando crei un'istanza DB RDS Custom for SQL Server, assicurati di fornire l'identificatore della chiave KMS come parametro.

`kms-key-id` Per ulteriori informazioni, consulta Creazione e connessione a un'istanza database per Amazon RDS Custom per SQL Server.

Sono disponibili le seguenti opzioni:

- Se disponi già di una chiave KMS gestita dal cliente Account AWS, puoi utilizzarla con RDS Custom. Non è richiesta alcuna operazione aggiuntiva.
- Se hai già creato una chiave KMS di crittografia simmetrica gestita dal cliente per un motore RDS Custom diverso, puoi riutilizzare la stessa chiave KMS. Non è richiesta alcuna operazione aggiuntiva.
- Se non disponi di una chiave KMS di crittografia simmetrica gestita dal cliente esistente nel tuo account, crea una chiave KMS seguendo le istruzioni in Creazione di chiavi nella Guida per gli sviluppatori di AWS Key Management Service .
- Se stai creando un'istanza DB personalizzata CEV o RDS e la tua chiave KMS si trova in un'altra istanza Account AWS, assicurati di utilizzare la. AWS CLI Non puoi utilizzare la AWS console con chiavi KMS per più account.

> **Important**
>
> RDS Custom non supporta le chiavi KMS AWS gestite.

Assicurati che la tua chiave di crittografia simmetrica conceda l'accesso al ruolo `kms:Decrypt` and `kms:GenerateDataKey` operations to the AWS Identity and Access Management (IAM) nel profilo dell'istanza IAM. Se hai una nuova chiave di crittografia simmetrica nel tuo account, non sono necessarie modifiche. Altrimenti, assicurati che la policy della chiave di crittografia simmetrica fornisca l'accesso a queste operazioni.

Per ulteriori informazioni, consulta Fase 4: Configurazione di IAM for RDS Custom per Oracle.

Creazione manuale del ruolo IAM e del profilo dell'istanza

Puoi creare manualmente un profilo di istanza e utilizzarlo per avviare istanze RDS Custom. Se intendi creare l'istanza in AWS Management Console, salta questa sezione. AWS Management Console Consente di creare e collegare un profilo di istanza alle istanze DB personalizzate RDS. Per ulteriori informazioni, consulta Creazione automatica del profilo di istanza utilizzando il AWS Management Console.

Quando crei manualmente un profilo di istanza, passa il nome del profilo dell'istanza come `custom-iam-instance-profile` parametro al comando `create-db-instance` CLI. RDS Custom utilizza il ruolo associato a questo profilo di istanza per eseguire l'automazione e gestire l'istanza.

Per utilizzare il profilo dell'istanza IAM e ruoli IAM per RDS Custom per SQL Server

1. Creare il ruolo IAM denominato `AWSRDSCustomSQLServerInstanceRole` con una policy di attendibilità che Amazon EC2 può utilizzare per assumere questo ruolo.
2. Aggiungi la policy AWS gestita `AmazonRDSCustomInstanceProfileRolePolicy` a. `AWSRDSCustomSQLServerInstanceRole`
3. Crea un profilo dell'istanza IAM per RDS Custom per SQL Server denominato `AWSRDSCustomSQLServerInstanceProfile`.
4. Aggiungere `AWSRDSCustomSQLServerInstanceRole` al profilo dell'istanza.

Crea il ruolo AWSRDSCustomSQLServerInstanceRole IAM

L'esempio seguente crea il ruolo `AWSRDSCustomSQLServerInstanceRole`. Utilizzando la policy di affidabilità, Amazon EC2 può assumere il ruolo.

```
aws iam create-role \
    --role-name AWSRDSCustomSQLServerInstanceRole \
    --assume-role-policy-document '{
        "Version": "2012-10-17",
          "Statement": [
            {
              "Action": "sts:AssumeRole",
              "Effect": "Allow",
              "Principal": {
                  "Service": "ec2.amazonaws.com"
              }
            }
          ]
        }'
```

Aggiungi una politica di accesso a AWSRDSCustomSQLServerInstanceRole

Per fornire le autorizzazioni richieste, allega la politica AWS gestita `AmazonRDSCustomInstanceProfileRolePolicy` a`AWSRDSCustomSQLServerInstanceRole`.

`AmazonRDSCustomInstanceProfileRolePolicy`consente alle istanze RDS Custom di inviare e ricevere messaggi ed eseguire varie azioni di automazione.

> **Note**
>
> Assicurati che le autorizzazioni nella policy di accesso non siano limitate da SCP o dai limiti di autorizzazione associati al ruolo del profilo dell'istanza.

L'esempio seguente associa una policy AWS gestita `AWSRDSCustomSQLServerIamRolePolicy` al ruolo. `AWSRDSCustomSQLServerInstanceRole`

```
aws iam attach-role-policy \
    --role-name AWSRDSCustomSQLServerInstanceRole \
    --policy-arn arn:aws:iam::aws:policy/AmazonRDSCustomInstanceProfileRolePolicy
```

Creare un profilo dell'istanza RDS Custom for SQL server

Un profilo dell'istanza è un container che include un ruolo IAM singolo. RDS Custom utilizza il profilo dell'istanza per trasferire il ruolo all'istanza.

Se utilizzi il per AWS Management Console creare un ruolo per Amazon EC2, la console crea automaticamente un profilo di istanza e gli assegna lo stesso nome del ruolo al momento della creazione del ruolo. Creare il profilo dell'istanza IAM come segue, denominandolo `AWSRDSCustomSQLServerInstanceProfile`.

```
aws iam create-instance-profile \
    --instance-profile-name AWSRDSCustomSQLServerInstanceProfile
```

Aggiungilo AWSRDSCustomSQLServerInstanceRole al tuo profilo di istanza RDS Custom for SQL Server

Aggiungi il `AWSRDSCustomInstanceRoleForRdsCustomInstance` ruolo al `AWSRDSCustomSQLServerInstanceProfile` profilo creato in precedenza.

```
aws iam add-role-to-instance-profile \
    --instance-profile-name AWSRDSCustomSQLServerInstanceProfile \
    --role-name AWSRDSCustomSQLServerInstanceRole
```

Configurazione manuale del VPC

L'istanza database di RDS Custom si trova in un cloud privato virtuale (VPC) basato sul servizio Amazon VPC, proprio come un'istanza Amazon EC2 o un'istanza Amazon RDS. Fornisci e configuri il VPC personalizzato. Pertanto, hai il pieno controllo sulla configurazione della rete delle istanze.

RDS Custom invia la comunicazione dall'istanza database ad altri Servizi AWS. Assicurati che i seguenti servizi siano accessibili dalla sottorete in cui crei le tue istanze DB personalizzate RDS:

- Amazon CloudWatch
- CloudWatch Registri Amazon
- CloudWatch Eventi Amazon
- Amazon EC2
- Amazon EventBridge
- Amazon S3
- AWS Secrets Manager
- AWS Systems Manager

Se si creano implementazioni Multi-AZ

- Amazon Simple Queue Service

Se RDS Custom non è in grado di comunicare con i servizi necessari, pubblica i seguenti eventi:

```
Database instance in incompatible-network. SSM Agent connection not available. Amazon
 RDS can't connect to the dependent AWS services.
```

```
Database instance in incompatible-network. Amazon RDS can't connect to dependent AWS
 services. Make sure port 443 (HTTPS) allows outbound connections, and try again.
 "Failed to connect to the following services: s3 events"
```

Per evitare `incompatible-network` errori, assicurati che i componenti VPC coinvolti nella comunicazione tra l'istanza DB personalizzata di RDS Servizi AWS soddisfino i seguenti requisiti:

- L'istanza database può effettuare connessioni in uscita sulla porta 443 ad altri Servizi AWS.
- Il VPC consente risposte in entrata alle richieste che originano dall'istanza database RDS Custom.

- RDS Custom può risolvere correttamente i nomi di dominio degli endpoint per ogni Servizio AWS.

Se hai già configurato un VPC per un motore di database RDS Custom diverso, puoi riutilizzare tale VPC e ignorare questo processo.

Argomenti

- Configura il tuo gruppo di sicurezza VPC
- Configura gli endpoint per dipendenti Servizi AWS
- Configurazione del servizio di metadati dell'istanza

Configura il tuo gruppo di sicurezza VPC

Un gruppo di sicurezza funge da firewall virtuale per un'istanza VPC, controllando il traffico in entrata e quello in uscita. Un'istanza DB personalizzata RDS dispone di un gruppo di sicurezza collegato all'interfaccia di rete che protegge l'istanza. Assicurati che il tuo gruppo di sicurezza consenta il traffico tra RDS Custom e altri Servizi AWS tramite HTTPS. Questo gruppo di sicurezza viene passato come `vpc-security-group-ids` parametro nella richiesta di creazione dell'istanza.

Per configurare il gruppo di sicurezza per RDS Custom

1. Accedi AWS Management Console e apri la console Amazon VPC all'indirizzo https://console.aws.amazon.com/vpc.
2. Consenti a RDS Custom di utilizzare il gruppo di sicurezza predefinito o di creare un gruppo di sicurezza personalizzato.

   Per istruzioni dettagliate, vedi Fornisci accesso alla istanza database nel VPC creando un gruppo di sicurezza.
3. Assicurati che il gruppo di sicurezza consenta le connessioni in uscita sulla porta 443. RDS Custom ha bisogno di questa porta per comunicare con Servizi AWS dipendenti.
4. Se disponi di un VPC privato e utilizzi endpoint VPC, assicurati che il gruppo di sicurezza associato all'istanza database consenta le connessioni in uscita sulla porta 443 a endpoint VPC. Assicurati inoltre che il gruppo di sicurezza associato all'endpoint VPC consenta connessioni in entrata sulla porta 443 dall'istanza database.

   Se le connessioni in entrata non sono consentite, l'istanzaRDS Custom non è in grado di connettersi a AWS Systems Manager e agli endpoint Amazon EC2. Per ulteriori informazioni,

consulta Creazione di un endpoint di un Virtual Private Cloud nella Guida per l'utente di AWS Systems Manager .

5. Per le istanze RDS Custom for SQL Server Multi-AZ, assicurati che il gruppo di sicurezza associato all'istanza DB consenta le connessioni in entrata e in uscita sulla porta 1120 verso questo gruppo di sicurezza stesso. Ciò è necessario per la connessione peer-host su un'istanza DB Multi-AZ RDS Custom for SQL Server.

Per ulteriori informazioni sui gruppi di sicurezza, consulta Gruppi di sicurezza per il VPC nella Guida per gli sviluppatori Amazon VPC.

Configura gli endpoint per dipendenti Servizi AWS

È consigliato aggiungere endpoint per ogni servizio VPC utilizzando le seguenti istruzioni. Tuttavia, puoi utilizzare qualsiasi soluzione che consenta al tuo VPC di comunicare con gli endpoint del AWS servizio. Ad esempio, è possibile utilizzare Network Address Translation (NAT) o AWS Direct Connect.

Per configurare gli endpoint Servizi AWS con cui funziona RDS Custom

1. Apri alla console Amazon VPC all'indirizzo https://console.aws.amazon.com/vpc/.
2. Per cambiare regione, usa il selettore di regione nella barra di navigazione per scegliere Regione AWS.
3. Nel pannello di navigazione, seleziona Endpoints (Endpoint). Nel riquadro principale, seleziona Create Endpoint (Crea endpoint).
4. Per Service category (Categoria servizio), scegli Servizi AWS.
5. Per Service Name (Nome servizio), scegliere l'endpoint mostrato nella tabella.
6. In VPC, seleziona il VPC.
7. In Subnets (Sottoreti), scegli una sottorete per ogni zona di disponibilità da includere.

   L'endpoint VPC può estendersi su più zone di disponibilità. AWS crea un'interfaccia di rete elastica per l'endpoint VPC in ogni sottorete scelta. Ogni interfaccia di rete dispone di un nome host Domain Name System (DNS) e di un indirizzo IP privato.
8. Per Security groups (Gruppi di sicurezza), seleziona o crea un gruppo di sicurezza.

   Puoi utilizzare i gruppi di sicurezza per controllare l'accesso al tuo endpoint, come se utilizzassi un firewall. Assicurati che il gruppo di sicurezza consenta le connessioni in entrata sulla porta

443 dalle istanze DB. Per ulteriori informazioni sui gruppi di sicurezza, consulta Gruppi di sicurezza per il VPC nella Guida per l'utente di Amazon VPC.

9. Facoltativamente, puoi collegare una policy all'endpoint VPC. Le policy degli endpoint possono controllare l'accesso al sistema Servizio AWS a cui ti stai connettendo. La policy predefinita consente a tutte le richieste di passare attraverso l'endpoint. Se utilizzi una policy personalizzata, assicurati che le richieste dall'istanza database siano consentite nella policy.
10. Seleziona Crea endpoint.

Nella tabella seguente viene illustrato come trovare l'elenco degli endpoint di cui il VPC ha bisogno per le comunicazioni in uscita.

Servizio	Formato dell'endpoint	Note e link
AWS Systems Manager	Utilizzare i seguenti formati dell'endpoint: • `ssm.region.amazonaws.com` • `ssmmessages. region.amazonaws.com`	Per un elenco di tutti gli endpoint in ogni regione, consulta Endpoint e quote di AWS Systems Manager in Riferimenti generali di Amazon Web Services.
AWS Secrets Manager	Utilizzare il formato dell'endpoint `secretsmanager. region.amazonaws.com .`	Per un elenco di tutti gli endpoint in ogni regione, consulta Endpoint e quote di AWS Secrets Manager in Riferimenti generali di Amazon Web Services.
Amazon CloudWatch	Utilizzare i seguenti formati dell'endpoint: • Per le CloudWatch metriche, usa `monitoring. region.amazonaws.com` • Per CloudWatch gli eventi, usa `events.region.amazonaws.com` • Per CloudWatch i registri, usa `logs.region.amazonaws.com`	Per l'elenco degli endpoint in ogni regione, consultare: • CloudWatch Endpoint e quote Amazon nel Riferimenti generali di Amazon Web Services • Amazon CloudWatch registra gli endpoint e le quote nel Riferimenti generali di Amazon Web Services

Servizio	Formato dell'endpoint	Note e link
		• Endpoint e quote di Amazon CloudWatch Events nel Riferimenti generali di Amazon Web Services
Amazon EC2	Utilizzare i seguenti formati dell'endpoint: • `ec2.region.amazonaws.com` • `ec2messages. region.amazonaws.com`	Per un elenco completo degli endpoint in ogni regione, consulta Endpoint e quote di Amazon Elastic Compute Cloud in Riferimenti generali di Amazon Web Services.
Amazon S3	Utilizzare il formato dell'endpoint `s3.region.amazonaws.com` .	Per un elenco completo degli endpoint in ogni regione, consulta Endpoint e quote di Amazon Simple Storage Service in Riferimenti generali di Amazon Web Services. Per ulteriori informazioni sugli endpoint gateway per Simple Storage Service (Amazon S3), consultare Endpoint per Amazon S3 nella Guida per gli sviluppatori Amazon VPC. Per informazioni su come creare un punto di accesso, consultare Creazione di access point nella Guida per gli sviluppatori Amazon VPC. Per informazioni su come creare endpoint gateway per Simple Storage Service (Amazon S3), consultare Endpoint VPC gateway.

Servizio	Formato dell'endpoint	Note e link
Amazon Simple Queue Service	Usa il formato endpoint `sqs.region.amazonaws.com`	Per l'elenco degli endpoint in ogni regione, consulta Endpoint e quote di Amazon Simple Queue Service.

Configurazione del servizio di metadati dell'istanza

Verificare che l'istanza possa fare:

- Accedere ai metadati dell'istanza utilizzando la versione 2 del servizio di metadati dell'istanza (IMDSv2).
- Consenti comunicazioni in uscita tramite la porta 80 (HTTP) all'indirizzo IP del collegamento IMDS.
- Richiedere metadati dell'istanza da`http://169.254.169.254`, il link IMDSv2.

Per ulteriori informazioni, consultare Utilizzare IMDSv2 nella Guida per l'utente di Amazon EC2 per le istanze Linux.

Restrizione tra istanze

Quando si crea un profilo di istanza seguendo i passaggi precedenti, viene utilizzata la policy AWS gestita `AmazonRDSCustomInstanceProfileRolePolicy` per fornire le autorizzazioni necessarie a RDS Custom, che consente la gestione delle istanze e l'automazione del monitoraggio. La politica gestita garantisce che le autorizzazioni consentano l'accesso solo alle risorse necessarie a RDS Custom per eseguire l'automazione. Si consiglia di utilizzare la policy gestita per supportare nuove funzionalità e soddisfare i requisiti di sicurezza che vengono applicati automaticamente ai profili di istanza esistenti senza intervento manuale. Per ulteriori informazioni, consulta la policy AWS gestita: AmazonRDSCuso m. InstanceProfileRolePolicy

La politica `AmazonRDSCustomInstanceProfileRolePolicy` gestita limita l'accesso a più account del profilo dell'istanza, ma potrebbe consentire l'accesso ad alcune risorse gestite di RDS Custom tra istanze RDS Custom all'interno dello stesso account. In base alle tue esigenze, puoi utilizzare i limiti di autorizzazione per limitare ulteriormente l'accesso tra istanze. I limiti di autorizzazione definiscono le autorizzazioni massime che le politiche basate sull'identità possono concedere a un'entità, ma non concedono le autorizzazioni di per sé. Per ulteriori informazioni, consulta Valutazione delle autorizzazioni effettive con limiti.

Ad esempio, la seguente politica limita il ruolo del profilo dell'istanza all'accesso a una AWS KMS chiave specifica e limita l'accesso alle risorse gestite RDS Custom tra istanze che utilizzano chiavi diverse. AWS KMS

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "DenyOtherKmsKeyAccess",
            "Effect": "Deny",
            "Action": "kms:*",
            "NotResource": "arn:aws:kms:region:acct_id:key/KMS_key_ID"
        },
        {
            "Sid": "NoBoundarySetByDefault",
            "Effect": "Allow",
            "Action": "*",
            "Resource": "*"
        }
    ]
}
```

> **Note**
>
> Assicurati che il limite delle autorizzazioni non blocchi le autorizzazioni concesse a RDS Custom. `AmazonRDSCustomInstanceProfileRolePolicy`

Modello Porta i tuoi media (BYOM) con RDS Custom per SQL Server

RDS Custom per SQL Server supporta due modelli di licenza: License Included (LI) e Porta i tuoi media (BYOM).

Con il modello di licenza BYOM, è possibile eseguire le seguenti operazioni:

1. Fornire e installare i file binari di Microsoft SQL Server con aggiornamenti cumulativi supportati (CU) su un'AMI Windows di AWS EC2.
2. Salvare l'AMI come immagine "gold", ovvero un modello che è possibile usare per creare una versione del motore personalizzato (CEV).
3. Creare un CEV a partire dall'immagine "gold".
4. Creare istanze database Amazon RDS Custom per SQL Server utilizzando la CEV.

Amazon RDS gestisce automaticamente queste istanze database.

> **Note**
>
> Se si dispone anche di un'istanza database RDS Custom for SQL Server con licenza inclusa (LI), non è possibile utilizzare il software SQL Server di questa istanza database con il modello di licenza BYOM. È necessario importare i file binari di SQL Server in ambiente BYOM.

Requisiti relativi al modello di licenza BYOM per RDS Custom per SQL Server

Gli stessi requisiti generali relativi alle versioni del motore personalizzato validi per RDS Custom per SQL Server si applicano anche al modello di licenza BYOM. Per ulteriori informazioni, consulta Requisiti per le CEV per RDS Custom per SQL Server.

Quando si usa il modello di licenza BYOM, assicurarsi di soddisfare i seguenti requisiti aggiuntivi:

- Utilizza una delle seguenti edizioni supportate: SQL Server 2022 o 2019 Enterprise, Standard o Developer edition.
- Concedere il privilegio del ruolo server sysadmin (SA) di SQL Server a `NT AUTHORITY\SYSTEM`.
- Mantenere il sistema operativo Windows Server configurato in base al fuso orario `UTC`.

Le istanze Windows di Amazon EC2 sono impostate sul fuso orario UTC per impostazione predefinita. Per ulteriori informazioni sulla visualizzazione e sulla modifica dell'ora per un'istanza Windows, consulta Impostazione dell'orario di un'istanza Windows.

- Aprire la porta TCP 1433 e la porta UDP 1434 per consentire le connessioni SSM.

Limitazioni relative al modello di licenza BYOM per RDS Custom per SQL Server

Le stesse limitazioni generali valide per RDS Custom per SQL Server si applicano anche al modello di licenza BYOM. Per ulteriori informazioni, consulta Requisiti e limitazioni per Amazon RDS Custom for SQL Server.

Per il modello di licenza BYOM, sono valide le seguenti limitazioni aggiuntive:

- È supportata solo l'istanza SQL Server predefinita (MSSQLSERVER). Le istanze SQL Server denominate non sono supportate. RDS Custom per SQL Server rileva e monitora solo l'istanza SQL Server predefinita.
- In ogni AMI è supportata una sola installazione di SQL Server. Non sono supportate installazioni multiple di versioni diverse di SQL Server.
- SQL Server Web Edition non è supportata con il modello di licenza BYOM.
- Le versioni di prova delle edizioni di SQL Server non sono supportate con il modello di licenza BYOM. Quando installi SQL Server, non selezionare la casella di controllo per l'utilizzo di una versione di prova.
- Il supporto varia a seconda delle versioni specifiche di ciascun motore di database e a seconda delle Regioni AWS. Per ulteriori informazioni, consultare Disponibilità delle regioni per le CEV di RDS Custom per SQL Server e Supporto delle versioni per le CEV di RDS Custom per SQL Server.

Creazione di un'istanza database RDS Custom per SQL Server con il modello di licenza BYOM

Per preparare e creare un'istanza database RDS Custom for SQL Server con il modello di licenza BYOM, consulta Preparazione di una CEV utilizzando il modello Porta i tuoi media (BYOM).

Utilizzo di versioni del motore personalizzate per RDS Custom per SQL Server

Una versione del motore personalizzato (CEV) per RDS Custom per SQL Server è una Amazon Machine Image (AMI) con Microsoft SQL Server preinstallato.

I passaggi fondamentali del flusso di lavoro per una CEV sono i seguenti:

1. Scegliere un'AMI Windows di AWS EC2 da utilizzare come immagine di base per una CEV. È possibile utilizzare Microsoft SQL Server preinstallato o utilizzare il modello Bring Your Own Media (BYOM) per installare SQL Server autonomamente.
2. Installare altro software sul sistema operativo (OS) e personalizzare la configurazione del sistema operativo e di SQL Server in base alle specifiche esigenze aziendali.
3. Salvare l'AMI come immagine "gold".
4. Creare una versione del motore personalizzato (CEV) a partire dall'immagine "gold".
5. Creare istanze database Amazon RDS Custom per SQL Server utilizzando la CEV.

Amazon RDS gestisce quindi queste istanze database automaticamente.

Una CEV consente di mantenere la configurazione di base preferita del sistema operativo e del database. L'utilizzo di una CEV garantisce che la configurazione dell'host, ad esempio l'installazione di agenti di terze parti o altre personalizzazioni del sistema operativo, venga mantenuta nelle istanze database RDS Custom per SQL Server. Una CEV consente di implementare rapidamente parchi istanze di istanze database RDS Custom per SQL Server con la stessa configurazione.

Argomenti

Preparazione alla creazione di una CEV per RDS Custom per SQL Server

È possibile creare una CEV utilizzando una Amazon Machine Image (AMI) contenente Microsoft SQL Server preinstallato e con licenza inclusa (LI) o con una AMI su cui è installato il supporto di installazione di SQL Server (BYOM).

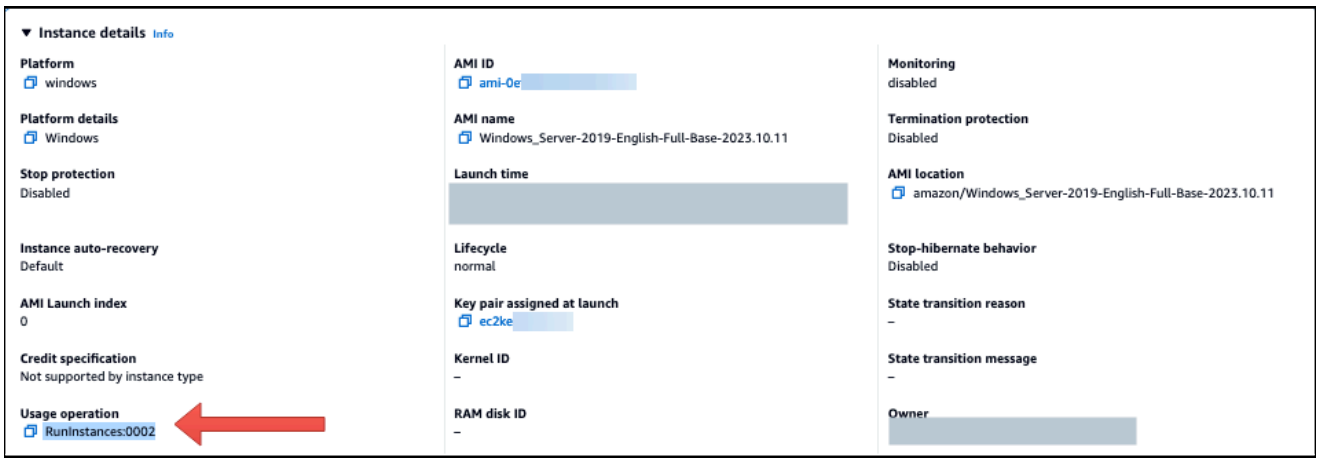
Indice

Preparazione di una CEV utilizzando il modello Porta i tuoi media (BYOM)

I passaggi seguenti utilizzano un'AMI con Windows Server 2019 Base come esempio.

Creazione di una CEV utilizzando il modello BYOM

1. Sulla console Amazon EC2, scegli Launch Instance.
2. Per Nome, inserisci il nome dell'istanza.
3. In Avvio rapido, scegli Windows.
4. Scegli Microsoft Windows Server 2019 Base.
5. Scegli il tipo di istanza, la coppia di chiavi, le impostazioni di rete e archiviazione appropriati e avvia l'istanza.
6. Dopo aver avviato o creato l'istanza EC2, assicurati che sia stata selezionata l'AMI Windows corretta dal passaggio 4:
    a. Seleziona l'istanza EC2 nella console Amazon EC2.
    b. Nella sezione Dettagli, controlla l'operazione di utilizzo e assicurati che sia impostata su:0002. RunInstances



7. Accedere all'istanza EC2 e copiarvi il supporto di installazione di SQL Server.

> **Note**
>
> Se stai creando un CEV utilizzando l'edizione SQL Server Developer, potrebbe essere necessario ottenere i supporti di installazione utilizzando l'abbonamento a Microsoft Visual Studio.

8. Installare SQL Server. Completare le seguenti operazioni:

   a. Revisione Requisiti relativi al modello di licenza BYOM per RDS Custom per SQL Server e. Supporto delle versioni per le CEV di RDS Custom per SQL Server

   b. Impostare la directory root dell'istanza sul valore predefinito `C:\Program Files \Microsoft SQL Server\`. Non modificare questa cartella.

   c. Impostare il nome dell'account del motore di database SQL Server su `NT Service \MSSQLSERVER` o `NT AUTHORITY\NETWORK SERVICE`.

   d. Impostare la modalità di avvio di SQL Server su Manuale.

   e. Per la modalità di autenticazione di SQL Server, scegliere Mista.

   f. Non modificare le impostazioni correnti predefinite per le directory Data e le posizioni TempDB.

9. Concedere il privilegio del ruolo server sysadmin (SA) di SQL Server a `NT AUTHORITY \SYSTEM`:

```
USE [master]
GO
```

```
EXEC master..sp_addsrvrolemember @loginame = N'NT AUTHORITY\SYSTEM' , @rolename =
 N'sysadmin'
GO
```

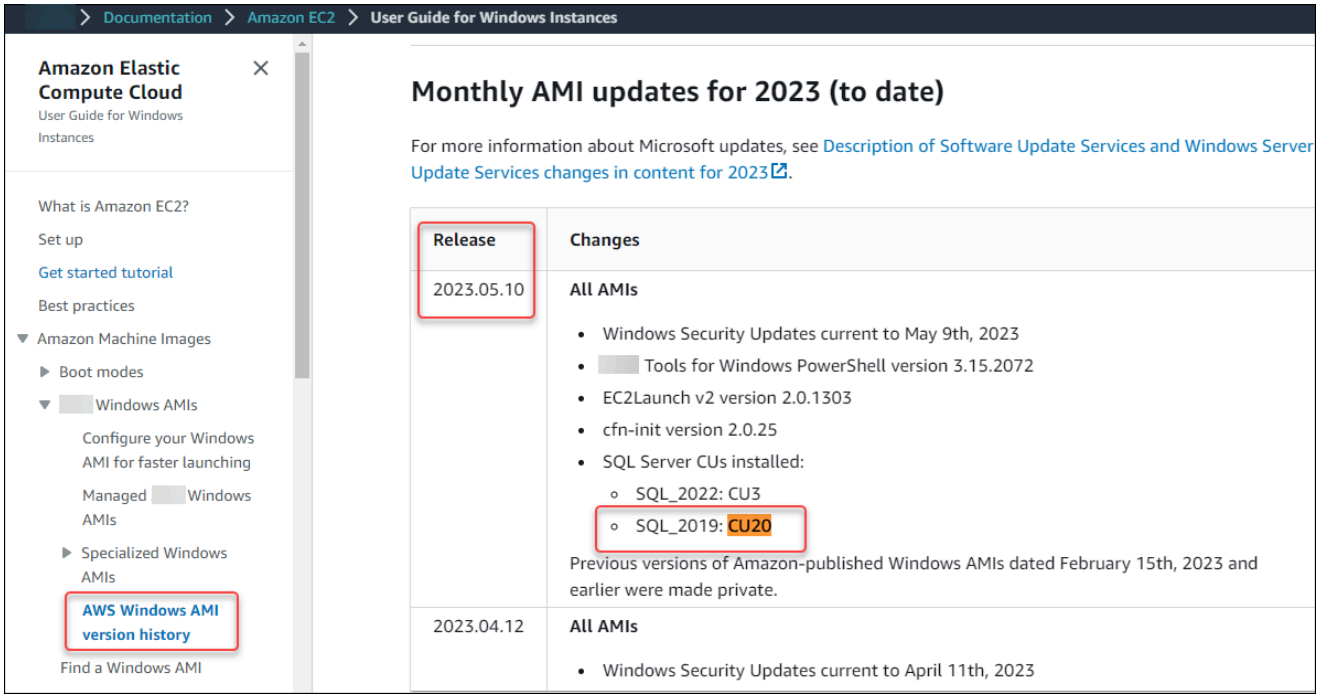
10. Installare il software aggiuntivo o personalizzare la configurazione del sistema operativo e del database in base alle specifiche esigenze di lavoro.
11. Esegui Sysprep sull'istanza EC2. Per ulteriori informazioni, consulta Creare un'immagine Amazon Machine Image (AMI) standardizzata utilizzando Sysprep.
12. Salvare l'AMI contenente la versione di SQL Server installata, altro software e personalizzazioni. Questa sarà l'immagine "gold".
13. Crea una nuova CEV fornendo l'ID AMI dell'immagine che hai creato. Per informazioni dettagliate sulle fasi, consulta Creazione di una CEV per RDS Custom per SQL Server.
14. Crea un'istanza database Amazon RDS Custom per SQL Server utilizzando la CEV. Per informazioni dettagliate sulle fasi, consulta Creazione di un'istanza database RDS Custom per SQL Server da una CEV.

Preparazione di una CEV utilizzando SQL Server (LI) preinstallato

I passaggi seguenti relativi alla creazione di una CEV utilizzando Microsoft SQL Server (LI) preinstallato utilizzano come esempio un'AMI con SQL Server con CU20 numero di versione `2023.05.10`. Quando si crea una CEV, scegliere un'AMI con il numero di versione più recente. Ciò garantisce l'utilizzo di una versione supportata di Windows Server e SQL Server con l'ultimo aggiornamento cumulativo (CU).

Creazione di una CEV utilizzando Microsoft SQL Server (LI) preinstallato

1. Scegli l'ultima versione disponibile di AWS EC2 Windows Amazon Machine Image (AMI) con licenza inclusa (LI) Microsoft Windows Server e SQL Server.
   a. Cercare CU20 nella cronologia delle versioni dell'AMI di Windows.
   b. Annotare il numero di versione. Per SQL Server 2019 con CU20, il numero di versione è `2023.05.10`.



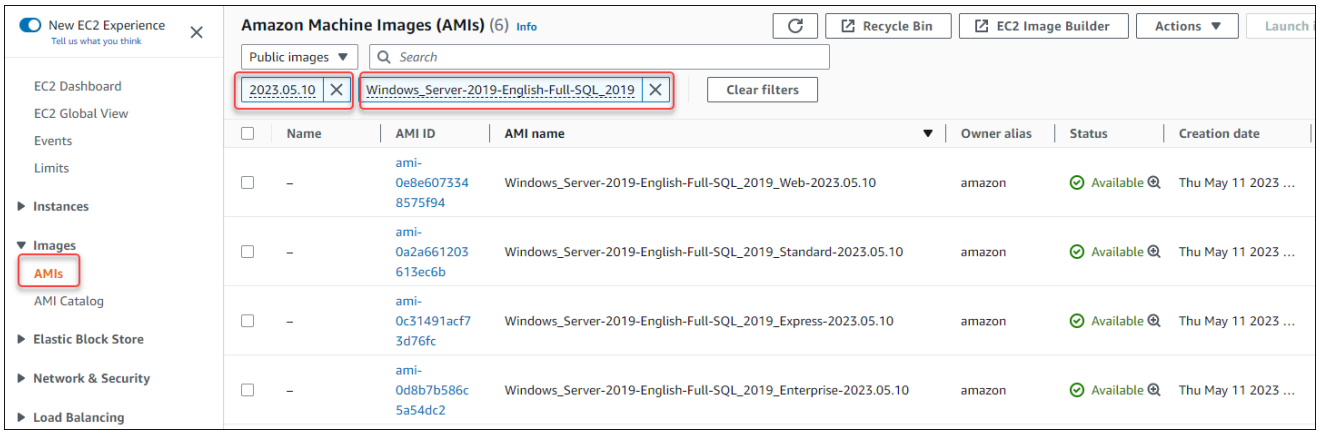
c. Apri la console Amazon EC2 all'indirizzo https://console.aws.amazon.com/ec2/.

d. Nel riquadro di navigazione della console Amazon EC2 scegli Images (Immagini) e quindi AMIs (AMI).

e. Scegliere Immagini pubbliche.

f. Immetti `2023.05.10` nella casella di ricerca. Viene visualizzato un elenco di AMI.

g. Immetti `Windows_Server-2019-English-Full-SQL_2019` nella casella di ricerca per filtrare i risultati. Vengono visualizzati i seguenti risultati.



h. Scegli l'AMI con l'edizione di SQL Server che vuoi utilizzare.

2. Crea o avvia un'istanza EC2 dall'AMI che hai scelto.

3. Accedi all'istanza EC2 e installa il software aggiuntivo o personalizza la configurazione del sistema operativo e del database per soddisfare le tue esigenze.

4. Esegui Sysprep sull'istanza EC2. Per ulteriori informazioni sulla preparazione di un'AMI utilizzando Sysprep, consulta Creare un'immagine Amazon Machine Image (AMI) standardizzata utilizzando Sysprep.
5. Salvare l'AMI contenente la versione di SQL Server installata, altro software e personalizzazioni. Questa sarà l'immagine "gold".
6. Crea una nuova CEV fornendo l'ID AMI dell'immagine che hai creato. Per i passaggi dettagliati sulla creazione di una CEV, consulta Creazione di una CEV per RDS Custom per SQL Server.
7. Crea un'istanza database Amazon RDS Custom per SQL Server utilizzando la CEV. Per informazioni dettagliate sulle fasi, consulta Creazione di un'istanza database RDS Custom per SQL Server da una CEV.

Disponibilità delle regioni per le CEV di RDS Custom per SQL Server

Il supporto per la versione Custom Engine (CEV) per RDS Custom for SQL Server è disponibile nei seguenti casi: Regioni AWS

- Stati Uniti orientali (Ohio)
- Stati Uniti orientali (Virginia settentrionale)
- US West (Oregon)
- Asia Pacifico (Mumbai)
- Asia Pacifico (Seoul)
- Asia Pacifico (Singapore)
- Asia Pacifico (Sydney)
- Asia Pacifico (Tokyo)
- Canada (Centrale)
- Europa (Francoforte)
- Europa (Irlanda)
- Europe (London)
- Europe (Stockholm)
- Sud America (San Paolo)

Supporto delle versioni per le CEV di RDS Custom per SQL Server

La creazione di CEV per RDS Custom for SQL Server è supportata per le seguenti AWS AMI Windows EC2:

- Per i CEV che utilizzano supporti preinstallati, AMI Windows AWS EC2 con licenza inclusa (LI) Microsoft Windows Server 2019 (OS) e SQL Server 2022 o 2019
- Per i CEV che utilizzano Bring your own media (BYOM), AMI AWS Windows EC2 con Microsoft Windows Server 2019 (OS)

La creazione di CEV per RDS Custom per SQL Server è supportata per le seguenti edizioni del sistema operativo e del database:

- Per i CEV che utilizzano supporti preinstallati:
  - SQL Server 2022 con CU9, per le edizioni Enterprise, Standard e Web
  - SQL Server 2019 con CU17, CU18, CU20 e CU24, per le edizioni Enterprise, Standard e Web
- Per i CEV che utilizzano Bring your own media (BYOM):
  - SQL Server 2022 con CU9, per le edizioni Enterprise, Standard e Developer
  - SQL Server 2019 con CU17, CU18, CU20 e CU24, per le edizioni Enterprise, Standard e Developer
- Per le CEV che utilizzano supporti preinstallati o il modello Bring Your Own Media (BYOM), Windows Server 2019 è l'unico sistema operativo supportato

Requisiti per le CEV per RDS Custom per SQL Server

I requisiti seguenti si applicano alla creazione di una CEV per RDS Custom per SQL Server:

- L'AMI utilizzata per creare una CEV deve basarsi su una configurazione del sistema operativo e del database supportata da RDS Custom per SQL Server. Per ulteriori informazioni sulle configurazioni supportate, consulta Requisiti e limitazioni per Amazon RDS Custom for SQL Server.
- La CEV deve avere un nome univoco. Non è possibile creare una CEV con lo stesso nome di una CEV esistente.
- È necessario denominare la CEV in base al modello di denominazione della versione principale + versione secondaria + stringa personalizzata di SQL Server. La versione principale + versione secondaria deve corrispondere alla versione di SQL Server fornita con l'AMI. Ad esempio, puoi denominare un'AMI con SQL Server 2019 CU17 come 15.00.4249.2.my_cevtest.

- È necessario preparare un'AMI con Sysprep. Per ulteriori informazioni sulla preparazione di un'immagine AMI utilizzando Sysprep, consulta Creare un'immagine Amazon Machine Image (AMI) standardizzata utilizzando Sysprep.
- Sei responsabile del mantenimento del ciclo di vita dell'AMI. Un'istanza database RDS Custom per SQL Server creata da una CEV non memorizza una copia dell'AMI. Mantiene un puntatore all'AMI che hai usato per creare la CEV. L'AMI deve esistere affinché un'istanza database RDS Custom per SQL Server rimanga funzionale.

Limitazioni relative alle CEV per RDS Custom per SQL Server

Le seguenti restrizioni si applicano alle versioni del motore personalizzate con RDS Custom per SQL Server:

- Non è possibile eliminare una CEV se ad essa sono associate risorse, come istanze database o snapshot di database.
- Per creare un'istanza database RDS Custom per SQL Server, lo stato di una CEV deve essere `pending-validation`, `available`, `failed` o `validating`. Non è possibile creare un'istanza database RDS Custom per SQL Server utilizzando una CEV se lo stato della CEV è `incompatible-image-configuration`.
- Per modificare un'istanza database RDS Custom per SQL Server in modo che utilizzi una nuova CEV, lo stato della CEV deve essere `available`.
- Non è possibile creare un'AMI o una CEV da un'istanza database RDS Custom per SQL Server esistente.
- Non è possibile modificare una CEV esistente per utilizzare una AMI diversa. È tuttavia possibile modificare un'istanza database RDS Custom per SQL Server per utilizzare una CEV diversa. Per ulteriori informazioni, consulta Modifica di un'istanza database RDS Custom per SQL Server.
- La copia delle CEV tra regioni non è supportata.
- La copia delle CEV tra account non è supportata.
- Non è possibile recuperare o ripristinare una CEV dopo averla eliminata. Puoi tuttavia creare una nuova CEV dalla stessa AMI.
- Un'istanza database RDS Custom per SQL Server archivia i file del database SQL Server nell'unità D:\. L'AMI associata a una CEV deve archiviare i file del database di sistema Microsoft SQL Server nell'unità C:\.
- Un'istanza database RDS Custom per SQL Server mantiene le modifiche della configurazione apportate a SQL Server. Qualsiasi modifica della configurazione al sistema operativo su un'istanza

database RDS Custom per SQL Server in esecuzione creata da una CEV non viene mantenuta. Se è necessario apportare una modifica permanente della configurazione al sistema operativo e mantenerla come nuova configurazione di base, crea una nuova CEV e modifica l'istanza database per utilizzare la nuova CEV.

> **Important**
>
> La modifica di un'istanza database RDS Custom per SQL Server per utilizzare una nuova CEV è un'operazione offline. È possibile eseguire la modifica immediatamente o programmarla in modo che venga eseguita durante una finestra di manutenzione settimanale.

- Quando una CEV viene modificata, Amazon RDS non invia tali modifiche a nessuna istanza database RDS Custom per SQL Server associata. È necessario modificare ciascuna istanza database RDS Custom per SQL Server per utilizzare la CEV nuova o aggiornata. Per ulteriori informazioni, consulta Modifica di un'istanza database RDS Custom per SQL Server.
-

> **Important**
>
> Se un'AMI utilizzata da una CEV viene eliminata, qualsiasi modifica che potrebbe richiedere la sostituzione dell'host, ad esempio l'elaborazione della scala, avrà esito negativo. L'istanza database RDS Custom per SQL Server verrà quindi posizionata all'esterno del perimetro del supporto RDS. Ti consigliamo di evitare di eliminare qualsiasi AMI associata a una CEV.

Creazione di una CEV per RDS Custom per SQL Server

È possibile creare una versione del motore personalizzata (CEV) utilizzandoAWS Management Console o AWS CLI. Quindi è possibile usare la CEV per creare un'istanza RDS Custom per SQL Server.

Accertati che Amazon Machine Image (AMI) si trovi nello stesso account e Regione AWS della tua CEV. In caso contrario, il processo di creazione di un CEV non riesce.

Per ulteriori informazioni, consulta Creazione e connessione a un'istanza database per Amazon RDS Custom per SQL Server.

> **Important**
>
> I passaggi per creare una CEV sono gli stessi per le AMI create con SQL Server preinstallato e per quelle create utilizzando il modello Bring Your Own Media (BYOM).

Console

Per creare un CEV

1. Accedi alla AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel pannello di navigazione, scegliere Versioni motore personalizzate.

   La pagina Versioni motore personalizzate mostra tutti i CEV attualmente esistenti. Se non è stata creata alcuna CEV, la pagina è vuota.
3. Scegliere Creazione della versione del motore personalizzata.
4. Per Engine type (Tipo motore) scegli Microsoft SQL Server.
5. Per Edition, scegli l'edizione del motore DB che desideri utilizzare.
6. Per Major version (Versione principale) scegli la versione principale del motore installata sull'AMI.
7. In Version details (Dettagli versione), inserisci un nome valido in Custom engine version name (Nome della versione del motore personalizzato).

   Il formato del nome è *major-engine-version*.*minor-engine-version*.*customized_string*. Il nome utente può contenere solo 1–50 caratteri alfanumerici, punti e trattini (-, _). Ad esempio, è possibile inserire il nome **15.00.4249.2.my_cevtest**.

   Facoltativamente, inserisci una descrizione del tuo CEV.
8. Per Installation Media (Supporti di installazione), cerca o inserisci l'ID AMI da cui desideri creare la CEV.
9. Nella sezione Tags (Tag), aggiungi qualsiasi tag per identificare la CEV.
10. Scegliere Creazione della versione del motore personalizzata.

La pagina Versioni motore personalizzate viene visualizzata. La CEV viene mostrata con lo stato pending-validation (in attesa di convalida)

AWS CLI

Per creare un CEV utilizzandoAWS CLI, esegui il comando create-custom-db-engine-version.

Sono richieste le seguenti opzioni:

- `--engine`
- `--engine-version`
- `--image-id`

È anche possibile specificare le seguenti opzioni:

- `--description`
- `--region`
- `--tags`

L'esempio seguente crea un CEV denominato `15.00.4249.2.my_cevtest`. Assicurati che il nome della CEV inizi con il numero di versione principale del motore.

Example

PerLinux, omacOS: Unix

```
aws rds create-custom-db-engine-version \
    --engine custom-sqlserver-ee \
    --engine-version 15.00.4249.2.my_cevtest \
    --image-id ami-0r93cx31t5r596482 \
    --description "Custom SQL Server EE 15.00.4249.2 cev test"
```

Il seguente output parziale mostra il motore, i gruppi di parametri e altre informazioni.

```
"DBEngineVersions": [
    {
    "Engine": "custom-sqlserver-ee",
    "MajorEngineVersion": "15.00",
    "EngineVersion": "15.00.4249.2.my_cevtest",
    "DBEngineDescription": "Microsoft SQL Server Enterprise Edition for RDS Custom for
 SQL Server",
```

```
    "DBEngineVersionArn": "arn:aws:rds:us-east-1:<my-account-id>:cev:custom-sqlserver-
ee/15.00.4249.2.my_cevtest/a1234a1-123c-12rd-bre1-1234567890",
    "DBEngineVersionDescription": "Custom SQL Server EE 15.00.4249.2 cev test",

    "Image": [
        "ImageId": "ami-0r93cx31t5r596482",
        "Status": "pending-validation"
     ],
    "CreateTime": "2022-11-20T19:30:01.831000+00:00",
    "SupportsLogExportsToCloudwatchLogs": false,
    "SupportsReadReplica": false,
    "Status": "pending-validation",
    "SupportsParallelQuery": false,
    "SupportsGlobalDatabases": false,
    "TagList": []
    }
]
```

Se il processo di creazione di una CEV non riesce, RDS Custom per SQL Server genera `RDS-EVENT-0198` con il messaggio `Creation failed for custom engine version` *`major-engine-version.cev_name`*. Il messaggio include i dettagli sull'errore, ad esempio, l'evento stampa dei file mancanti. Per trovare idee per la risoluzione dei problemi relativi alla creazione di CEV, consulta Risoluzione degli errori della CEV per RDS Custom per SQL Server.

Creazione di un'istanza database RDS Custom per SQL Server da una CEV

Dopo aver creato correttamente una CEV, viene visualizzato lo stato della CEV `pending-validation`. È ora possibile creare una nuova istanza database di RDS Custom per SQL Server utilizzando la CEV. Per creare una nuova istanza database RDS Custom per SQL Server da una CEV, consulta Creazione di un'istanza database RDS Custom per SQL Server.

Ciclo di vita di una CEV

Il ciclo di vita della CEV include i seguenti stati.

Stato della CEV	Descrizione	Suggerimenti sulla risoluzione dei problemi	
`pending-validation`	È stata creata una CEV ed è in attesa della convalida	Se non ci sono attività esistenti, è necessario creare una nuova istanza database RDS Custom per SQL Server dalla CEV. Quando crei l'istanza database RDS Custom per	

Stato della CEV	Descrizione	Suggerimenti sulla risoluzione dei problemi	
	dell'AMI associata. Una CEV rimarrà nello stato `pending-validation` fino alla creazione di un'istanza database RDS Custom per SQL Server.	SQL Server, il sistema tenta di convalidare l'AMI associata per una CEV.	
`validating`	È in corso un'attività di creazione per l'istanza database RDS Custom per SQL Server basata su una nuova CEV. Quando crei l'istanza database RDS Custom per SQL Server, il sistema tenta di convalidare l'AMI associata di una CEV.	Attendi il completamento dell'attività di creazione dell'istanza database RDS Custom per SQL Server. È possibile utilizzare la console RDS EVENTS (EVENTI RDS) per esaminare i messaggi di evento dettagliati per la risoluzione dei problemi.	

Stato della CEV	Descrizione	Suggerimenti sulla risoluzione dei problemi
`available`	La CEV è stata convalidata. La CEV è nello stato `available` quando un'istanza database RDS Custom per SQL Server viene creata.	La CEV non richiede alcuna convalida aggiuntiva. Può essere utilizzata per creare nuove istanze database RDS Custom per SQL Server o per modificare quelle esistenti.
`inactive`	La CEV è stata impostata sullo stato inattivo.	Non è possibile creare o aggiornare un'istanza RDS Custom con questa CEV. Non è possibile ripristinare uno snapshot di database per creare una nuova istanza database RDS Custom con questa CEV. Per informazioni su come modificare lo stato in `ACTIVE`, consulta Modifica di una CEV per RDS Custom per SQL Server.
`failed`	La fase di creazione dell'istanza database non è riuscita per questa CEV prima della convalida dell'AMI. In alternativa, l'AMI sottostante utilizzata dalla CEV non è in uno stato disponibile.	Individua la causa principale per cui il sistema non è riuscito a creare l'istanza database. Visualizza il messaggio di errore dettagliato e prova a creare una nuova istanza database. Verifica che l'AMI sottostante utilizzata dalla CEV sia in uno stato disponibile.

Stato della CEV	Descrizione	Suggerimenti sulla risoluzione dei problemi	
`incompatible-image-configuration`	Si è verificato un errore durante la convalida dell'AMI.	Visualizza i dettagli tecnici dell'errore. Non è possibile tentare di convalidare nuovamente l'AMI con questa CEV. Prendi in considerazione i seguenti suggerimenti: • Assicurati che la CEV sia denominata in base al modello di denominazione richiesto della versione principale + versione secondaria + stringa personalizzata di SQL Server. • Assicurati che la versione di SQL Server nel nome CEV corrisponda alla versione fornita con l'AMI. • Assicurati che la versione della build del sistema operativo soddisfi la versione della build minima richiesta. • Assicurati che la versione principale del sistema operativo soddisfi la versione principale minima richiesta. Crea una nuova CEV utilizzando le informazioni corrette. Se necessario, crea una nuova istanza EC2 utilizzando un'AMI supportata ed esegui il processo Sysprep.	

Modifica di una CEV per RDS Custom per SQL Server

È possibile modificare un CEV tramite AWS Management Console o AWS CLI. È possibile modificare la descrizione CEV o il relativo stato di disponibilità. Il CEV ha uno dei seguenti valori di stato:

- `available` – È possibile utilizzare questo CEV per creare una nuova istanza database RDS Custom o aggiornare un'istanza database. Questo è lo stato predefinito per un CEV appena creato.

- `inactive` – Non è possibile creare o aggiornare un'istanza database RDS Custom con questa CEV. Non è possibile ripristinare una snapshot DB per creare una nuova istanza database RDS Custom con questo CEV.

È possibile modificare lo stato della CEV da `available` a `inactive` o da `inactive` a `available`. È possibile modificare lo stato su `INACTIVE` per impedire l'uso accidentale di una CEV o rendere nuovamente idoneo l'uso di una CEV sospesa.

Console

Per modificare un CEV

1. Accedi alla AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel pannello di navigazione, scegliere Versioni motore personalizzate.
3. Scegliere un CEV di cui si desidera modificare la descrizione o lo stato.
4. Per Operazioni, scegli Modifica.
5. Effettua una qualsiasi delle seguenti modifiche:
   - Per CEV status settings (Impostazioni dello stato del CEV) scegliere un nuovo stato di disponibilità.
   - In Version description (Descrizione versione), inserire una nuova descrizione.
6. Scegliere Modify CEV (Modifica CEV).

   Se il CEV è in uso, la console visualizza You can't modify the CEV status (Non puoi modificare lo stato CEV). Risolvi i problemi e riprova.

La pagina Versioni motore personalizzate viene visualizzata.

AWS CLI

Per modificare un CEV utilizzandoAWS CLI, eseguire il comando modify-custom-db-engine-version. È possibile trovare i CEV da modificare eseguendo il comando. describe-db-engine-versions

Sono richieste le seguenti opzioni:

- `--engine`

- `--engine-version` *`cev`*, dove *`cev`* è il nome della versione del motore personalizzata che si desidera modificare
- `--status` *`status`*, dove *`status`* è lo stato di disponibilità che si desidera assegnare al CEV

L'esempio seguente cambia un CEV denominato `15.00.4249.2.my_cevtest` dal suo stato attuale a `inactive`.

Example

PerLinux, omacOS: Unix

```
aws rds modify-custom-db-engine-version \
    --engine custom-sqlserver-ee \
    --engine-version 15.00.4249.2.my_cevtest \
    --status inactive
```

Per Windows:

```
aws rds modify-custom-db-engine-version ^
    --engine custom-sqlserver-ee ^
    --engine-version 15.00.4249.2.my_cevtest ^
    --status inactive
```

Modifica di un'istanza database RDS Custom per SQL Server per usare una nuova CEV

È possibile modificare un'istanza database RDS Custom per SQL Server esistente per usare una CEV diversa. Le modifiche che puoi apportare sono:

- Modifica della CEV
- Modifica della classe di istanza database
- Modifica del periodo di conservazione del backup e della finestra di backup
- Modifica della finestra di manutenzione

Console

Modifica di un'istanza database RDS Custom per SQL Server

1. Accedi alla AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.

2. Nel riquadro di navigazione, scegliere Databases (Database).
3. Scegliere l'istanza database che si vuole modificare.
4. Scegliere Modify (Modifica).
5. Eseguire le seguenti modifiche secondo necessità:
    a. Per DB engine version (Versione del motore di database), scegli una CEV diversa.
    b. Modificare il valore per DB instance class (Classe istanza database). Per le classi supportate, consultare Supporto delle classi di istanza database per RDS Custom for SQL Server.
    c. Modificare il valore per Backup retention period (Periodo di retention dei backup).
    d. Per Backup window (Finestra di backup), imposta i valori per Ora di inizio e Durata.
    e. Per Finestra di manutenzione istanza database, imposta i valori per Start day (Avvia giorno), Start time (Ora di inizio) e Duration (Durata).
6. Scegliere Continue (Continua).
7. Scegliere Apply immediately (Applica immediatamente) o Apply during the next scheduled maintenance window (Applica durante la prossima finestra di manutenzione pianificata).
8. Scegliere Modify DB Instance (Modifica istanza database).

> **Note**
>
> Quando si modifica un'istanza database da una CEV a un'altra, ad esempio quando si aggiorna una versione secondaria, i database di sistema SQL Server, inclusi i relativi dati e configurazioni, vengono mantenuti dall'attuale istanza database RDS Custom per SQL Server.

AWS CLI

Per modificare un'istanza DB in modo che utilizzi un CEV diverso utilizzando ilAWS CLI, esegui il modify-db-instancecomando.

Sono richieste le seguenti opzioni:

- `--db-instance-identifier`
- `--engine-version` *`cev`*, dove *`cev`* è il nome della versione del motore personalizzata che si desidera venga modificata dall'istanza database.

L'esempio seguente modifica un'istanza database denominata `my-cev-db-instance` per utilizzare una CEV denominata `15.00.4249.2.my_cevtest_new` e applica immediatamente la modifica.

Example

Per LinuxmacOS, oUnix:

```
aws rds modify-db-instance \
    --db-instance-identifier my-cev-db-instance \
    --engine-version 15.00.4249.2.my_cevtest_new \
    --apply-immediately
```

Per Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier my-cev-db-instance ^
    --engine-version 15.00.4249.2.my_cevtest_new ^
    --apply-immediately
```

Visualizzazione dei dettagli della CEV per Amazon RDS Custom per SQL Server

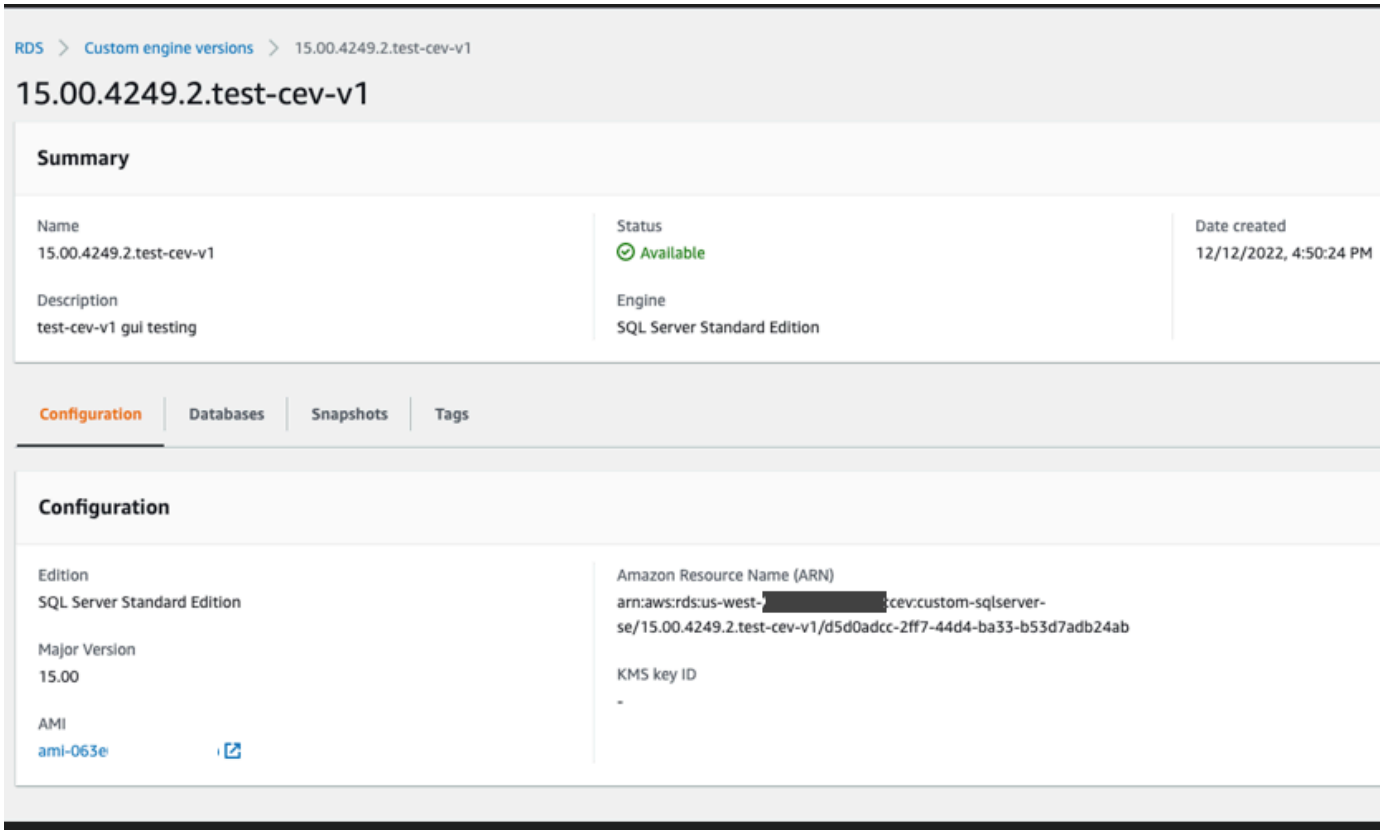
Puoi visualizzare i dettagli della CEV utilizzando la AWS Management Console o AWS CLI.

Console

Per visualizzare i dettagli della CEV

1. Accedi alla AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel pannello di navigazione, scegliere Versioni motore personalizzate.

   La pagina Versioni motore personalizzate mostra tutti i CEV attualmente esistenti. Se non è stato creato alcun CEV, la pagina è vuota.
3. Seleziona il nome della CEV che vuoi visualizzare.
4. Scegli Configuration (Configurazione) per visualizzare i dettagli.



AWS CLI

Per informazioni su una CEV usando AWS CLI, esegui il comando describe-db-engine-versions.

È anche possibile specificare le seguenti opzioni:

- `--include-all`, per visualizzare tutte le CEV con qualsiasi stato del ciclo di vita. Senza l'opzione `--include-all` vengono restituite solo le CEV nello stato del ciclo di vita `available`.

```
aws rds describe-db-engine-versions --engine custom-sqlserver-ee --engine-version
 15.00.4249.2.my_cevtest --include-all
{
    "DBEngineVersions": [
        {
            "Engine": "custom-sqlserver-ee",
            "MajorEngineVersion": "15.00",
            "EngineVersion": "15.00.4249.2.my_cevtest",
            "DBParameterGroupFamily": "custom-sqlserver-ee-15.0",
            "DBEngineDescription": "Microsoft SQL Server Enterprise Edition for custom
 RDS",
```

```
            "DBEngineVersionArn": "arn:aws:rds:us-east-1:{my-account-id}:cev:custom-
sqlserver-ee/15.00.4249.2.my_cevtest/a1234a1-123c-12rd-bre1-1234567890",
            "DBEngineVersionDescription": "Custom SQL Server EE 15.00.4249.2 cev test",
            "Image": {
                "ImageId": "ami-0r93cx31t5r596482",
                "Status": "pending-validation"
            },
            "DBEngineMediaType": "AWS Provided",
            "CreateTime": "2022-11-20T19:30:01.831000+00:00",
            "ValidUpgradeTarget": [],
            "SupportsLogExportsToCloudwatchLogs": false,
            "SupportsReadReplica": false,
            "SupportedFeatureNames": [],
            "Status": "pending-validation",
            "SupportsParallelQuery": false,
            "SupportsGlobalDatabases": false,
            "TagList": [],
            "SupportsBabelfish": false
        }
    ]
}
```

Puoi utilizzare i filtri per visualizzare le CEV con un determinato stato del ciclo di vita. Ad esempio, per visualizzare le CEV con lo stato del ciclo di vita `pending-validation`, `available` o `failed`:

```
aws rds describe-db-engine-versions engine custom-sqlserver-ee
                region us-west-2 include-all query 'DBEngineVersions[?Status ==
 pending-validation ||
                Status == available || Status == failed]'
```

Eliminazione di una CEV per RDS Custom per SQL Server

È possibile eliminare un CEV tramite AWS Management Console o AWS CLI. In genere l'attività richiede pochi minuti.

Prima di eliminare una CEV, assicurati che non venga utilizzata da nessuno dei seguenti elementi:

- Un'istanza database RDS Custom
- Una snapshot di un'istanza database RDS Custom
- Backup automatico dell'istanza database RDS Custom

Console

Per eliminare un CEV

1. Accedi alla AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel pannello di navigazione, scegliere Versioni motore personalizzate.
3. Scegliere un CEV di cui si desidera eliminare la descrizione o lo stato.
4. In Actions (Azioni), scegliere Delete (Elimina).

   Viene visualizzata la finestra di dialogo Delete ***cev_name***? (Elimina cev_name?).
5. Immettere **`delete me`**, quindi scegliere Delete (Elimina).

   Nella pagina Versioni motore personalizzate, il banner mostra che il tuo CEV è stato eliminato.

AWS CLI

Per eliminare un CEV utilizzandoAWS CLI, eseguire il comando delete-custom-db-engine-version.

Sono richieste le seguenti opzioni:

- `--engine custom-sqlserver-ee`
- `--engine-version` *`cev`*, in cui *`cev`* è il nome della versione del motore personalizzata da eliminare

L'esempio seguente elimina un CEV denominato `15.00.4249.2.my_cevtest`.

Example

PerLinux, omacOS: Unix

```
aws rds delete-custom-db-engine-version \
    --engine custom-sqlserver-ee \
    --engine-version 15.00.4249.2.my_cevtest
```

Per Windows:

```
aws rds delete-custom-db-engine-version ^
    --engine custom-sqlserver-ee ^
```

```
    --engine-version 15.00.4249.2.my_cevtest
```

Creazione e connessione a un'istanza database per Amazon RDS Custom per SQL Server

Puoi creare un'istanza DB personalizzata RDS e poi connetterti ad essa utilizzando AWS Systems Manager il nostro Remote Desktop Protocol (RDP).

> **Important**
>
> Assicurati di completare le attività indicate su Configurazione dell'ambiente per Amazon RDS Custom per SQL Server prima di poter creare o connettersi a RDS Custom per un'istanza database di SQL Server.
> È possibile contrassegnare le istanze database RDS Custom quando le crei, ma non creare o modificare il tag `AWSRDSCustom` richiesto per l'automazione di RDS Custom. Per ulteriori informazioni, consulta Assegnazione di tag alle risorse RDS Custom for SQL Server.
> La prima volta che crei un'istanza database RDS Custom per SQL Server, potresti ricevere il seguente errore: Il ruolo collegato ai servizi è nel processo di creazione. Riprova più tardi. In questo caso, attendere alcuni minuti e riprovare a creare l'istanza database.

Argomenti

Creazione di un'istanza database RDS Custom per SQL Server

Crea un'istanza database Amazon RDS Custom per SQL Server utilizzando AWS Management Console o. AWS CLI La procedura è simile alla procedura per la creazione di un'istanza database Amazon RDS.

Per ulteriori informazioni, consulta Creazione di un'istanza database Amazon RDS.

Console

Per creare un'istanza database RDS Custom per SQL Server

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, scegliere Databases (Database).
3. Scegliere Create database (Crea database).
4. Scegliere Standard Create (Creazione standard) come metodo di creazione del database.
5. In Engine options (Opzioni motore), selezionare Microsoft SQL Server per il tipo di motore.
6. Per il tipo di gestione del database, selezionare Amazon RDS Custom.
7. Nella sezione Edition (Edizione), scegliere l'edizione del motore di database che desideri utilizzare.
8. (Facoltativo) Se intendi creare l'istanza database da una CEV, seleziona la casella di controllo Use custom engine version (CEV) (Usa la versione del motore personalizzata (CEV). Seleziona la CEV nell'elenco a discesa.
9. Per la versione del database, mantieni la versione con valore predefinito.
10. Per Templates (Modelli), scegliere Production (Produzione).
11. Nella sezione Settings (Impostazioni) inserire un nuovo nome per DB instance identifier (Identificatore istanze DB).
12. Per inserire la password principale, procedere come segue:

    a. Nella sezione Settings (Impostazioni), aprire Credential Settings (Impostazioni credenziali).
    b. Deselezionare la casella di controllo Auto generate a password (Genera automaticamente una password).
    c. Modificare il valore Master username (Nome utente principale) e inserire la stessa password in Master password (Password principale) e Confirm password (Conferma password).

    Per impostazione predefinita, la nuova istanza database RDS Custom utilizza una password generata automaticamente per l'utente principale.
13. Nella sezione della dimensione dell'istanza del database, selezionare un valore per DB instance class (Classe istanza database).

    Per le classi supportate, consultare Supporto delle classi di istanza database per RDS Custom for SQL Server.

14. Scegliere le impostazioni Storage.
15. Per la sicurezzaRDS Custom, procedere come segue:

    a. Per il profilo di istanza IAM, sono disponibili due opzioni per scegliere il profilo di istanza per l'istanza DB RDS Custom for SQL Server.

        1. Scegli Crea un nuovo profilo di istanza e fornisci un suffisso per il nome del profilo di istanza. Per ulteriori informazioni, consulta Creazione automatica del profilo di istanza utilizzando il AWS Management Console.
        2. Scegli un profilo di istanza esistente. Dall'elenco a discesa, scegli il profilo dell'istanza che inizia con. `AWSRDSCustom`

    b. Per Encryption (Crittografia), selezionare Enter a key ARN (Inserisci l'ARN della chiave) per elencare le chiavi AWS KMS disponibili. Scegliere quindi la propria chiave dall'elenco.

        È richiesta una AWS KMS chiave per RDS Custom. Per ulteriori informazioni, consulta Assicurati di disporre di una chiave di crittografia simmetrica AWS KMS.

16. Per le restanti sezioni, specifica le impostazioni dell'istanza database RDS Custom preferite. Per informazioni su ciascuna impostazione, consulta Impostazioni per istanze database. Le impostazioni seguenti non appaiono nella console e non sono supportate:

    - Caratteristiche processore
    - Storage autoscaling (Auto Scaling dello storage)
    - Disponibilità e durabilità
    - Opzione Password and Kerberos authentication (Password e autenticazione Kerberos) in Database authentication (Autenticazione del database) (solo Autenticazione password è supportata)
    - Il gruppo Opzioni database in Configurazione aggiuntiva
    - Approfondimenti sulle prestazioni
    - Log exports (Esportazioni log)
    - Abilita aggiornamento automatico della versione secondaria
    - Deletion protection (Protezione da eliminazione)

    Backup retention period (Periodo di retention dei backup) è supportato, ma non puoi scegliere 0 giorni.

17. Scegliere Crea database.

Il pulsante View credential details (Vedi dettagli delle credenziali) viene visualizzato sulla pagina Database.

Per vedere nome utente e password per l'istanza database RDS Custom, seleziona View credential details (Vedi dettagli delle credenziali).

Per connetterti all'istanza database come utente principale, utilizza il nome utente e la password visualizzati.

> **Important**
>
> Non potrai visualizzare di nuovo la password dell'utente principale. Se non la registri, potresti doverla modificare. Per modificare la password dell'utente principale dopo che l'istanza database RDS Custom è disponibile, modificare l'istanza database. Per ulteriori informazioni sulla modifica di un'istanza database, consulta Gestione di un'istanza database per Amazon RDS Custom for SQL Server.

18. Scegliere Database per visualizzare l'elenco delle istanze database RDS Custom.
19. Scegliere l'istanza database RDS Custom appena creata.

    Nella console RDS vengono visualizzati i dettagli per la nuova istanza database RDS Custom:

    - L'istanza database RDS Custom rimane nello stato creating (creazione in corso) fino a quando non è stata creata e non è pronta per l'uso. Quando lo stato cambia in available (disponibile) è possibile connettersi all'istanza database. A seconda della classe di istanza e dello storage allocato, potrebbero trascorrere diversi minuti prima che la nuova istanza database sia disponibile.
    - Ruolo ha il valore Istanza (RDS Custom).
    - Modalità di automazione RDS Custom ha il valore Automazione completa. Questa impostazione indica che l'istanza database fornisce il monitoraggio automatico e il ripristino dell'istanza.

AWS CLI

È possibile creare un'istanza DB personalizzata RDS utilizzando il create-db-instance AWS CLI comando.

Sono richieste le seguenti opzioni:

- `--db-instance-identifier`
- `--db-instance-class` (per l'elenco delle classi di istanza supportate, vedere Supporto delle classi di istanza database per RDS Custom for SQL Server)
- `--engine` (`custom-sqlserver-ee`, `custom-sqlserver-se` o `custom-sqlserver-web`)
- `--kms-key-id`
- `--custom-iam-instance-profile`

Nell'esempio seguente viene creata un'istanza database RDS Custom per SQL Server denominata `my-custom-instance`. Il periodo di retention dei backup è di 3 giorni.

> **Note**
>
> Per creare un'istanza database da una versione del motore personalizzata (CEV), specifica un nome CEV esistente nel parametro `--engine-version`. Ad esempio, `--engine-version 15.00.4249.2.my_cevtest`

Example

Per LinuxmacOS, oUnix:

```
aws rds create-db-instance \
    --engine custom-sqlserver-ee \
    --engine-version 15.00.4073.23.v1 \
    --db-instance-identifier my-custom-instance \
    --db-instance-class db.m5.xlarge \
    --allocated-storage 20 \
    --db-subnet-group mydbsubnetgroup \
    --master-username myuser \
    --master-user-password mypassword \
    --backup-retention-period 3 \
    --no-multi-az \
    --port 8200 \
    --kms-key-id mykmskey \
    --custom-iam-instance-profile AWSRDSCustomInstanceProfileForRdsCustomInstance
```

Per Windows:

```
aws rds create-db-instance ^
```

```
    --engine custom-sqlserver-ee ^
    --engine-version 15.00.4073.23.v1 ^
    --db-instance-identifier my-custom-instance ^
    --db-instance-class db.m5.xlarge ^
    --allocated-storage 20 ^
    --db-subnet-group mydbsubnetgroup ^
    --master-username myuser ^
    --master-user-password mypassword ^
    --backup-retention-period 3 ^
    --no-multi-az ^
    --port 8200 ^
    --kms-key-id mykmskey ^
    --custom-iam-instance-profile AWSRDSCustomInstanceProfileForRdsCustomInstance
```

> **Note**
>
> Specifica una password diversa dal prompt mostrato qui come best practice per la sicurezza.

Ottenere informazioni sull'istanza tramite il comando `describe-db-instances`.

```
aws rds describe-db-instances --db-instance-identifier my-custom-instance
```

Il seguente output parziale mostra il motore, i gruppi di parametri e altre informazioni.

```
{
    "DBInstances": [
        {
            "PendingModifiedValues": {},
            "Engine": "custom-sqlserver-ee",
            "MultiAZ": false,
            "DBSecurityGroups": [],
            "DBParameterGroups": [
                {
                    "DBParameterGroupName": "default.custom-sqlserver-ee-15",
                    "ParameterApplyStatus": "in-sync"
                }
            ],
            "AutomationMode": "full",
            "DBInstanceIdentifier": "my-custom-instance",
            "TagList": []
        }
```

```
    ]
}
```

Ruolo collegato ai servizi RDS Custom

Un ruolo collegato al servizio offre ad Amazon RDS Custom l'accesso alle risorse del tuo. Account AWS Ciò rende più semplice l'utilizzo di RDS Custom perché non si devono aggiungere manualmente le autorizzazioni necessarie. RDS Custom definisce le autorizzazioni dei ruoli associato ai servizi e, salvo diversamente definito, solo RDS Custom può assumere tali ruoli. Le autorizzazioni definite includono la policy di attendibilità e la policy delle autorizzazioni che non può essere collegata a nessun'altra entità IAM.

Quando crei un'istanza DB personalizzata RDS, vengono creati e utilizzati sia i ruoli collegati ai servizi (se non già esistenti) Amazon RDS che RDS Custom. Per ulteriori informazioni, consulta Utilizzo di ruoli collegati ai servizi per Amazon RDS.

La prima volta che crei un'istanza database RDS Custom per SQL Server, potresti ricevere il seguente errore: Il ruolo collegato ai servizi è nel processo di creazione. Riprova più tardi. In questo caso, attendere alcuni minuti e riprovare a creare l'istanza database.

Connessione alla tua istanza DB personalizzata RDS tramite AWS Systems Manager

Dopo aver creato l'istanza database RDS Custom, è possibile connettersi ad essa utilizzando AWS Systems Manager Session Manager. Session Manager è una funzionalità Systems Manager che puoi utilizzare per gestire le istanze Amazon EC2 attraverso una shell basata su browser o tramite AWS CLI. Per ulteriori informazioni, consulta AWS Systems Manager Session Manager.

Console

Per connettersi all'istanza database utilizzando Session Manager

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, scegliere Databases (Database) e quindi scegliere l'istanza database RDS Custom a cui desideri connetterti.
3. Scegliere Configuration (Configurazione).
4. Annota il valore Resource ID (Risorsa ID)per l'istanza database. Ad esempio, l'ID risorsa potrebbe essere `db-ABCDEFGHIJKLMNOPQRS0123456`.
5. Apri la console Amazon EC2 all'indirizzo https://console.aws.amazon.com/ec2/.

6. Nel riquadro di navigazione, seleziona Istanze.
7. Cerca il nome dell'istanza EC2, quindi scegli l'ID istanza associato con esso. Ad esempio, l'istanza ID potrebbe essere `i-abcdefghijklm01234`.
8. Scegli Connetti.
9. Scegli Session Manager.
10. Scegli Connetti.

    Si apre una finestra per la sessione.

AWS CLI

Puoi connettere l'istanza database RDS Custom tramite AWS CLI. Questa tecnica richiede il plugin Session Manager per AWS CLI. Per informazioni su come installare il plugin, consultare Installare il plugin di Session Manager per AWS CLI.

Per trovare l'ID della risorsa DB dell'istanza database RDS Custom, utilizzare `describe-db-instances`.

```
aws rds describe-db-instances \
    --query 'DBInstances[*].[DBInstanceIdentifier,DbiResourceId]' \
    --output text
```

Il seguente output di esempio mostra l'ID della risorsa per l'istanza RDS Custom. Il prefisso è `db-`.

```
db-ABCDEFGHIJKLMNOPQRS0123456
```

Per trovare l'ID dell'istanza EC2 della tua istanza database, utilizzare `aws ec2 describe-instances`. Nell'esempio seguente viene utilizzato `db-ABCDEFGHIJKLMNOPQRS0123456` per l'ID risorsa.

```
aws ec2 describe-instances \
    --filters "Name=tag:Name,Values=db-ABCDEFGHIJKLMNOPQRS0123456" \
    --output text \
    --query 'Reservations[*].Instances[*].InstanceId'
```

L'output di esempio seguente mostra l'ID dell'istanza EC2.

```
i-abcdefghijklm01234
```

Utilizzo del comando `aws ssm start-session`, che fornisce l'ID istanza EC2 nel parametro `--target`.

```
aws ssm start-session --target "i-abcdefghijklm01234"
```

Se l'operazione riesce, la connessione sarà simile al seguente.

```
Starting session with SessionId: yourid-abcdefghijklm1234
[ssm-user@ip-123-45-67-89 bin]$
```

Connessione all'istanza database RDS Custom tramite RDP

Dopo aver creato l'istanza database RDS Custom, è possibile connettersi a questa istanza utilizzando un client RDP. La procedura è la stessa di quella per la connessione a un'istanza Amazon EC2. Per ulteriori informazioni, consulta Connessione all'istanza Windows.

Per connettersi all'istanza database, è necessaria la coppia di chiavi associata all'istanza. RDS Custom crea per te la coppia di chiavi. Il nome della coppia utilizza il prefisso `do-not-delete-rds-custom-`*`DBInstanceIdentifier`*. AWS Secrets Manager memorizza la tua chiave privata come un segreto.

Completa l'attività nei passaggi seguenti:

1. Configurare l'istanza database per consentire connessioni RDP.
2. Recupera la tua chiave segreta.
3. Connettersi all'istanza EC2 utilizzando l'utility RDP.

Configurare l'istanza database per consentire connessioni RDP

Per consentire le connessioni RDP, configurare il gruppo di sicurezza VPC e impostare una regola firewall sull'host.

Configura il tuo gruppo di sicurezza VPC

Assicurati che il gruppo di sicurezza VPC associato all'istanza database consenta le connessioni in ingresso sulla porta 3389 per Transmission Control Protocol (TCP). Per informazioni su come configurare il gruppo di sicurezza VPC, consultare Configura il tuo gruppo di sicurezza VPC.

Impostare la regola del firewall sull'host

Per consentire connessioni in ingresso sulla porta 3389 per TCP, impostare una regola firewall sull'host. Gli esempi seguenti mostrano come fare.

Ti consigliamo di utilizzare il valore `-Profile` specifico: `Public`, `Private` oppure `Domain`. L'utilizzo di `Any` si riferisce a tutti e tre i valori. È inoltre possibile specificare una combinazione di valori separati da una virgola. Per ulteriori informazioni sull'impostazione delle regole del firewall, vedere Set- NetFirewallRule nella documentazione Microsoft.

Per utilizzare Systems Manager Session Manager per configurare una regola del firewall

1. Collegarsi a Session Manager come mostrato in Connessione alla tua istanza DB personalizzata RDS tramite AWS Systems Manager.
2. Esegui il comando seguente.

```
Set-NetFirewallRule -DisplayName "Remote Desktop - User Mode (TCP-In)" -Direction
 Inbound -LocalAddress Any -Profile Any
```

Per utilizzare i comandi CLI Systems Manager per configurare una regola del firewall

1. Utilizzare il comando seguente per aprire RDP sull'host.

```
OPEN_RDP_COMMAND_ID=$(aws ssm send-command --region $AWS_REGION \
    --instance-ids $RDS_CUSTOM_INSTANCE_EC2_ID \
    --document-name "AWS-RunPowerShellScript" \
    --parameters '{"commands":["Set-NetFirewallRule -DisplayName \"Remote Desktop -
 User Mode (TCP-In)\" -Direction Inbound -LocalAddress Any -Profile Any"]}' \
    --comment "Open RDP port" | jq -r ".Command.CommandId")
```

2. Utilizzare l'ID comando restituito nell'output per ottenere lo stato del comando precedente. Per utilizzare la query seguente per restituire l'ID comando, assicurarsi che sia installato il plug-in jq.

```
aws ssm list-commands \
    --region $AWS_REGION \
    --command-id $OPEN_RDP_COMMAND_ID
```

Recupera la tua chiave segreta

Recupera la tua chiave segreta utilizzando uno AWS Management Console o il AWS CLI.

Console

Per recuperare la chiave segreta

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, scegliere Databases (Database) e quindi scegliere l'istanza database RDS Custom a cui desideri connetterti.
3. Scegli la scheda Configurazione.
4. Annota l'ID dell'istanza database per l'istanza database, ad esempio *`my-custom-instance`*.
5. Apri la console Amazon EC2 all'indirizzo https://console.aws.amazon.com/ec2/.
6. Nel riquadro di navigazione, seleziona Istanze.
7. Cerca il nome dell'istanza EC2, quindi scegli l'ID istanza associato con esso.

   In questo esempio, l'ID dell'istanza è `i-abcdefghijklm01234`.
8. In Details (Dettagli), trovare Key pair name (Nome della coppia di chiavi). Il nome della coppia include l'identificatore di database. In questo esempio, il nome della coppia è `do-not-delete-rds-custom-`*`my-custom-instance`*`-`*`0d726c`*.
9. Nel riepilogo dell'istanza, trovare Public IPv4 DNS (DNS IPv4 pubblico). Ad esempio, il DNS pubblico potrebbe essere `ec2-12-345-678-901.us-east-2.compute.amazonaws.com`.
10. Apri la AWS Secrets Manager console all'indirizzo https://console.aws.amazon.com/secretsmanager/.
11. Scegliere il segreto che ha lo stesso nome della tua coppia di chiavi.
12. Scegli Retrieve secret value (Recupera il valore del segreto).

AWS CLI

Per recuperare la chiave privata

1. Ottieni l'elenco delle istanze database RDS Custom richiamando il comando `aws rds describe-db-instances`.

```
aws rds describe-db-instances \
```

```
    --query 'DBInstances[*].[DBInstanceIdentifier,DbiResourceId]' \
    --output text
```

2. Scegliere l'identificatore dell'istanza database dall'output di esempio, ad esempio `do-not-delete-rds-custom-`*`my-custom-instance`*.
3. Trova l'ID dell'istanza EC2 della tua istanza database richiamando il comando `aws ec2 describe-instances`. Nell'esempio seguente viene utilizzato il nome dell'istanza EC2 per descrivere l'istanza database.

```
aws ec2 describe-instances \
    --filters "Name=tag:Name,Values=do-not-delete-rds-custom-my-custom-instance" \
    --output text \
    --query 'Reservations[*].Instances[*].InstanceId'
```

L'output di esempio seguente mostra l'ID dell'istanza EC2.

```
i-abcdefghijklm01234
```

4. Trovare il nome della chiave specificando l'ID istanza EC2, come illustrato nell'esempio seguente.

```
aws ec2 describe-instances \
    --instance-ids i-abcdefghijklm01234 \
    --output text \
    --query 'Reservations[*].Instances[*].KeyName'
```

Il seguente output di esempio mostra il nome della chiave, che utilizza il prefisso `do-not-delete-rds-custom-`*`DBInstanceIdentifier`*.

```
do-not-delete-rds-custom-my-custom-instance-0d726c
```

Connettersi all'istanza EC2 utilizzando l’utility RDP

Seguire la procedura in Connessione a un'istanza Windows tramite RDP nella Guida per l'utente di Amazon EC2 per le istanze Windows. Questa procedura presuppone che sia stato creato un file .pem con che contiene la tua chiave privata.

Gestione di un'istanza database per Amazon RDS Custom for SQL Server

Amazon RDS Custom for SQL Server supporta un sottoinsieme delle normali attività di gestione per le istanze database Amazon RDS. Di seguito puoi trovare le istruzioni per le attività di gestione RDS Custom for SQL Server supportate utilizzando la AWS Management Console e AWS CLI.

Argomenti

Sospensione e ripristino dell'automazione RDS Custom

RDS Custom fornisce automaticamente il monitoraggio e il ripristino delle istanze per un'istanza database RDS Custom for SQL Server. Per personalizzare l'istanza, procedi nel seguente modo:

1. Sospendi l'automazione RDS Custom per un periodo specificato. La pausa garantisce che le personalizzazioni non interferiscano con l'automazione di RDS Custom.
2. Personalizza l'istanza database RDS Custom for SQL Server secondo le necessità.
3. Esegui una delle operazioni seguenti:
   - Riprendi l'automazione manualmente.
   - Attendi che il periodo di pausa finisca. In questo caso, RDS Custom riprende automaticamente il monitoraggio e il ripristino delle istanze.

> **Important**
>
> La sospensione e la ripresa dell'automazione sono le uniche attività di automazione supportate durante la modifica di un'istanza database RDS Custom for SQL Server.

Console

Per sospendere o riprendere l'automazione RDS Custom

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel pannello di navigazione, scegliere Databases (Database) e selezionare l'istanza database RDS Custom da modificare.
3. Scegliere Modify (Modifica). Viene visualizzata la pagina Modify DB Instance (Modifica istanza database).
4. Per Modalità di automazione RDS Custom, scegliere una delle seguenti opzioni:
   - Paused (In pausa) sospende il monitoraggio e il ripristino dell'istanza per l'istanza database RDS Custom. Inserire la durata di pausa desiderata (in minuti) Durata della modalità di automazione. Il valore minimo è 60 minuti (predefinito). Il valore massimo è 1.440 minuti.
   - Automazione completa riprende l'automazione.
5. Scegliere Continue (Continua) per controllare il riepilogo delle modifiche.

   Un messaggio indica che RDS Custom applicherà immediatamente le modifiche.
6. Se le modifiche sono corrette, selezionare Modify DB Instance (Modifica istanza database). Oppure scegliere Back (Indietro) per cambiare le modifiche o Cancel (Annulla) per annullare le modifiche.

   Nella console RDS vengono visualizzati i dettagli per la modifica. Se hai interrotto l'automazione, lo Stato della tua istanza database RDS Custom indica Automation paused (Sospensione dell'automazione).
7. (Opzionale) Nel pannello di navigazione, scegliere Databases (Database), quindi scegliere un'istanza database RDS Custom.

   Nel pannello Summary (Riepilogo), la Modalità di automazione RDS Custom indica lo stato dell'automazione. Se l'automazione è sospesa, il valore è In pausa. L'automazione riprende in ***num*** minuti.

AWS CLI

Per mettere in pausa o riprendere l'automazione RDS Custom, usa il comando. `modify-db-instance` AWS CLI Identificare l'istanza database utilizzando il parametro richiesto `--db-instance-identifier`. Controllare la modalità di automazione con i seguenti parametri:

- `--automation-mode` specifica lo stato di pausa dell'istanza database. I valori validi sono `all-paused`, che mette in pausa l'automazione e `full`, che la riprende.
- `--resume-full-automation-mode-minutes` specifica la durata della pausa. Il valore predefinito è di 60 minuti.

> **Note**
>
> Indipendentemente dal fatto che tu specifichi `--no-apply-immediately` o `--apply-immediately`, RDS Custom applica le modifiche in modo asincrono il prima possibile.

Nella risposta al comando, `ResumeFullAutomationModeTime` indica l'orario di ripristino come timestamp UTC. Quando la modalità di automazione è `all-paused`, è possibile utilizzare `modify-db-instance` per riprendere la modalità di automazione o prolungare il periodo di pausa. Non sono supportate altre opzioni `modify-db-instance`.

L'esempio seguente sospende l'automazione per `my-custom-instance` per 90 minuti.

Example

PerLinux, o: macOS Unix

```
aws rds modify-db-instance \
    --db-instance-identifier my-custom-instance \
    --automation-mode all-paused \
    --resume-full-automation-mode-minutes 90
```

Per Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier my-custom-instance ^
    --automation-mode all-paused ^
    --resume-full-automation-mode-minutes 90
```

L'esempio seguente estende la durata della pausa di altri 30 minuti. I 30 minuti vengono aggiunti all'orario di origine mostrato in `ResumeFullAutomationModeTime`.

Example

Per LinuxmacOS, oUnix:

```
aws rds modify-db-instance \
    --db-instance-identifier my-custom-instance \
    --automation-mode all-paused \
    --resume-full-automation-mode-minutes 30
```

Per Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier my-custom-instance ^
    --automation-mode all-paused ^
    --resume-full-automation-mode-minutes 30
```

L'esempio seguente riprende l'automazione completa per `my-custom-instance`.

Example

Per LinuxmacOS, oUnix:

```
aws rds modify-db-instance \
    --db-instance-identifier my-custom-instance \
    --automation-mode full \
```

Per Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier my-custom-instance ^
    --automation-mode full
```

Nel seguente output di esempio parziale, il valore `AutomationMode` in attesa è `full`.

```
{
    "DBInstance": {
        "PubliclyAccessible": true,
        "MasterUsername": "admin",
        "MonitoringInterval": 0,
```

```
        "LicenseModel": "bring-your-own-license",
        "VpcSecurityGroups": [
            {
                "Status": "active",
                "VpcSecurityGroupId": "0123456789abcdefg"
            }
        ],
        "InstanceCreateTime": "2020-11-07T19:50:06.193Z",
        "CopyTagsToSnapshot": false,
        "OptionGroupMemberships": [
            {
                "Status": "in-sync",
                "OptionGroupName": "default:custom-oracle-ee-19"
            }
        ],
        "PendingModifiedValues": {
            "AutomationMode": "full"
        },
        "Engine": "custom-oracle-ee",
        "MultiAZ": false,
        "DBSecurityGroups": [],
        "DBParameterGroups": [
            {
                "DBParameterGroupName": "default.custom-oracle-ee-19",
                "ParameterApplyStatus": "in-sync"
            }
        ],
        ...
        "ReadReplicaDBInstanceIdentifiers": [],
        "AllocatedStorage": 250,
        "DBInstanceArn": "arn:aws:rds:us-west-2:012345678912:db:my-custom-instance",
        "BackupRetentionPeriod": 3,
        "DBName": "ORCL",
        "PreferredMaintenanceWindow": "fri:10:56-fri:11:26",
        "Endpoint": {
            "HostedZoneId": "ABCDEFGHIJKLMNO",
            "Port": 8200,
            "Address": "my-custom-instance.abcdefghijk.us-west-2.rds.amazonaws.com"
        },
        "DBInstanceStatus": "automation-paused",
        "IAMDatabaseAuthenticationEnabled": false,
        "AutomationMode": "all-paused",
        "EngineVersion": "19.my_cev1",
        "DeletionProtection": false,
```

```
        "AvailabilityZone": "us-west-2a",
        "DomainMemberships": [],
        "StorageType": "gp2",
        "DbiResourceId": "db-ABCDEFGHIJKLMNOPQRSTUVW",
        "ResumeFullAutomationModeTime": "2020-11-07T20:56:50.565Z",
        "KmsKeyId": "arn:aws:kms:us-west-2:012345678912:key/
aa111a11-111a-11a1-1a11-1111a11a1a1a",
        "StorageEncrypted": false,
        "AssociatedRoles": [],
        "DBInstanceClass": "db.m5.xlarge",
        "DbInstancePort": 0,
        "DBInstanceIdentifier": "my-custom-instance",
        "TagList": []
    }
```

Modifica di un'istanza database RDS Custom per SQL Server

La modifica di un'istanza database RDS Custom per SQL Server è simile a quella di Amazon RDS, ma le modifiche che è possibile apportare sono limitate ai seguenti casi:

- Modifica della classe di istanza database
- Modifica del periodo di conservazione del backup e della finestra di backup
- Modifica della finestra di manutenzione
- Aggiornamento della versione del motore database quando diventa disponibile una nuova versione
- Modifica dell'archiviazione allocata, della capacità di IOPS allocata e del tipo di archiviazione
- Modifica della porta del database
- Modifica dell'identificatore dell'istanza database
- Modifica delle credenziali master
- Consentire e rimuovere le implementazioni Multi-AZ
- Consentire l'accesso pubblico
- Modifica dei gruppi di sicurezza
- Modifica dei gruppi di sottorete

Le seguenti limitazioni si applicano alla modifica di un'istanza database RDS Custom per SQL Server:

- I gruppi di parametri e opzioni di DB personalizzato non sono supportati.

- Tutti i volumi di archiviazione collegati manualmente all'istanza database RDS Custom si trovano al di fuori del perimetro di supporto.

  Per ulteriori informazioni, consulta Perimetro di supporto RDS Custom.

Console

Modifica di un'istanza database RDS Custom per SQL Server

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, scegliere Databases (Database).
3. Scegliere l'istanza database che si vuole modificare.
4. Scegliere Modify (Modifica).
5. Eseguire le seguenti modifiche secondo necessità:
   a. Per DB engine version (Versione motore database) scegliere la nuova versione.
   b. Modificare il valore per DB instance class (Classe istanza database). Per le classi supportate, consulta Supporto delle classi di istanza database per RDS Custom for SQL Server.
   c. Modificare il valore per Backup retention period (Periodo di retention dei backup).
   d. Per Backup window (Finestra di backup), imposta i valori per Ora di inizio e Durata.
   e. Per Finestra di manutenzione istanza database, imposta i valori per Start day (Avvia giorno), Start time (Ora di inizio) e Duration (Durata).
6. Scegliere Continue (Continua).
7. Scegliere Apply immediately (Applica immediatamente) o Apply during the next scheduled maintenance window (Applica durante la prossima finestra di manutenzione pianificata).
8. Scegliere Modify DB Instance (Modifica istanza database).

AWS CLI

Per modificare un'istanza DB RDS Custom for SQL Server, usa il modify-db-instance AWS CLI comando. Impostazione dei parametri seguenti in base alle esigenze:

- `--db-instance-class` – Per le classi supportate, consulta Supporto delle classi di istanza database per RDS Custom for SQL Server.

- `--engine-version` – Numero di versione del motore del database a cui eseguire l'aggiornamento.
- `--backup-retention-period` – Quanto tempo mantenere i backup automatici, da 0 a 35 giorni.
- `--preferred-backup-window` – Intervallo di tempo giornaliero durante il quale vengono creati i backup automatici.
- `--preferred-maintenance-window` – L'intervallo temporale settimanale (in UTC) durante il quale può avvenire la manutenzione dei sistemi.
- `--apply-immediately`: utilizza `--apply-immediately` per applicare immediatamente le modifiche.

  Oppure utilizza `--no-apply-immediately` (impostazione di default) per applicare le modifiche durante la finestra di manutenzione successiva.

Modifica dell'archiviazione per un'istanza database RDS Custom per Oracle

La modifica dello spazio di archiviazione di un'istanza database RDS Custom per SQL Server è simile a quella di un'istanza database Amazon RDS, ma è possibile eseguire le seguenti operazioni:

- Aumento della dimensione dello spazio di archiviazione allocato.
- Modifica del tipo di archiviazione. Puoi utilizzare i tipi di archiviazione disponibili, ad esempio archiviazione per uso generico o capacità di IOPS allocata. Provisioned IOPS è supportato per i tipi di storage gp3, io1 e io2 Block Express.
- Modifica gli IOPS assegnati, se utilizzi i tipi di volume che supportano Provisioned IOPS.

Le seguenti limitazioni si applicano alla modifica dell'archiviazione di un'istanza database RDS Custom per SQL Server:

- L'archiviazione minima allocata per RDS Custom per SQL Server è 20 GiB e la dimensione di archiviazione massima supportata è 16 TiB.
- Come per Amazon RDS, non è possibile ridurre lo storage allocato. Si tratta di una limitazione dei volumi Amazon Elastic Block Store (Amazon EBS). Per ulteriori informazioni, consultare Uso dello storage per istanze database di Amazon RDS
- La scalabilità automatica dell'archiviazione non è supportata per le istanze database RDS Custom per SQL Server.

- Tutti i volumi di archiviazione che colleghi manualmente alla tua istanza database RDS Custom non vengono presi in considerazione per la scalabilità dell'archiviazione. Solo i volumi di dati predefiniti forniti da RDS, ovvero l'unità D, vengono presi in considerazione per la scalabilità dell'archiviazione.

  Per ulteriori informazioni, consulta Perimetro di supporto RDS Custom.
- Il dimensionamento dell'archiviazione di solito non causa alcuna interruzione o peggioramento delle prestazioni dell'istanza database. Dopo aver modificato le dimensioni di storage di un'istanza database, lo stato passa a storage-optimization (ottimizzazione-storage).
- L'ottimizzazione dello spazio di archiviazione può richiedere alcune ore. Non puoi apportare altre modifiche all'archiviazione prima di sei (6) ore dal completamento dell'ottimizzazione dello spazio di archiviazione nell'istanza. Per ulteriori informazioni, consultare Uso dello storage per istanze database di Amazon RDS

Per ulteriori informazioni sullo storage, consultare Storage delle istanze di database Amazon RDS.

Per informazioni generali sulla modifica dello storage, consulta Uso dello storage per istanze database di Amazon RDS.

Console

Per modificare l'archiviazione per un'istanza database RDS Custom per SQL Server

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, scegliere Databases (Database).
3. Scegliere l'istanza database che si vuole modificare.
4. Scegliere Modify (Modifica).
5. Eseguire le seguenti modifiche secondo necessità:

   a. Inserire un nuovo valore per Allocated Storage (Storage allocato). Questo valore deve essere maggiore di quello corrente e compreso tra 20 GiB e 16 TiB.

   b. Modificare il valore per Storage Type (Tipo di storage). Puoi scegliere tra i tipi di storage General Purpose o Provisioned IOPS disponibili. Provisioned IOPS è supportato per i tipi di storage gp3, io1 e io2 Block Express.

   c. Se si specifica un tipo di storage che supporta Provisioned IOPS, è possibile definire il valore Provisioned IOPS.

6. Scegliere Continue (Continua).
7. Scegliere Apply immediately (Applica immediatamente) o Apply during the next scheduled maintenance window (Applica durante la prossima finestra di manutenzione pianificata).
8. Scegliere Modify DB Instance (Modifica istanza database).

AWS CLI

Per modificare lo storage per un'istanza DB RDS Custom for SQL Server, usa il comando. modify-db-instance AWS CLI Impostazione dei parametri seguenti in base alle esigenze:

- `--allocated-storage`: la quantità di spazio di archiviazione, in gibibyte, da allocare per l'istanza database. Questo valore deve essere maggiore di quello corrente e compreso tra 20 e 16.384 GiB.
- `--storage-type`— Il tipo di archiviazione, ad esempio gp2, gp3, io1 o io2.
- `--iops` – La capacità di IOPS allocata per l'istanza database. È possibile specificarlo solo per i tipi di storage che supportano Provisioned IOPS (gp3, io1 e io2).
- `--apply-immediately`: utilizza `--apply-immediately` per applicare immediatamente le modifiche.

  Oppure utilizza `--no-apply-immediately` (impostazione di default) per applicare le modifiche durante la finestra di manutenzione successiva.

L'esempio seguente modifica la dimensione dello storage my-custom-instance a 200 GiB, il tipo di storage a io1 e Provisioned IOPS a 3000.

Example

PerLinux, o: macOS Unix

```
aws rds modify-db-instance \
    --db-instance-identifier my-custom-instance \
    --storage-type io1 \
    --iops 3000 \
    --allocated-storage 200 \
    --apply-immediately
```

Per Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier my-custom-instance ^
    --storage-type io1 ^
    --iops 3000 ^
    --allocated-storage 200 ^
    --apply-immediately
```

Assegnazione di tag alle risorse RDS Custom for SQL Server

Puoi taggare le risorse RDS Custom come con le risorse Amazon RDS, ma con alcune importanti differenze:

- Non creare o modificare la tag `AWSRDSCustom` richiesta per l'automazione RDS Custom. Se lo fai, potresti interrompere l'automazione.
- Il tag `Name` viene aggiunto alle risorse RDS Custom con il valore del prefisso `do-not-delete-rds-custom`. Qualsiasi valore passato dal cliente per la chiave viene sovrascritto.
- Le tag aggiunte alle istanze database RDS Custom durante la creazione vengono propagate a tutte le altre risorse RDS Custom correlate.
- Le tag non vengono propagate quando le aggiungi alle risorse RDS Custom dopo la creazione dell'istanza database.

Per informazioni sul tagging delle risorse, consulta Tagging delle risorse Amazon RDS.

Eliminazione di un'istanza database RDS Custom for SQL Server

Per eliminare un'istanza database RDS Custom per SQL Server, occorre eseguire quanto segue:

- Fornire il nome dell'istanza database.
- Seleziona o deseleziona l'opzione per acquisire uno snapshot di database finale dell'istanza database.
- Scegliere o deselezionare l'opzione per mantenere i backup automatici.

È possibile eliminare un'istanza database RDS Custom per SQL Server utilizzando la console o l'interfaccia della linea di comando. Il tempo necessario per eliminare l'istanza database può variare a seconda del periodo di conservazione del backup, ovvero del numero di backup da eliminare, della quantità di dati eliminati e dell'esecuzione di uno snapshot finale.

> **Warning**
>
> L'eliminazione di un'istanza database RDS Custom per SQL Server elimina definitivamente l'istanza EC2 e i volumi Amazon EBS associati. Non devi mai terminare o eliminare queste risorse. In caso contrario, l'eliminazione e la creazione dello snapshot finale potrebbero non riuscire.

> **Note**
>
> Non puoi creare una snapshot di database finale dell'istanza database se presenta uno di questi stati: `creating`, `failed`, `incompatible-create`, `incompatible-restore` o `incompatible-network`. Per ulteriori informazioni, consulta Visualizzazione dello stato dell'istanza database di Amazon RDS .

> **Important**
>
> Se scegli di acquisire uno snapshot finale, ti consigliamo di evitare di scrivere i dati nell'istanza database durante l'eliminazione dell'istanza database. Una volta avviata l'eliminazione dell'istanza database, non è garantito che le modifiche ai dati vengano acquisite dallo snapshot finale.

Console

Per eliminare un'istanza database RDS Custom

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, scegli Databases (Database) e seleziona l'istanza database RDS Custom per SQL Server da eliminare. Le istanze database RDS Custom per SQL Server mostrano il ruolo Instance (RDS Custom for SQL Server) (Istanza (RDS Custom per SQL Server)).
3. In Actions (Azioni), scegliere Delete (Elimina).
4. Per acquisire uno snapshot finale, scegli Create final snapshot (Crea snapshot finale) e fornisci un nome per Final snapshot name (Nome dello snapshot finale).

5. Per mantenere i backup automatici, scegliere Retain automated backups (Mantieni backup automatici).
6. Immettere **`delete me`** nella casella.
7. Scegliere Delete (Elimina).

AWS CLI

È possibile eliminare un'istanza DB RDS Custom for SQL Server utilizzando il delete-db-instance AWS CLI comando. Identificare l'istanza database utilizzando il parametro richiesto `--db-instance-identifier`. I parametri rimanenti sono gli stessi di un'istanza database Amazon RDS.

L'esempio seguente elimina l'istanza database RDS Custom per SQL Server denominata `my-custom-instance`, acquisisce uno snapshot finale e mantiene i backup automatici.

Example

Per LinuxmacOS, oUnix:

```
aws rds delete-db-instance \
    --db-instance-identifier my-custom-instance \
    --no-skip-final-snapshot \
    --final-db-snapshot-identifier my-custom-instance-final-snapshot \
    --no-delete-automated-backups
```

Per Windows:

```
aws rds delete-db-instance ^
    --db-instance-identifier my-custom-instance ^
    --no-skip-final-snapshot ^
    --final-db-snapshot-identifier my-custom-instance-final-snapshot ^
    --no-delete-automated-backups
```

Per acquisire uno snapshot finale, l'opzione `--final-db-snapshot-identifier` è obbligatoria e deve essere specificata.

Per ignorare lo snapshot finale, specifica nel comando l'opzione `--skip-final-snapshot` anziché le opzioni `--no-skip-final-snapshot` e `--final-db-snapshot-identifier`.

Per eliminare i backup automatici, specifica nel comando l'opzione `--delete-automated-backups` anziché `--no-delete-automated-backups`.

Avvio e arresto di un'istanza database RDS Custom per SQL Server

È possibile avviare e arrestare l'istanza database RDS Custom per SQL Server. Gli stessi requisiti e limitazioni per le istanze database RDS per SQL Server sono validi anche per l'arresto e l'avvio delle istanze database RDS Custom per SQL Server. Per ulteriori informazioni, consulta Arresto temporaneo di un'istanza database Amazon RDS.

Le seguenti considerazioni valgono per l'avvio e l'arresto dell'istanza database RDS Custom per SQL Server:

- La modifica di un attributo dell'istanza database RDS Custom per SQL Server mentre lo stato dell'istanza database è `STOPPED` non è supportata.
- È possibile arrestare e avviare un'istanza database RDS Custom per SQL Server solo se è configurata per una singola zona di disponibilità. Non puoi arrestare un'istanza database Amazon RDS per SQL Server in una configurazione multi-AZ.
- Verrà creato uno snapshot `SYSTEM` quando arresti un'istanza database RDS Custom per SQL Server. Lo snapshot verrà eliminato automaticamente al riavvio dell'istanza database RDS Custom per SQL Server.
- Se elimini l'istanza database EC2 quando l'istanza database RDS Custom per SQL Server è stata arrestata, l'unità `C:` verrà sostituita al riavvio dell'istanza database RDS Custom per SQL Server.
- L'unità `C:\`, il nome host e le configurazioni personalizzate vengono conservate in caso di arresto di un'istanza database RDS Custom per SQL Server, a condizione che il tipo di istanza non venga modificato.
- Le seguenti azioni faranno sì che RDS Custom posizioni l'istanza database al di fuori del perimetro di supporto. In questo caso, i costi delle ore di utilizzo dell'istanza database continueranno a essere addebitati:
  - Avvio dell'istanza EC2 sottostante mentre Amazon RDS è in stato di arresto. Per risolvere questo problema, puoi chiamare l'API Amazon RDS `start-db-instance` o arrestare l'istanza EC2 in modo che lo stato dell'istanza RDS Custom diventi `STOPPED`.
  - Arresto dell'istanza EC2 sottostante quando lo stato dell'istanza database RDS Custom per SQL Server è `ACTIVE`.

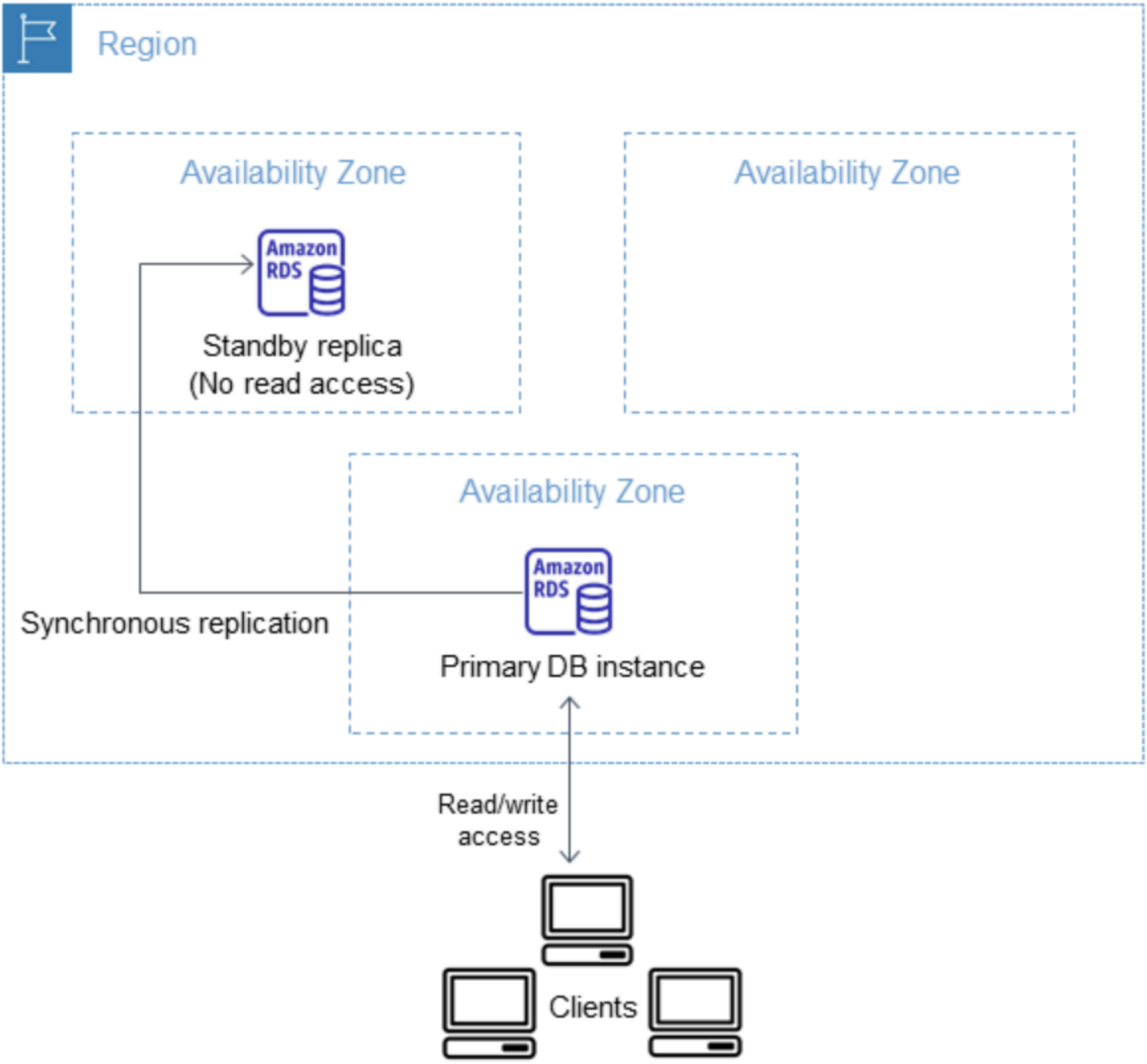
Per ulteriori informazioni sull'arresto e sull'avvio delle istanze database, consulta Arresto temporaneo di un'istanza database Amazon RDS e Avvio di un'istanza database Amazon RDS arrestata in precedenza.

Gestione di un'implementazione Multi-AZ per RDS Custom per SQL Server

In un'implementazione Multi-AZ di istanze DB per RDS Custom per SQL Server, Amazon RDS effettua automaticamente il provisioning e mantiene una replica in standby sincrona in un'altra zona di disponibilità (AZ). L'istanza database principale viene replicata in modo sincrono tra le zone di disponibilità in una replica in standby per garantire la ridondanza dei dati.

> **Important**
>
> Un'implementazione Multi-AZ per RDS Custom per SQL Server è diversa da un'implementazione Multi-AZ per RDS per SQL Server. A differenza di un'implementazione Multi-AZ per RDS per SQL Server, è necessario impostare i prerequisiti per RDS Custom per SQL Server prima di creare l'istanza DB Multi-AZ perché RDS Custom viene eseguito all'interno del tuo account e per questo scenario è necessario disporre di autorizzazioni. Se non vengono configurati i prerequisiti, l'istanza DB Multi-AZ potrebbe non funzionare o venire automaticamente convertita in un'istanza DB Single-AZ. Per ulteriori informazioni sui prerequisiti, consulta Prerequisiti per un'implementazione Multi-AZ con RDS Custom per SQL Server.

L’esecuzione di un'istanza database con disponibilità elevata può migliorare la disponibilità durante la manutenzione pianificata del sistema. In caso di manutenzione pianificata del database o interruzione non pianificata del servizio, Amazon RDS esegue automaticamente il failover sull'istanza DB up-to-date secondaria. Questa funzionalità consente alle operazioni del database di riprendere velocemente senza intervento manuale. Le istanze primarie e di standby usano lo stesso endpoint, il cui indirizzo di rete fisico passa alla replica secondaria come parte del processo di failover. Non è necessario riconfigurare l'applicazione quando si verifica un failover.



Puoi creare un'implementazione Multi-AZ per RDS Custom per SQL Server specificando Multi-AZ durante la creazione di un'istanza DB RDS Custom. Puoi utilizzare la console per convertire le istanze DB RDS Custom per SQL Server esistenti in implementazioni Multi-AZ modificando l'istanza DB e specificando l'opzione Multi-AZ. Inoltre, puoi specificare un'implementazione Multi-AZ per un'istanza DB mediante l'interfaccia della linea di comando AWS o l'API Amazon RDS.

La console RDS mostra la zona di disponibilità della replica in standby (zona di disponibilità secondaria). Puoi anche utilizzare il comando dell'interfaccia della riga di comando `describe-`

`db-instances` o l'operazione API `DescribeDBInstances` per trovare la zona di disponibilità secondaria.

Le istanze DB RDS Custom per SQL che utilizzano implementazioni Multi-AZ possono avere una latenza di scrittura e di commit maggiore rispetto a un'implementazione Single-AZ. Questo incremento può verificarsi a causa della replica sincrona dei dati tra istanze DB. È possibile che si verifichi una modifica nella latenza in caso di failover dell'implementazione nella replica di standby, sebbene AWS sia progettato con una connettività di rete a bassa latenza tra zone di disponibilità.

> **Note**
>
> Per carichi di lavoro di produzione, è consigliabile utilizzare una classe di istanze DB con capacità di IOPS (operazioni di input/output al secondo) allocata per ottenere prestazioni veloci e coerenti. Per altre informazioni sulle classi di istanza database, consulta Requisiti e limitazioni per Amazon RDS Custom for SQL Server.

Argomenti

Disponibilità di regioni e versioni

Le implementazioni Multi-AZ per RDS Custom per SQL Server sono supportate per le seguenti edizioni di SQL Server:

- SQL Server 2022 e 2019: Enterprise, Standard, Web e Developer Edition

> **Note**
>
> Le distribuzioni Multi-AZ per RDS Custom for SQL Server non sono supportate su SQL Server 2019 CU8 (15.00.4073.23) o versioni precedenti.

Le implementazioni Multi-AZ per RDS Custom per SQL Server sono disponibili in tutte le regioni in cui è disponibile RDS Custom per SQL Server. Per ulteriori informazioni sulla disponibilità delle implementazioni Multi-AZ per RDS Custom per SQL Server, consulta Regioni e motori DB supportati per RDS Custom per SQL Server.

Limitazioni per un'implementazione Multi-AZ con RDS Custom per SQL Server

Le implementazioni Multi-AZ con RDS Custom per SQL Server sono caratterizzate dalle seguenti limitazioni:

- Le implementazioni Multi-AZ tra regioni non sono supportate.
- Non è possibile configurare l'istanza DB secondaria per accettare attività di lettura del database.
- Quando si utilizza una versione del motore personalizzato (CEV) con un'implementazione Multi-AZ, anche l'istanza DB secondaria utilizzerà la stessa versione. L'istanza DB secondaria non può utilizzare una versione del motore personalizzato (CEV) diversa.

Prerequisiti per un'implementazione Multi-AZ con RDS Custom per SQL Server

Se si dispone di un'implementazione Single-AZ di RDS Custom per SQL Server, è necessario configurare i seguenti prerequisiti prima di convertirla in implementazione Multi-AZ. È possibile scegliere di completare i prerequisiti manualmente o con il modello fornito. CloudFormation Il CloudFormation modello più recente contiene i prerequisiti per le implementazioni Single-AZ e Multi-AZ.

> **Important**
>
> Per semplificare la configurazione, si consiglia di utilizzare il file modello AWS CloudFormation più recente fornito nelle istruzioni di configurazione della rete per creare i prerequisiti. Per ulteriori informazioni, consulta Configurazione con AWS CloudFormation.

> **Note**
>
> Quando per RDS Custom per SQL Server converti un'implementazione Single-AZ in implementazione Multi-AZ, è necessario configurare questi prerequisiti. Se non vengono impostati i prerequisiti, la configurazione Multi-AZ avrà esito negativo. Per configurare i prerequisiti, segui la procedura descritta in Modifica di un'implementazione RDS Custom per SQL Server Single-AZ in implementazione Multi-AZ.

- Aggiorna le regole in entrata e in uscita del gruppo di sicurezza RDS per consentire l'uso della porta 1120.
- Aggiungi una regola nella lista di controllo degli accessi (ACL) della tua rete privata che consenta l'uso delle porte TCP `0-65535` per il cloud privato virtuale (VPC) dell'istanza DB.
- Crea nuovi endpoint VPC per Amazon SQS VPC che consentano la comunicazione tra l'istanza DB RDS Custom per SQL Server e SQS.
- Aggiorna le autorizzazioni SQS nel ruolo del profilo dell'istanza.

Creazione di un'implementazione Multi-AZ per RDS Custom per SQL

Per creare un'implementazione Multi-AZ per RDS Custom per SQL Server, segui la procedura descritta in Creazione e connessione a un'istanza database per Amazon RDS Custom per SQL Server.

> **Important**
>
> Per semplificare la configurazione, si consiglia di utilizzare il file modello AWS CloudFormation più recente fornito nelle istruzioni di configurazione della rete. Per ulteriori informazioni, consulta Configurazione con AWS CloudFormation.

Per il completamento della creazione di un'implementazione Multi-AZ sono necessari alcuni minuti.

Modifica di un'implementazione RDS Custom per SQL Server Single-AZ in implementazione Multi-AZ

Puoi modificare un'istanza DB RDS Custom per SQL Server esistente da implementazione Single-AZ a implementazione Multi-AZ. Quando modifichi l'istanza DB, Amazon RDS esegue diverse operazioni:

- Crea uno snapshot dell'istanza DB primaria.
- Creazione di nuovi volumi per la replica in standby basati sullo snapshot. Questi volumi vengono inizializzati in background e le massime prestazioni del volume vengono raggiunte dopo la completa inizializzazione dei dati.
- Attiva la replica sincrona a livello di blocco tra le istanze DB primaria e secondaria.

> **Important**
>
> Si consiglia di evitare di modificare l'istanza DB RDS Custom per SQL Server da un'implementazione Single-AZ a implementazione Multi-AZ su un'istanza DB di produzione durante i periodi di picco dell'attività.

AWS utilizza uno snapshot per la creazione dell'istanza in standby per evitare tempi di inattività durante la conversione dell'implementazione da Single-AZ a Multi-AZ. Tuttavia si può verificare una riduzione delle prestazioni durante e dopo la conversione in implementazione Multi-AZ. Questo impatto può essere significativo per carichi di lavoro sensibili alla latenza di scrittura. Sebbene consenta di ripristinare grandi volumi di dati da snapshot, questa funzionalità può causare un aumento significativo della latenza delle operazioni I/O a causa della replica sincrona. Questa latenza può compromettere le prestazioni del database.

Argomenti

- Configurazione dei prerequisiti per modificare una distribuzione da Single-AZ a una Multi-AZ utilizzando CloudFormation
- Configurazione dei prerequisiti per modificare manualmente un'implementazione Single-AZ in implementazione Multi-AZ
- Modifica tramite la console RDS, l'interfaccia della linea di comando AWS o l'API RDS.

Configurazione dei prerequisiti per modificare una distribuzione da Single-AZ a una Multi-AZ utilizzando CloudFormation

Per utilizzare una distribuzione Multi-AZ, è necessario assicurarsi di aver applicato il CloudFormation modello più recente con i prerequisiti o configurare manualmente i prerequisiti più recenti. Se hai già applicato il modello di CloudFormation prerequisito più recente, puoi saltare questi passaggi.

Per configurare i prerequisiti di distribuzione RDS Custom for SQL Server Multi-AZ utilizzando CloudFormation

1. Aprire la CloudFormation console all'indirizzo https://console.aws.amazon.com/cloudformation.
2. Per avviare la procedura guidata Crea stack, seleziona lo stack esistente utilizzato per creare un'implementazione Single-AZ e scegli Aggiorna.

   Viene visualizzata la pagina Aggiorna stack.
3. In Prerequisito - Prepara modello, scegli Sostituisci il modello corrente.
4. Per Specify template (Specifica modello), procedi come segue:

   a. Scarica il file modello AWS CloudFormation più recente. Apri il menu contestuale (fai clic con il pulsante destro del mouse) per il link custom-sqlserver-onboard.zip e scegli Salva collegamento con nome.

   b. Salva ed estrai il file `custom-sqlserver-onboard.json` sul computer.

   c. Come Template source (Origine modello), scegliere Upload a template file (Carica un file di modello).

   d. Per Choose file (Scegli file), individua e quindi scegli `custom-sqlserver-onboard.json`.
5. Seleziona Avanti.

   Viene visualizzata la pagina Specify stack details (Specifica dettagli stack).
6. Per mantenere le opzioni predefinite, scegli Next (Avanti).

   Viene visualizzata la pagina Opzioni avanzate.
7. Per mantenere le opzioni predefinite, scegli Next (Avanti).
8. Per mantenere le opzioni predefinite, scegli Next (Avanti).
9. Nella pagina Rivedi modifiche, effettua le operazioni seguenti:

a. In Capabilities (Capacità), selezionare la casella di spunta I acknowledge that AWS CloudFormation might create IAM resources with custom names (Conferma che potrebbe creare risorse IAM con nomi personalizzati).

b. Seleziona Invia.

10. Verifica che l'aggiornamento abbia avuto esito positivo. Verifica che lo stato di operazione riuscita sia `UPDATE_COMPLETE`.

Se l'aggiornamento ha esito negativo, qualsiasi nuova configurazione specificata nel processo di aggiornamento verrà ripristinata. La risorsa esistente sarà ancora utilizzabile. Ad esempio, se si aggiungono regole ACL di rete numerate 18 e 19, ma esistono regole con gli stessi numeri, l'aggiornamento restituisce il seguente errore: `Resource handler returned message: "The network acl entry identified by 18 already exists.` In questo scenario è possibile modificare le regole ACL esistenti per utilizzare un numero inferiore a 18, quindi riprovare a eseguire l'aggiornamento.

Configurazione dei prerequisiti per modificare manualmente un'implementazione Single-AZ in implementazione Multi-AZ

> **Important**
>
> Per semplificare la configurazione, si consiglia di utilizzare il file modello AWS CloudFormation più recente fornito nelle istruzioni di configurazione della rete. Per ulteriori informazioni, consulta Configurazione dei prerequisiti per modificare una distribuzione da Single-AZ a una Multi-AZ utilizzando CloudFormation.

Se scegli di configurare i prerequisiti manualmente, esegui le seguenti operazioni.

1. Apri alla console Amazon VPC all'indirizzo https://console.aws.amazon.com/vpc/.
2. Scegli Endpoint. La pagina Creazione endpoint viene visualizzata.
3. In Categoria servizio, scegli Servizi AWS.
4. In Servizi, cerca *SQS*
5. In VPC, scegli il cloud privato virtuale (VPC) in cui viene implementata l'istanza DB di RDS Custom per SQL Server.

6. In Sottoreti, scegli le sottoreti in cui viene implementata l'istanza DB di RDS Custom per SQL Server.
7. In Gruppi di sicurezza, scegli il vpc-endpoint-sg gruppo `-`.
8. In Policy, scegli Personalizzato
9. Nella policy personalizzata, sostituisci *`AWS partition`*, *`Region`*, *`accountId`* e *`IAM-Instance-role`* con i tuoi valori.

```
                 {
        "Version": "2012-10-17",
        "Statement": [
            {
                "Condition": {
                    "StringLike": {
                        "aws:ResourceTag/AWSRDSCustom": "custom-sqlserver"
                    }
                },
                "Action": [
                    "SQS:SendMessage",
                    "SQS:ReceiveMessage",
                    "SQS:DeleteMessage",
                    "SQS:GetQueueUrl"
                ],
                "Resource": "arn:${AWS::Partition}:sqs:${AWS::Region}:
${AWS::AccountId}:do-not-delete-rds-custom-*",
                "Effect": "Allow",
                "Principal": {
                    "AWS": "arn:${AWS::Partition}:iam::${AWS::AccountId}:role/{IAM-
Instance-role}"
                }
            }
        ]
    }
```

10. Aggiorna il valore nel campo Profilo istanza in base all'autorizzazione ad accedere ad Amazon SQS. Sostituisci *`AWS partition`*, *`Region`* e *`accountId`* con i tuoi valori.

```
                    {
    "Sid": "SendMessageToSQSQueue",
```

```
    "Effect": "Allow",
    "Action": [
      "SQS:SendMessage",
      "SQS:ReceiveMessage",
      "SQS:DeleteMessage",
      "SQS:GetQueueUrl"

    ],
    "Resource": [
      {
        "Fn::Sub": "arn:${AWS::Partition}:sqs:${AWS::Region}:${AWS::AccountId}:do-
not-delete-rds-custom-*"
      }
    ],
    "Condition": {
      "StringLike": {
        "aws:ResourceTag/AWSRDSCustom": "custom-sqlserver"
      }
    }
  }
                        >
```

11. Aggiorna le regole in entrata e in uscita del gruppo di sicurezza Amazon RDS per consentire l'uso della porta 1120.

    a. In Gruppi di sicurezza, scegli il rds-custom-instance-sg gruppo -.

    b. *Per **le regole in entrata**, crea una regola **TCP personalizzata** per consentire la porta 1120 dal gruppo di origine. rds-custom-instance-sg*

    c. *Per **le regole in uscita**, crea una regola **TCP personalizzata** per consentire alla porta 1120 di accedere al gruppo di destinazione. rds-custom-instance-sg*

12. Aggiungi una regola nella lista di controllo degli accessi (ACL) della tua rete privata che consenta l'uso delle porte TCP `0-65535` per le sottoreti dell'istanza DB.

> **Note**
>
> Quando crei regole in Regola in entrata e Regola in uscita, annota il numero più alto esistente visualizzato nel campo Numero regola. Nel campo Numero regola le nuove

regole create devono avere un numero inferiore a 100 e non devono corrispondere a nessun altro valore esistente visualizzato nel campo Numero regola.

a. *In **Network ACL**, scegli il gruppo -. private-network-acl*

b. In Regole in entrata, crea una regola in Tutti TCP per consentire l'uso delle porte TCP `0-65535` con un'origine da *`privatesubnet1`* e *`privatesubnet2`*.

c. In Regole in uscita, crea una regola in Tutti TCP per consentire l'uso delle porte TCP `0-65535` alla destinazione *`privatesubnet1`* e *`privatesubnet2`*.

Modifica tramite la console RDS, l'interfaccia della linea di comando AWS o l'API RDS.

Dopo aver configurato i prerequisiti, è possibile modificare un'istanza DB RDS Custom per SQL Server da un'implementazione Single-AZ a implementazione Multi-AZ utilizzando la console RDS, l'Interfaccia della linea di comando AWS o l'API RDS.

Console

Per modificare un'implementazione RDS Custom per SQL Server esistente da Single-AZ a Multi-AZ

1. Accedi alla AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nella console Amazon RDS scegliere Databases (Database).

   Verrà visualizzato il riquadro Databases (Database).
3. Scegli l'istanza DB di RDS Custom per SQL Server da modificare.
4. In Operazioni, scegli Conversione in implementazione Multi-AZ.
5. Nella pagina Conferma, scegli Applica immediatamente per applicare le modifiche immediatamente. La scelta di questa opzione non causa tempi di inattività, ma è possibile riscontrare un impatto sulle prestazioni. In alternativa, puoi scegliere di applicare l'aggiornamento durante la successiva finestra di manutenzione. Per ulteriori informazioni, consulta Impostazione delle modifiche alla pianificazione.
6. Nella pagina Conferma, scegli Conversione in Multi-AZ.

AWS CLI

Per eseguire la conversione in una distribuzione di istanze DB Multi-AZ utilizzando ilAWS CLI, chiamate il modify-db-instancecomando e impostate l'`--multi-az`opzione. Specifica l'identificatore dell'istanza DB e i valori delle altre opzioni da modificare. Per ulteriori informazioni su ciascuna opzione, consulta Impostazioni per istanze database.

Example

Il codice seguente modifica `mycustomdbinstance` includendo l'opzione `--multi-az`. Le modifiche vengono applicate durante la prossima finestra di manutenzione utilizzando `--no-apply-immediately`. Utilizza `--apply-immediately` per applicare immediatamente le modifiche. Per ulteriori informazioni, consulta Impostazione delle modifiche alla pianificazione.

Per LinuxmacOS, oUnix:

```
aws rds modify-db-instance \
    --db-instance-identifier mycustomdbinstance \
    --multi-az \
    --no-apply-immediately
```

Per Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mycustomdbinstance ^
    --multi-az  \ ^
    --no-apply-immediately
```

API RDS

Per effettuare la conversione delle istanze DB in implementazione Multi-AZ tramite l'API RDS, chiama l'operazione ModifyDBInstance impostando il parametro `MultiAZ` su true.

Modifica di un'implementazione RDS Custom per SQL Server Multi-AZ in implementazione Single-AZ

Puoi modificare un'istanza DB RDS Custom per SQL Server esistente da implementazione Multi-AZ a implementazione Single-AZ.

Console

Per modificare un'istanza DB RDS Custom per SQL Server da implementazione Multi-AZ a implementazione Single-AZ.

1. Accedi alla AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nella console Amazon RDS scegliere Databases (Database).

   Verrà visualizzato il riquadro Databases (Database).
3. Scegli l'istanza DB di RDS Custom per SQL Server da modificare.
4. In Implementazione Multi-AZ, scegli No.
5. Nella pagina Conferma, scegli Applica immediatamente per applicare le modifiche immediatamente. La scelta di questa opzione non causa tempi di inattività, ma è possibile riscontrare un impatto sulle prestazioni. In alternativa, puoi scegliere di applicare l'aggiornamento durante la successiva finestra di manutenzione. Per ulteriori informazioni, consulta Impostazione delle modifiche alla pianificazione.
6. Nella pagina Conferma, scegli Modifica istanza database.

AWS CLI

Per modificare una distribuzione Multi-AZ in una distribuzione Single-AZ utilizzandoAWS CLI, chiama il modify-db-instancecomando e includi l'`--no-multi-az`opzione. Specifica l'identificatore dell'istanza DB e i valori delle altre opzioni da modificare. Per ulteriori informazioni su ciascuna opzione, consulta Impostazioni per istanze database.

Example

Il codice seguente modifica `mycustomdbinstance` includendo l'opzione `--no-multi-az`. Le modifiche vengono applicate durante la prossima finestra di manutenzione utilizzando `--no-apply-immediately`. Utilizza `--apply-immediately` per applicare immediatamente le modifiche. Per ulteriori informazioni, consulta Impostazione delle modifiche alla pianificazione.

PerLinux, omacOS: Unix

```
aws rds modify-db-instance \
    --db-instance-identifier mycustomdbinstance \
    --no-multi-az  \
```

```
    --no-apply-immediately
```

Per Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mycustomdbinstance ^
    --no-multi-az \ ^
    --no-apply-immediately
```

API RDS

Per modificare un'implementazione Multi-AZ in implementazione Single-AZ tramite l'API RDS, chiama l'operazione ModifyDBInstance impostando il parametro `MultiAZ` su `false`.

Processo di failover per un'implementazione Multi-AZ di RDS Custom per SQL Server

Se un'interruzione pianificata o non pianificata dell'istanza database comporta un defect dell'infrastruttura, Amazon RDS passa automaticamente a una replica in standby in un'altra zona di disponibilità, se hai abilitato l'implementazione Multi-AZ. Il tempo necessario per il completamento del failover varia in base all'attività del database e ad altre condizioni presenti quando l'istanza database primaria diventa non disponibile. Il failover richiede in genere da 60 a 120 secondi. tempo che può tuttavia aumentare in caso di transazioni di grandi dimensioni o di un processo di ripristino di lunga durata. Al termine del failover, la modifica della console RDS in base alla nuova zona di disponibilità può richiedere ulteriore tempo.

> **Note**
>
> Puoi forzare un failover manualmente quando riavvii un'istanza database con failover. Per ulteriori informazioni sul riavvio di un'istanza database, consulta Riavvio di un'istanza database.

Amazon RDS gestisce i failover automaticamente, in modo da consentirti di riprendere le operazioni database il più rapidamente possibile, senza alcun intervento amministrativo. L'istanza database principale passa automaticamente alla replica di standby qualora si verifichi una delle condizioni riportate nella seguente tabella. Puoi visualizzare questi motivi di failover nel log eventi RDS.

Motivo del failover	Descrizione
`The operating system for the RDS Custom for SQL Server Multi-AZ DB instance is being patched in an offline operation`	È stato attivato un failover durante la finestra di manutenzione per una patch del sistema operativo o un aggiornamento di sicurezza . Per ulteriori informazioni, consulta Manutenzione di un'istanza database.
`The primary host of the RDS Custom for SQL Server Multi-AZ DB instance is unhealthy.`	L'implementazione istanza database Multi-AZ ha rilevato un'istanza database primaria compromessa e ha attivato il failover.
`The primary host of the RDS Custom for SQL Server Multi-AZ DB instance is unreachable due to loss of network connectivity.`	Il monitoraggio RDS ha rilevato un errore di raggiungibilità della rete per l'istanza database primaria e ha attivato un failover.
`The RDS Custom for SQL Server Multi-AZ DB instance was modified by the customer.`	Una modifica dell'istanza database RDS ha attivato un failover. Per ulteriori informazioni, consulta Modifica di un'istanza database RDS Custom per SQL Server.
`The storage volume of the primary host of the RDS Custom for SQL Server Multi-AZ DB instance experienced a failure.`	L'implementazione dell'istanza database Multi-AZ ha rilevato un problema di archiviazioine nell'istanza DB principale e ha avviato il failover.

Motivo del failover	Descrizione
`The user requested a failover of the RDS Custom for SQL Server Multi-AZ DB instance.`	L'istanza DB Multi-AZ di RDS Custom per SQL Server è stata riavviata con un failover. Per ulteriori informazioni, consulta Riavvio di un'istanza database.
`The RDS Custom for SQL Server Multi-AZ primary DB instance is busy or unresponsive.`	L'istanza database primaria non risponde. Si consiglia di effettuare la procedura seguente: • Esamina i registri degli eventi e i CloudWatch registri per verificar e l'utilizzo eccessivo di CPU, memoria o spazio di swap. Per ulteriori informazioni, consulta Utilizzo della notifica degli eventi di Amazon RDS. • Crea una regola che si attiva se si verifica un evento RDS: Per ulteriori informazioni, consulta Creazione di una regola che si attiva su un evento Amazon RDS. • Valuta il carico di lavoro per determinare se si sta utilizzando la classe di istanza database appropriata. Per ulteriori informazioni, consulta Classi di istanze database.

Per determinare se l'istanza database Multi-AZ è soggetta a failover, è possibile eseguire le seguenti operazioni:

- Configura gli abbonamenti a eventi database per inviare una notifica tramite e-mail o SMS in caso di failover. Per ulteriori informazioni sugli eventi di , consulta Utilizzo della notifica degli eventi di Amazon RDS.
- Visualizza gli eventi database utilizzando la console RDS o le operazioni dell'API.
- Visualizza lo stato corrente dell'implementazione Multi-AZ dell'istanza DB RDS Custom per SQL Server utilizzando la console RDS, la CLI o le operazioni API.

Impostazioni Time to live (TTL) con applicazioni che utilizzano un'implementazione Multi-AZ di RDS Custom per SQL Server

Il meccanismo di failover modifica automaticamente il record Domain Name System (DNS) dell'istanza database in modo da fare riferimento all'istanza database standby. Di conseguenza, sarà necessario ristabilire le connessioni esistenti alla propria istanza database. Assicurati che qualsiasi valore di configurazione della cache DNS time-to-live (TTL) sia basso e verifica che l'applicazione non memorizzi nella cache DNS per un periodo di tempo prolungato. Un valore TTL elevato potrebbe impedire all'applicazione di riconnettersi rapidamente all'istanza DB dopo il failover.

Backup e ripristino di un'istanza database di Amazon RDS Custom per SQL Server

Come Amazon RDS, RDS Custom crea e salva backup automatici dell'istanza DB di RDS Custom for SQL Server quando è abilitata la conservazione dei backup. Puoi inoltre eseguire il backup dell'istanza database manualmente. I backup automatici comprendono backup di istantanee e backup dei log delle transazioni. I backup istantanei vengono eseguiti per l'intero volume di archiviazione dell'istanza DB durante la finestra di backup specificata. I backup dei log delle transazioni vengono eseguiti per i database idonei al PITR a intervalli regolari. RDS Custom salva i backup automatici dell'istanza DB in base al periodo di conservazione dei backup specificato. È possibile utilizzare i backup automatici per ripristinare l'istanza DB in un determinato momento entro il periodo di conservazione dei backup.

È inoltre possibile eseguire backup di istantanee manualmente. È possibile creare una nuova istanza DB da questi backup istantanei in qualsiasi momento. Per ulteriori informazioni sulla creazione di uno snapshot di database, consulta Creazione di una snapshot RDS Custom per SQL Server.

Sebbene i backup istantanei funzionino operativamente come backup completi, ti viene fatturato solo l'utilizzo incrementale dello storage. La prima snapshot di un'istanza database RDS Custom contiene i dati dell'intera istanza database. Gli snapshot successivi dello stesso database sono incrementali, ovvero vengono salvati solo i dati che sono cambiati dal salvataggio dell'ultimo snapshot.

Argomenti

Creazione di una snapshot RDS Custom per SQL Server

RDS Custom per SQL Server crea una snapshot dei volumi di storage dell'istanza database, eseguendo il backup dell'intera istanza database anziché dei singoli database. Quando crei una snapshot, specifica di quale istanza database RDS Custom per SQL Server eseguire il backup. Dai un nome alla snapshot database in modo che tu possa ripristinarla in un secondo momento.

Quando crei uno snapshot, RDS Custom for SQL Server crea uno snapshot Amazon EBS per volume(`D:`), che è il volume del database collegato all'istanza DB. Per semplificare l'associazione delle snapshot a un'istanza database specifica, sono contrassegnate con `DBSnapshotIdentifier`, `DbiResourceId` e `VolumeType`.

La creazione di una snapshot DB si traduce in una breve interruzione delle operazioni di I/O. Questa sospensione può durare da pochi secondi a pochi minuti, a seconda delle dimensioni e della classe dell'istanza database. Il tempo di creazione dello snapshot varia in base al numero e alla dimensione totali dei database. Per ulteriori informazioni sul numero di database idonei per un'operazione di ripristino in tempo reale (PITR), vedere. Numero di database idonei per il PITR per tipo di classe di istanza

Poiché lo snapshot include l'intero volume d'archiviazione, la dimensione dei file, come i file temporanei, influisce sul tempo di creazione dello snapshot. Per ulteriori informazioni sulla creazione di snapshot, consulta Creazione di uno snapshot DB per un'istanza DB Single-AZ.

Creazione di una snapshot RDS Custom per SQL Server utilizzando la console o la AWS CLI.

Console

Per creare una snapshot RDS Custom

1. Accedi alla AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel pannello di navigazione, scegliere Databases (Database).
3. Nell'elenco di istanze database RDS Custom scegliere l'istanza database per cui si desidera acquisire uno snapshot.
4. Per Actions (Operazioni), selezionare Take snapshot (Acquisisci snapshot).

   Viene visualizzata la finestra Acquisizione di snapshot DB.
5. Per Nome snapshot, inserisci il nome dello snapshot.
6. Seleziona Acquisisci snapshot.

AWS CLI

È possibile creare un'istantanea di un'istanza DB personalizzata RDS utilizzando il comando. create-db-snapshotAWS CLI

Puoi specificare le seguenti opzioni:

- `--db-instance-identifier` – Identificare l'istanza database RDS Custom di cui effettuare il backup
- `--db-snapshot-identifier` – Assegna i nomi alla snapshot RDS Custom in modo che tu possa ripristinarla in un secondo momento

In questo esempio crei uno snapshot database denominata *`my-custom-snapshot`* per un'istanza database RDS Custom denominata *`my-custom-instance`*.

Example

PerLinux, omacOS: Unix

```
aws rds create-db-snapshot \
    --db-instance-identifier my-custom-instance \
    --db-snapshot-identifier my-custom-snapshot
```

Per Windows:

```
aws rds create-db-snapshot ^
    --db-instance-identifier my-custom-instance ^
    --db-snapshot-identifier my-custom-snapshot
```

Ripristino da una snapshot database RDS Custom per SQL Server

Quando ripristini un'istanza database RDS Custom per SQL Server, devi fornire il nome della snapshot database e il nome della nuova istanza. Non puoi eseguire il ripristino da una snapshot a un'istanza database RDS Custom esistente. Quando esegui il ripristino, viene creata una nuova istanza database RDS Custom per SQL Server.

Il ripristino da un'istantanea ripristinerà il volume di archiviazione al momento in cui è stata scattata l'istantanea. Ciò includerà tutti i database e tutti gli altri file presenti nel volume. (`D:`)

Console

Per ripristinare un'istanza database RDS Custom da uno snapshot database

1. Accedi alla AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, selezionare Snapshots (Snapshot).

3. Scegliere la snapshot DB dalla quale effettuare il ripristino.
4. Per Actions (Operazioni), selezionare Restore Snapshot (Ripristina snapshot).
5. Nella pagina Restore DB Instance (Ripristina istanza database), per DB Instance Identifier (Identificatore istanze DB), immettere il nome dell'istanza database RDS Custom ripristinata.
6. Selezionare Ripristina istanza database.

AWS CLI

È possibile ripristinare un'istantanea RDS Custom DB utilizzando il comando restore-db-instance-fromAWS CLI-db-snapshot.

Se la snapshot da cui si sta ripristinando è per un'istanza database privata, assicurarsi di specificare entrambi i valori corretti `db-subnet-group-name` e `no-publicly-accessible`. In caso contrario, l'istanza database è accessibile pubblicamente per impostazione predefinita. Sono richieste le seguenti opzioni:

- `db-snapshot-identifier` – Identifica la snapshot da cui eseguire il ripristino
- `db-instance-identifier` – Specifica il nome dell'istanza database RDS Custom da creare dalla snapshot database
- `custom-iam-instance-profile`: specifica il profilo di istanza associato all'istanza Amazon EC2 sottostante di un'istanza database RDS Custom.

Il codice seguente ripristina la snapshot denominata `my-custom-snapshot` per `my-custom-instance`.

Example

Per, o: Linux macOS Unix

```
aws rds restore-db-instance-from-db-snapshot \
  --db-snapshot-identifier my-custom-snapshot \
  --db-instance-identifier my-custom-instance \
  --custom-iam-instance-profile AWSRDSCustomInstanceProfileForRdsCustomInstance \
  --no-publicly-accessible
```

Per Windows:

```
aws rds restore-db-instance-from-db-snapshot ^
```

```
  --db-snapshot-identifier my-custom-snapshot ^
  --db-instance-identifier my-custom-instance ^
  --custom-iam-instance-profile AWSRDSCustomInstanceProfileForRdsCustomInstance ^
  --no-publicly-accessible
```

Ripristino di un'istanza RDS Custom per SQL Server in un determinato momento

Puoi ripristinare un'istanza database in un punto temporale specifico (PITR), creando una nuova istanza database. Per supportare PITR, le istanze DB devono avere la conservazione dei backup abilitata.

L'ultimo orario ripristinabile di un'istanza database RDS Custom per SQL Server dipende da diversi fattori, ma generalmente è entro 5 minuti dall'orario attuale. Per visualizzare l'ora di ripristino più recente per un'istanza DB, usa il AWS CLI describe-db-instancescomando e guarda il valore restituito nel `LatestRestorableTime` campo per l'istanza DB. Per visualizzare l'ora di ripristino più recente per ogni istanza del DB nella console Amazon RDS, scegliere Backup automatici.

Puoi eseguire il ripristino point-in-time durante il periodo di retention dei backup. Per visualizzare il tempo di ripristino più breve per ogni istanza del DB, scegliere Backup automatici nella console Amazon RDS.

Per informazioni generali su PITR, consulta Ripristino a un'ora specifica per un'istanza database.

Argomenti

- Considerazioni PITR per RDS Custom per SQL Server
- Numero di database idonei per il PITR per tipo di classe di istanza
- Rendere i database non idonei per PITR
- Log sulle transazioni in Amazon S3
- PITR Restore utilizzando l'API AWS Management ConsoleAWS CLI, the o RDS.

Considerazioni PITR per RDS Custom per SQL Server

In RDS Custom per SQL Server, PITR differisce secondo le seguenti importanti modalità da PITR in Amazon RDS:

- PITR ripristina solo i database nell'istanza database. Non ripristina il sistema operativo o i file sull'unità C:.

- Per un'istanza DB RDS Custom per SQL Server, viene eseguito automaticamente il backup di un database ed è idoneo per PITR solo alle seguenti condizioni:
  - Il database è online.
  - Il suo modello di ripristino è impostato su `FULL`.
  - È scrivibile.
  - Ha i suoi file fisici sull'unità D:.
  - Non è elencato nella tabella `rds_pitr_blocked_databases`. Per ulteriori informazioni, consulta Rendere i database non idonei per PITR.
- I database idonei per PITR sono determinati dall'ordine del relativo ID di database. RDS Custom per SQL Server consente fino a 5.000 database per istanza database. Tuttavia, il numero massimo di database ripristinati da un'operazione PITR per un'istanza DB RDS Custom for SQL Server dipende dal tipo di classe di istanza. Per ulteriori informazioni, consulta Numero di database idonei per il PITR per tipo di classe di istanza.

  Altri database che non fanno parte di PITR possono essere ripristinati dalle istantanee del DB, inclusi i backup automatici delle istantanee utilizzati per PITR.
- L'aggiunta di un nuovo database, la rinominazione di un database o il ripristino di un database idoneo per PITR avvia uno snapshot dell'istanza database.
- Il numero massimo di database idonei per PITR cambia quando l'istanza del database viene sottoposta a un'operazione di calcolo su scala, a seconda del tipo di classe di istanza di destinazione. Se l'istanza viene scalata verso l'alto, in modo da consentire a più database sull'istanza di essere idonei per il PITR, viene eseguita una nuova istantanea.
- I database ripristinati hanno lo stesso nome dell'istanza database di origine. Non puoi specificare un nome diverso.
- `AWSRDSCustomSQLServerIamRolePolicy`richiede l'accesso ad altri servizi. AWS Per ulteriori informazioni, consulta Aggiungi una politica di accesso a AWSRDSCustomSQLServerInstanceRole.
- Le modifiche al fuso orario non sono supportate per RDS Custom per SQL Server. Se si modifica il fuso orario dell'istanza database o del sistema operativo, PITR (o un'altra automazione) non funziona.

Numero di database idonei per il PITR per tipo di classe di istanza

La tabella seguente mostra il numero massimo di database idonei per PITR in base al tipo di classe di istanza.

Tipo di classe di istanza	Numero massimo di database idonei al PITR
db.*.large	100
da db.*.xlarge a db.*.2xlarge	150
db.*.4xlarge a db.*.8xlarge	300
db.*.12xlarge a db.*.16xl arge	600
db.*.24xlarge, db.*32xlarge	1000

*Rappresenta diversi tipi di classi di istanze.

Il numero massimo di database idonei per PITR su un'istanza DB dipende dal tipo di classe di istanza. Il numero varia da 100 per i tipi di classe di istanza più piccoli a 1000 per i tipi di classi di istanze più grandi supportati da RDS Custom for SQL Server. I database (`master, model, msdb, tempdb`) di sistema SQL Server non sono inclusi in questo limite. Quando un'istanza DB viene ridimensionata verso l'alto o verso il basso, a seconda del tipo di classe dell'istanza di destinazione, RDS Custom aggiornerà automaticamente il numero di database idonei per PITR. RDS Custom for SQL Server verrà inviato `RDS-EVENT-0352` quando il numero massimo di database idonei per PITR cambia su un'istanza DB. Per ulteriori informazioni, consulta Eventi di versioni personalizzate del motore.

> **Note**
>
> Il supporto PITR per più di 100 database è disponibile solo sulle istanze DB create dopo il 26 agosto 2023. Per le istanze create prima del 26 agosto 2023, il numero massimo di database idonei per PITR è 100, indipendentemente dalla classe di istanza. Per abilitare il supporto

PITR per più di 100 database su istanze DB create prima del 26 agosto 2023, puoi eseguire la seguente azione:

- Aggiorna la versione del motore DB alla versione 15.00.4322.2.v1 o successiva

Durante un'operazione PITR, RDS Custom ripristinerà tutti i database che facevano parte di PITR sull'istanza DB di origine al momento del ripristino. Una volta che l'istanza DB di destinazione ha completato le operazioni di ripristino, se la conservazione dei backup è abilitata, l'istanza DB inizierà il backup in base al numero massimo di database idonei per il PITR sull'istanza DB di destinazione.

Ad esempio, se l'istanza DB viene eseguita su un'`db.*.xlarge`istanza con 200 database:

1. RDS Custom for SQL Server sceglierà i primi 150 database, ordinati in base all'ID del database, per il backup PITR.
2. L'istanza viene modificata per scalarla fino a db.*.4xlarge.
3. Una volta completata l'operazione di calcolo della scalabilità, RDS Custom for SQL Server sceglierà i primi 300 database, ordinati in base all'ID del database, per il backup PITR. Ciascuno dei 200 database che soddisfano i requisiti PITR sarà ora idoneo per il PITR.
4. Ora modifichi l'istanza per ridurla a db.*.xlarge.
5. Una volta completata l'operazione di calcolo della scalabilità, RDS Custom for SQL Server selezionerà nuovamente i primi 150 database, ordinati in base all'ID del database, per il backup PITR.

Rendere i database non idonei per PITR

È possibile scegliere di escludere singoli database da PITR. Per fare questo, metti i loro valori `database_id` in una tabella `rds_pitr_blocked_databases`. Utilizza il seguente script SQL per creare la tabella.

Per creare la tabella rds_pitr_blocked_databases

- Esegui il seguente script SQL.

```
create table msdb..rds_pitr_blocked_databases
(
database_id INT NOT NULL,
database_name SYSNAME NOT NULL,
```

```
db_entry_updated_date datetime NOT NULL DEFAULT GETDATE(),
db_entry_updated_by SYSNAME NOT NULL DEFAULT CURRENT_USER,
PRIMARY KEY (database_id)
);
```

Per l'elenco dei database idonei e non idonei, consulta il file `RI.End` nella directory `RDSCustomForSQLServer/Instances/`*`DB_instance_resource_ID`*`/TransactionLogMetadata` nel bucket Amazon S3 `do-not-delete-rds-custom-`*`$ACCOUNT_ID`*`-`*`$REGION`*`-`*`unique_identifier`*. Per ulteriori informazioni sul file `RI.End`, consulta Log sulle transazioni in Amazon S3.

È inoltre possibile determinare l'elenco dei database idonei per PITR utilizzando il seguente script SQL. Imposta la `@limit` variabile sul numero massimo di database idonei per il PITR per la classe di istanze. Per ulteriori informazioni, consulta Numero di database idonei per il PITR per tipo di classe di istanza.

Per determinare l'elenco dei database idonei per PITR su una classe di istanze DB

- Esegui il seguente script SQL.

```
DECLARE @Limit INT;
SET @Limit = (insert-database-instance-limit-here);

USE msdb;
IF (EXISTS (SELECT * FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_SCHEMA = 'dbo' AND
 TABLE_NAME = 'rds_pitr_blocked_databases'))
    WITH TABLE0 AS (
        SELECT hdrs.database_id as DatabaseId, sdb.name as DatabaseName,
 'ALWAYS_ON_NOT_WRITABLE_REPLICA' as Reason, NULL as DatabaseNameOnPitrTable
        FROM sys.dm_hadr_database_replica_states hdrs
        INNER JOIN sys.databases sdb ON sdb.database_id = hdrs.database_id
        WHERE (hdrs.is_local = 1 AND hdrs.is_primary_replica = 0)
        OR (sys.fn_hadr_is_primary_replica (sdb.name) = 1 AND DATABASEPROPERTYEX
 (sdb.name, 'Updateability') = 'READ_ONLY')
    ),
    TABLE1 as (
            SELECT dbs.database_id as DatabaseId, sysdbs.name as DatabaseName,
 'OPTOUT' as Reason,
            CASE WHEN dbs.database_name = sysdbs.name THEN NULL ELSE
 dbs.database_name END AS DatabaseNameOnPitrTable
            FROM msdb.dbo.rds_pitr_blocked_databases dbs
```

```
            INNER JOIN sys.databases sysdbs ON dbs.database_id = sysdbs.database_id
            WHERE sysdbs.database_id > 4
            ),
    TABLE2 as (
            SELECT
            db.name AS DatabaseName,
            db.create_date AS CreateDate,
            db.state_desc AS DatabaseState,
            db.database_id AS DatabaseId,
            rs.database_guid AS DatabaseGuid,
            rs.last_log_backup_lsn AS LastLogBackupLSN,
            rs.recovery_fork_guid RecoveryForkGuid,
            rs.first_recovery_fork_guid AS FirstRecoveryForkGuid,
            db.recovery_model_desc AS RecoveryModel,
            db.is_auto_close_on AS IsAutoClose,
            db.is_read_only as IsReadOnly,
            NEWID() as FileName,
            CASE WHEN(db.state_desc = 'ONLINE'
                    AND db.recovery_model_desc != 'SIMPLE'
                    AND((db.is_auto_close_on = 0 and db.collation_name IS NOT NULL)
 OR db.is_auto_close_on = 1))
                    AND db.is_read_only != 1
                    AND db.user_access = 0
                    AND db.source_database_id IS NULL
                    AND db.is_in_standby != 1
                    THEN 1 ELSE 0 END AS IsPartOfSnapshot,
            CASE WHEN db.source_database_id IS NULL THEN 0 ELSE 1 END AS
 IsDatabaseSnapshot
            FROM sys.databases db
            INNER JOIN sys.database_recovery_status rs
            ON db.database_id = rs.database_id
            WHERE DB_NAME(db.database_id) NOT IN('tempdb') AND
            db.database_id NOT IN (SELECT DISTINCT DatabaseId FROM TABLE1) AND
            db.database_id NOT IN (SELECT DISTINCT DatabaseId FROM TABLE0)
        ),
        TABLE3 as(
            Select @Limit+count(DatabaseName) as TotalNumberOfDatabases from TABLE2
 where TABLE2.IsPartOfSnapshot=1 and DatabaseName in ('master','model','msdb')
        )
        SELECT TOP(SELECT TotalNumberOfDatabases from TABLE3)
 DatabaseName,CreateDate,DatabaseState,DatabaseId from TABLE2 where
 TABLE2.IsPartOfSnapshot=1
        ORDER BY TABLE2.DatabaseID ASC
ELSE
```

```
   WITH TABLE0 AS (
       SELECT hdrs.database_id as DatabaseId, sdb.name as DatabaseName,
'ALWAYS_ON_NOT_WRITABLE_REPLICA' as Reason, NULL as DatabaseNameOnPitrTable
       FROM sys.dm_hadr_database_replica_states hdrs
       INNER JOIN sys.databases sdb ON sdb.database_id = hdrs.database_id
       WHERE (hdrs.is_local = 1 AND hdrs.is_primary_replica = 0)
       OR (sys.fn_hadr_is_primary_replica (sdb.name) = 1 AND DATABASEPROPERTYEX
(sdb.name, 'Updateability') = 'READ_ONLY')
   ),
   TABLE1 as (
           SELECT
           db.name AS DatabaseName,
           db.create_date AS CreateDate,
           db.state_desc AS DatabaseState,
           db.database_id AS DatabaseId,
           rs.database_guid AS DatabaseGuid,
           rs.last_log_backup_lsn AS LastLogBackupLSN,
           rs.recovery_fork_guid RecoveryForkGuid,
           rs.first_recovery_fork_guid AS FirstRecoveryForkGuid,
           db.recovery_model_desc AS RecoveryModel,
           db.is_auto_close_on AS IsAutoClose,
           db.is_read_only as IsReadOnly,
           NEWID() as FileName,
           CASE WHEN(db.state_desc = 'ONLINE'
                   AND db.recovery_model_desc != 'SIMPLE'
                   AND((db.is_auto_close_on = 0 and db.collation_name IS NOT NULL)
OR db.is_auto_close_on = 1))
                   AND db.is_read_only != 1
                   AND db.user_access = 0
                   AND db.source_database_id IS NULL
                   AND db.is_in_standby != 1
                   THEN 1 ELSE 0 END AS IsPartOfSnapshot,
           CASE WHEN db.source_database_id IS NULL THEN 0 ELSE 1 END AS
IsDatabaseSnapshot
           FROM sys.databases db
           INNER JOIN sys.database_recovery_status rs
           ON db.database_id = rs.database_id
           WHERE DB_NAME(db.database_id) NOT IN('tempdb') AND
           db.database_id NOT IN (SELECT DISTINCT DatabaseId FROM TABLE0)
       ),
       TABLE2 as(
           SELECT @Limit+count(DatabaseName) as TotalNumberOfDatabases from TABLE1
where TABLE1.IsPartOfSnapshot=1 and DatabaseName in ('master','model','msdb')
       )
```

```
        select top(select TotalNumberOfDatabases from TABLE2)
DatabaseName,CreateDate,DatabaseState,DatabaseId from TABLE1 where
TABLE1.IsPartOfSnapshot=1
        ORDER BY TABLE1.DatabaseID ASC
```

> **Note**
>
> I database che sono solo collegamenti simbolici sono inoltre esclusi dai database idonei per le operazioni PITR. La query precedente non filtra in base a questi criteri.

Log sulle transazioni in Amazon S3

Il periodo di retention dei backup determina se i log sulle transazioni per le istanze database RDS Custom per SQL Server vengono automaticamente estratti e caricati su Amazon S3. Un valore diverso da zero significa che vengono creati backup automatici e che l'agente RDS Custom carica i log sulle transazioni su S3 ogni 5 minuti.

I file di log delle transazioni su S3 sono crittografati mentre sono inattivi tramite AWS KMS key che hai fornito quando hai creato l'istanza database. Per ulteriori informazioni, consulta Protezione dei dati con la crittografia lato server nella Guida per l'utente di Amazon Simple Storage Service.

I log delle transazioni per ciascun database vengono caricati in un bucket S3 denominato `do-not-delete-rds-custom-`*`$ACCOUNT_ID`*`-`*`$REGION`*`-`*`unique_identifier`*. La directory `RDSCustomForSQLServer/Instances/`*`DB_instance_resource_ID`* nel bucket S3 contiene due sottodirectory:

- `TransactionLogs` – Contiene i log delle transazioni per ciascun database e i rispettivi metadati.

  Il nome del file di log delle transazioni segue il pattern *`yyyyMMddHHmm`*`.`*`database_id`*`.`*`timestamp`*, ad esempio:

  ```
  202110202230.11.1634769287
  ```

  Lo stesso nome del file con il suffisso `_metadata` contiene informazioni sul log delle transazioni come numeri di sequenza di log, nome del database e `RdsChunkCount`. `RdsChunkCount` determina quanti file fisici rappresentano un singolo file di log delle transazioni. Potresti vedere file con suffissi `_0001`, `_0002` e così via, il che significa i pezzi fisici di un file di log delle transazioni.

Se si desidera utilizzare un file di log delle transazioni a blocchi, assicurarsi di unire i blocchi dopo averli scaricati.

Considera uno scenario in cui hai i seguenti file:

- `202110202230.11.1634769287`
- `202110202230.11.1634769287_0001`
- `202110202230.11.1634769287_0002`
- `202110202230.11.1634769287_metadata`

Il valore del campo `RdsChunkCount` è `3`. L'ordine di unione dei file è il seguente: `202110202230.11.1634769287`, `202110202230.11.1634769287_0001`, `202110202230.11.1634769287_0002`.

- `TransactionLogMetadata` – Contiene informazioni sui metadati su ogni iterazione dell'estrazione del log delle transazioni.

  Il file `RI.End` contiene informazioni per tutti i database a cui sono stati estratti i log delle transazioni e per tutti i database esistenti ma che non hanno i log delle transazioni estratti. Il nome del file `RI.End` segue il pattern *`yyyyMMddHHmm`*`.RI.End.`*`timestamp`*, ad esempio:

```
202110202230.RI.End.1634769281
```

PITR Restore utilizzando l'API AWS Management ConsoleAWS CLI, the o RDS.

Puoi ripristinare un'istanza database RDS Custom per SQL Server a un punto temporale tramite AWS Management Console, AWS CLI o l'API di RDS.

Console

Per ripristinare un'istanza database RDS Custom un punto temporale specifico

1. Accedi a AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, selezionare Automated backups (Backup automatici).
3. Scegli l'istanza database RDS Custom da ripristinare.
4. In Actions (Operazioni), scegli Restore to point in time (Ripristina a un istante temporale).

   Viene visualizzata la finestra Restore to point in time (Ripristina a un istante temporale).

5. Scegliere Latest restorable time (Ultimo orario di ripristino) per eseguire il ripristino in base al momento più recente oppure scegliere Custom (Personalizzato) per scegliere una data e un'ora.

   Se scegli Personalizzato, specifica la data e l'ora in cui desideri ripristinare l'istanza.

   Gli orari vengono visualizzati nel fuso orario locale, indicato come un offset dell'ora UTC (Coordinated Universal Time). Ad esempio, UTC-5 è l'orario standard degli Stati Uniti orientali/ora legale degli Stati Uniti centrali.
6. Per DB Instance Identifier (Identificatore istanze database), inserire il nome dell'istanza database RDS Custom di destinazione ripristinata. Il nome deve essere univoco.
7. Scegli altre opzioni in base alle esigenze, ad esempio la classe di istanza database.
8. Scegli Restore to point in time (Ripristina per punto nel tempo).

AWS CLI

È possibile ripristinare un'istanza DB a un'ora specificata utilizzando il point-in-time AWS CLI comando restore-db-instance-to- per creare una nuova istanza DB personalizzata RDS.

Utilizzare una delle opzioni seguenti per specificare il backup da cui effettuare il ripristino:

- `--source-db-instance-identifier` *`mysourcedbinstance`*
- `--source-dbi-resource-id` *`dbinstanceresourceID`*
- `--source-db-instance-automated-backups-arn` *`backupARN`*

L'opzione `custom-iam-instance-profile` è obbligatoria.

Il seguente esempio ripristina `my-custom-db-instance` a una nuova istanza database denominata `my-restored-custom-db-instance`, a partire dal tempo specificato.

Example

Per LinuxmacOS, oUnix:

```
aws rds restore-db-instance-to-point-in-time \
    --source-db-instance-identifier my-custom-db-instance\
    --target-db-instance-identifier my-restored-custom-db-instance \
    --custom-iam-instance-profile AWSRDSCustomInstanceProfileForRdsCustomInstance \
    --restore-time 2022-10-14T23:45:00.000Z
```

Per Windows:

```
aws rds restore-db-instance-to-point-in-time ^
    --source-db-instance-identifier my-custom-db-instance ^
    --target-db-instance-identifier my-restored-custom-db-instance ^
    --custom-iam-instance-profile AWSRDSCustomInstanceProfileForRdsCustomInstance ^
    --restore-time 2022-10-14T23:45:00.000Z
```

Eliminazione di una snapshot RDS Custom per SQL Server

Elimina le snapshot database di RDS Custom per SQL Server quando non ti occorrono più. La procedura di eliminazione è la stessa per le istanze database Amazon RDS e RDS Custom.

Le snapshot Amazon EBS per i volumi binari e root rimangono nel tuo account per un periodo più lungo perché potrebbero essere collegate ad alcune istanze in esecuzione nel tuo account o ad altre snapshot RDS Custom per SQL Server. Queste snapshot EBS vengono eliminate automaticamente dopo che non sono più correlate a risorse RDS Custom per SQL Server esistenti (istanze database o backup).

Console

Per eliminare una snapshot di un'istanza database RDS Custom

1. Accedi alla AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel pannello di navigazione, selezionare Snapshots (Snapshot).
3. Scegliere la snapshot DB da eliminare.
4. Per Actions (Operazioni), scegliere Delete Snapshot (Elimina snapshot).
5. Nella pagina di conferma, scegliere Delete (Elimina).

AWS CLI

Per eliminare un'istantanea personalizzata RDS, utilizzare il AWS CLI comando. delete-db-snapshot

Si richiede la seguente opzione:

- `--db-snapshot-identifier` – La snapshot da eliminare

L'esempio seguente elimina la snapshot database `my-custom-snapshot`.

Example

PerLinux, omacOS: Unix

```
aws rds delete-db-snapshot \
  --db-snapshot-identifier my-custom-snapshot
```

Per Windows:

```
aws rds delete-db-snapshot ^
  --db-snapshot-identifier my-custom-snapshot
```

Eliminazione di backup automatici RDS Custom per SQL Server

Puoi eliminare i backup automatici mantenuti per RDS Custom per SQL Server quando non servono più. La procedura è la stessa della procedura per l'eliminazione dei backup Amazon RDS.

Console

Per eliminare i backup automatici mantenuti

1. Accedi alla AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, selezionare Automated backups (Backup automatici).
3. ScegliereRetained (Mantenuti).
4. Scegliere il backup automatico mantenuto da eliminare.
5. In Actions (Azioni), selezionare Delete (Elimina).
6. Nella pagina di conferma, immetti **`delete me`** e seleziona Elimina.

AWS CLI

È possibile eliminare un backup automatico conservato utilizzando il AWS CLI comando delete-db-instance-automated-backup.

La seguente opzione viene utilizzata per eliminare un backup automatico mantenuto:

- `--dbi-resource-id` – L'identificatore della risorsa per l'istanza database RDS Custom di origine.

È possibile trovare l'identificatore di risorsa per l'istanza DB di origine di un backup automatizzato mantenuto utilizzando il comando -backups. AWS CLI describe-db-instance-automated

Il seguente esempio elimina il backup automatico mantenuto con l'identificatore della risorsa di istanza DB source `custom-db-123ABCEXAMPLE`.

Example

PerLinux, o: macOS Unix

```
aws rds delete-db-instance-automated-backup \
    --dbi-resource-id custom-db-123ABCEXAMPLE
```

Per Windows:

```
aws rds delete-db-instance-automated-backup ^
    --dbi-resource-id custom-db-123ABCEXAMPLE
```

Migrazione di un database On-Premise ad Amazon RDS Custom per SQL Server

È possibile utilizzare il seguente processo per migrare un database Microsoft SQL Server On-Premise in Amazon RDS Custom per SQL Server utilizzando ripristino e backup nativi:

1. Effettua un backup completo del database sull'istanza database On-Premise.
2. Carica il file di backup su Amazon S3.
3. Scarica il file di backup da S3 nell'istanza database RDS Custom per SQL Server.
4. Ripristina un database utilizzando il file di backup scaricato sull'istanza DB RDS Custom per SQL Server.

Questo processo spiega la migrazione di un database da locale a RDS Custom per SQL Server, utilizzando ripristino e backup completamente nativi. Per ridurre il tempo di cutover durante il processo di migrazione, è possibile anche considerare l'utilizzo di backup di log o differenziali.

Per informazioni generali su ripristino e backup nativi per RDS per SQL Server, consulta Importazione ed esportazione di database SQL Server mediante backup e ripristino nativi.

Argomenti

Prerequisiti

Esegui le seguenti attività prima di eseguire la migrazione del database:

1. Configura Remote Desktop Connection (RDP) per l'istanza database RDS Custom per SQL Server. Per ulteriori informazioni, consultare Connessione all'istanza database RDS Custom tramite RDP.
2. Configura l'accesso ad Amazon S3 in modo da poter caricare e scaricare il file di backup del database. Per ulteriori informazioni, consultare Integrazione di un'istanza database Amazon RDS for SQL Server con Amazon S3.

Backup del database On-Premise

È possibile utilizzare il backup nativo di SQL Server per eseguire un backup completo del database sull'istanza database On-Premise.

L'esempio seguente mostra un backup di un database denominato `mydatabase`, con l'opzione `COMPRESSION` specificata per ridurre le dimensioni del file di backup.

Per eseguire il backup del database On-Premise

1. Utilizzando SQL Server Management Studio (SSMS), connettersi all'istanza di SQL Server On-Premise.
2. Esegui il seguente comando T-SQL.

```
backup database mydatabase to
disk ='C:\Program Files\Microsoft SQL Server\MSSQL13.MSSQLSERVER\MSSQL\Backup\mydb-full-compressed.bak'
with compression;
```

Caricamento del file di backup su Amazon S3

Utilizza AWS Management Console per caricare il file di backup `mydb-full-compressed.bak` su Amazon S3.

Per caricare il file di backup su S3

1. Accedi alla AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. In Buckets (Bucket) selezionare il nome del bucket in cui si desidera caricare il file di backup.
3. Scegli Carica.
4. Nella finestra Carica completa una delle seguenti operazioni:

   - Trascina e rilascia `mydb-full-compressed.bak` nella finestra Upload (Carica).
   - Scegli Add file (Aggiungi file), scegli `mydb-full-compressed.bak`, quindi scegli Open (Apri).

   Amazon S3 caricherà il file di backup come oggetto S3. Al termine del caricamento, sarà visualizzato un messaggio di successo nella pagina Carica: stato .

Download del file di backup da Amazon S3

Utilizza la console per scaricare il file di backup da S3 nell'istanza database di RDS Custom per SQL Server.

Per scaricare il file di backup da S3

1. Utilizzando RDP, connettersi all'istanza database RDS Custom per SQL Server.
2. Accedi alla AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
3. Nell'elenco Buckets (Bucket) selezionare il nome del bucket che contiene il file di backup.
4. Scegli il file di backup `mydb-full-compressed.bak`.
5. Per Actions (Operazioni), scegliere Download as (Scarica come).
6. Aprire il menu contestuale (clic con il tasto destro del mouse) per il collegamento fornito, quindi scegliere Save As (Salva come).
7. Salva il file `mydb-full-compressed.bak` nella directory `D:\rdsdbdata\BACKUP`.

Ripristino del file di backup nell'istanza database RDS Custom per SQL Server

Utilizzare il ripristino nativo di SQL Server per ripristinare il file di backup nell'istanza database RDS Custom per SQL Server.

In questo esempio, l'opzione `MOVE` è specificata perché le directory dei dati e dei file di log sono diverse dall'istanza database On-Premise.

Per ripristinare il file di backup

1. Utilizzando SSMS, connettersi all'istanza database RDS Custom per SQL Server.
2. Esegui il seguente comando T-SQL.

```
restore database mydatabase from disk='D:\rdsdbdata\BACKUP\mydb-full-
compressed.bak'
with move 'mydatabase' to 'D:\rdsdbdata\DATA\mydatabase.mdf',
move 'mydatabase_log' to 'D:\rdsdbdata\DATA\mydatabase_log.ldf';
```

Aggiornamento di un'istanza database per Amazon RDS Custom for SQL Server

È possibile aggiornare un'istanza database Amazon RDS Custom per SQL Server modificandola per utilizzare una nuova versione del motore DB, come avviene per Amazon RDS.

Le stesse limitazioni per l'aggiornamento di un'istanza database RDS Custom per SQL Server valgono per la modifica di un'istanza database RDS Custom per SQL Server in generale. Per ulteriori informazioni, consulta Modifica di un'istanza database RDS Custom per SQL Server.

Per informazioni generali sull'aggiornamento delle istanze database, consulta Aggiornamento della versione del motore di un'istanza database.

Se esegui l'upgrade di un'istanza DB RDS Custom for SQL Server in una distribuzione Multi-AZ, Amazon RDS esegue gli aggiornamenti in sequenza, quindi si verifica un'interruzione solo per la durata di un failover. Per ulteriori informazioni, consulta Considerazioni su Multi-AZ e sull'ottimizzazione in memoria.

Aggiornamenti di una versione principale

Amazon RDS Custom for SQL Server attualmente supporta i seguenti aggiornamenti di versione principali.

Versione corrente	Versioni supportate per l'aggiornamento
SQL Server 2019	SQL Server 2022

È possibile utilizzare una AWS CLI query, come nell'esempio seguente, per trovare gli aggiornamenti disponibili per una particolare versione del motore di database.

Example

Per LinuxmacOS, oUnix:

```
aws rds describe-db-engine-versions \
    --engine sqlserver-se \
    --engine-version 15.00.4322.2.v1 \
    --query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" \
    --output table
```

Per Windows:

```
aws rds describe-db-engine-versions ^
    --engine sqlserver-se ^
    --engine-version 15.00.4322.2.v1 ^
    --query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" ^
    --output table
```

Livello di compatibilità del database

È possibile utilizzare i livelli di compatibilità del database Microsoft SQL Server per modificare alcuni comportamenti del database in modo da emulare versioni precedenti di SQL Server. Per ulteriori informazioni, consulta Livello di compatibilità nella documentazione Microsoft.

Quando aggiorni l'istanza database, tutti i database esistenti rimangono impostati sul livello di compatibilità originale. Ad esempio, se si esegue l'aggiornamento da SQL Server 2019 a SQL Server 2022, tutti i database esistenti hanno un livello di compatibilità di 150. Tutti i nuovi database creati dopo l'aggiornamento hanno il livello di compatibilità 160.

È possibile modificare il livello di compatibilità di un database tramite il comando ALTER DATABASE. Ad esempio, per modificare un database denominato in `customeracct` modo che sia compatibile con SQL Server 2022, esegui il seguente comando:

```
ALTER DATABASE customeracct SET COMPATIBILITY_LEVEL = 160
```

Risoluzione dei problemi relativi ai database di Amazon RDS Custom per SQL Server

Il modello di responsabilità condivisa di RDS Custom fornisce l'accesso a livello di shell al sistema operativo e l'accesso come amministratore al database. RDS Custom esegue risorse nel proprio account, a differenza di Amazon RDS, che esegue le risorse in un account di sistema. Con un maggiore accesso si ottiene una maggiore responsabilità. Nelle sezioni seguenti sono descritte le procedure di risoluzione dei problemi relativi alle istanze database Amazon RDS Custom per SQL Server.

> **Note**
>
> Questa sezione spiega come risolvere i problemi relativi a RDS Custom per SQL Server. Per informazioni sulla risoluzione dei problemi relativi a RDS Custom per Oracle, consulta Risoluzione dei problemi relativi ai database di Amazon RDS Custom per Oracle.

Argomenti

Visualizzazione di eventi RDS Custom

La procedura per visualizzare gli eventi è la stessa per le istanze database Amazon RDS e RDS Custom. Per ulteriori informazioni, consulta Visualizzazione di eventi Amazon RDS.

Per visualizzare la notifica degli eventi RDS Custom utilizzando il AWS CLI, utilizzare il `describe-events` comando. RDS Custom presenta diversi nuovi eventi. Le categorie di eventi sono le stesse di Amazon RDS. Per l'elenco di eventi, consultare Categorie di eventi Amazon RDS e messaggi di evento.

Nell'esempio seguente vengono recuperati i dettagli per gli eventi verificati per l'istanza database RDS Custom specificata.

```
aws rds describe-events \
    --source-identifier my-custom-instance \
    --source-type db-instance
```

Iscrizione agli eventi RDS Custom

La procedura per visualizzare gli eventi è la stessa per le istanze database Amazon RDS e RDS Custom. Per ulteriori informazioni, consulta Sottoscrizione alle notifiche eventi di Amazon RDS.

Per abbonarsi alle notifiche degli eventi RDS Custom utilizzando la CLI, utilizza il comando `create-event-subscription`. Includi i parametri obbligatori seguenti:

- `--subscription-name`
- `--sns-topic-arn`

Nell'esempio seguente viene creata una sottoscrizione per gli eventi di backup e ripristino per un'istanza database RDS Custom nell'account AWS attuale. Le notifiche sono inviate a un argomento Amazon Simple Notification Service (Amazon SNS) specificato da `--sns-topic-arn`.

```
aws rds create-event-subscription \
    --subscription-name my-instance-events \
    --source-type db-instance \
    --event-categories '["backup","recovery"]' \
    --sns-topic-arn arn:aws:sns:us-east-1:123456789012:interesting-events
```

Risoluzione degli errori della CEV per RDS Custom per SQL Server

La creazione di una CEV potrebbe non riuscire. In questo caso, RDS Custom invia il messaggio dell'evento `RDS-EVENT-0198`. Per ulteriori informazioni sulla visualizzazione degli eventi RDS, consulta Categorie di eventi Amazon RDS e messaggi di evento.

Utilizza le seguenti informazioni per individuare le possibili cause.

Messaggio	Suggerimenti sulla risoluzione dei problemi		
`Custom Engine Version creation expected a`	Esegui Sysprep sull'istanza EC2 creata dall'AMI. Per ulteriori		

Messaggio	Suggerimenti sulla risoluzione dei problemi		
`Sysprep'd AMI. Retry creation using a Sysprep'd AMI.`	informazioni sulla preparazione di un'immagine AMI utilizzando Sysprep, consulta Creare un'immagine Amazon Machine Image (AMI) standardizzata utilizzando Sysprep.		
`EC2 Image permissions for image (AMI_ID) weren't found for customer (Customer_ID). Verify customer (Customer_ID) has valid permissions on the EC2 Image.`	Verifica che l'account e il profilo utilizzati per la creazione dispongano delle autorizzazioni necessarie per `create EC2 Instance` e `Describe Images` per l'AMI selezionata.		
`Failed to rebuild databases with server collation (collation name) due to missing setup.exe file for SQL Server.`	Verifica che il file `setup` sia disponibile in `C:\Program Files\Microsoft SQL Server\nnn\Setup Bootstrap\SQLnnnn\setup.exe` .		
`Image (AMI_ID) doesn't exist in your account (ACCOUNT_ID). Verify (ACCOUNT_ID) is the owner of the EC2 image.`	Assicurati che l'AMI sia presente nello stesso account cliente.		
`Image id (AMI_ID) isn't valid. Specify a valid image id, and try again.`	Il nome dell'AMI non è corretto. Assicurati che venga fornito l'ID AMI corretto.		

Messaggio	Suggerimenti sulla risoluzione dei problemi		
`Image (AMI_ID) operating system platform isn't supported. Specify a valid image, and try again.`	Scegli un'AMI supportata con Windows Server con SQL Server Enterprise, Standard o Web Edition. Scegli un'AMI con uno dei seguenti codici operativi di utilizzo dal Marketplace EC2: • RunInstancesIscrizione agli eventi personalizzati RDS ----sep----:0102 - Windows con SQL Server Enterprise • RunInstances:0102 - Windows con SQL Server Enterprise ----sep----:0006 - Windows con SQL Server Standard • RunInstances:0006 - Windows con SQL Server Standard ----sep----:0202 - Windows con SQL Server Web		
`SQL Server Web Edition isn't supported for creating a Custom Engine Version using Bring Your Own Media. Specify a valid image, and try again.`	Utilizzare un'AMI contenente un'edizione supportata di SQL Server. Per ulteriori informazioni, consulta Supporto delle versioni per le CEV di RDS Custom per SQL Server.		
`The custom engine version can't be the same as the OEV engine version. Specify a valid CEV, and try again.`	Le versioni classiche del motore RDS Custom per SQL Server non sono supportate. Ad esempio, la versione 15.00.4073.23.v1. Utilizza un numero di versione supportato.		

Messaggio	Suggerimenti sulla risoluzione dei problemi		
`The custom engine version isn't in an active state. Specify a valid CEV, and try again.`	La CEV deve essere nello stato `AVAILABLE` per poter completare l'operazione. Modifica la CEV da `INACTIVE` a `AVAILABLE` .		
`The custom engine version isn't valid for an upgrade. Specify a valid CEV with an engine version greater or equal to (X), and try again.`	La CEV di destinazione non è valida. Verifica i requisiti per un percorso di aggiornamento valido.		
`The custom engine version isn't valid. Names can include only lowercase letters (a-z), dashes (-), underscores (_), and periods (.). Specify a valid CEV, and try again.`	Segui la convenzione di denominazione della CEV richiesta. Per ulteriori informazioni, consulta Requisiti per le CEV per RDS Custom per SQL Server.		
`The custom engine version isn't valid. Specify valid database engine version, and try again. Example: 15.00.4073.23-cev123.`	È stata fornita una versione del motore di database non supportata. Usa una versione del motore di database supportata.		
`The expected architecture is (X) for image (AMI_ID), but architecture (Y) was found.`	Usa un'AMI basata sull'architettura x86_64.		

Messaggio	Suggerimenti sulla risoluzione dei problemi
`The expected owner of image (AMI_ID) is customer account ID (ACCOUNT_ID), but owner (ACCOUNT_ID) was found.`	Crea l'istanza EC2 dall'AMI di cui disponi dell'autorizzazione. Esegui Sysprep sull'istanza EC2 per creare e salvare un'immagine di base.
`The expected platform is (X) for image (AMI_ID), but platform (Y) was found.`	Usa un'AMI creata con la piattaforma Windows.
`The expected root device type is (X) for image %s, but root device type (Y) was found.`	Crea l'AMI con il tipo di dispositivo EBS.
`The expected SQL Server edition is (X), but (Y) was found.`	Scegli un'AMI supportata con Windows Server con SQL Server Enterprise, Standard o Web Edition. Scegli un'AMI con uno dei seguenti codici operativi di utilizzo dal Marketplace EC2: • RunInstances:0202 - Windows con SQL Server Web ----sep----:0102 - Windows con SQL Server Enterprise • RunInstances:0102 - Windows con SQL Server Enterprise ----sep----:0006 - Windows con SQL Server Standard • RunInstances:0006 - Windows con SQL Server Standard ----sep----:0202 - Windows con SQL Server Web

Messaggio	Suggerimenti sulla risoluzione dei problemi		
`The expected state is (X) for image (AMI_ID), but the following state was found: (Y).`	Assicurati che l'AMI sia nello stato `AVAILABLE` .		
`The provided Windows OS name (X) isn’t valid. Make sure the OS is one of the following: (Y).`	Usa un sistema operativo Windows supportato.		
`Unable to find bootstrap log file in path.`	Verifica che il file di log sia disponibile in `C:\Program Files\Microsoft SQL Server\nnn\Setup Bootstrap\Log\Summary.txt` .		
`RDS expected a Windows build version greater than or equal to (X), but found version (Y)..`	Usa un'AMI con la versione di build del sistema operativo minima 14393.		
`RDS expected a Windows major version greater than or equal to (X), but found version (Y)..`	Usa un'AMI con la versione principale del sistema operativo minima 10.0 o successiva.		

Correzione delle configurazioni non supportate in RDS Custom per SQL Server

In base al modello di responsabilità condivisa, è tua responsabilità risolvere i problemi di configurazione che comportano il passaggio dell'istanza database RDS Custom per SQL Server allo stato `unsupported-configuration`. Se il problema riguarda l' AWS infrastruttura, puoi utilizzare la console o AWS CLI risolverlo. Se il problema riguarda il sistema operativo o la configurazione del database, è possibile accedere all'host per risolverlo.

> **Note**
>
> Questa sezione spiega come correggere le configurazioni non supportate in RDS Custom per SQL Server. Per ulteriori informazioni su RDS Custom per Oracle, consulta Correzione delle configurazioni non supportate in RDS Custom per Oracle.

Nella tabella seguente puoi trovare le descrizioni delle notifiche e degli eventi inviati dal perimetro di supporto e come risolverli. Queste notifiche e il perimetro di supporto sono soggetti a modifiche. Per informazioni sul perimetro del supporto, consulta Perimetro di supporto RDS Custom. Per le descrizioni degli eventi, consulta Categorie di eventi Amazon RDS e messaggi di evento.

Codice evento	Area di configurazione	Messaggio dell'evento RDS	Processo di convalida
SP-S0000	Configurazione manuale non supportata	Lo stato dell'istanza RDS Custom DB è impostato su [Configurazione non supportata] a causa di:. X	Per risolvere questo problema, crea un caso di supporto.
AWS Risorsa (infrastruttura)			
SP-S1001	Stato dell'istanza EC2	Lo stato dell'istanza RDS Custom DB è impostato su [Configurazione non supportata] perché: l'istanza EC2 sottostante %s è stata interrotta senza interrompere l'istanza RDS. Puoi risolvere questo problema avviando l'istanza	Per verificare lo stato di un'istanza DB, usa la console o esegui il seguente comando: AWS CLI `aws rds describe-db-instances \ --db-instance-identifier db-instance-name |grep DBInstanceStatus`

Codice evento	Area di configurazione	Messaggio dell'evento RDS	Processo di convalida
		EC2 sottostante e assicurandoti che i volumi binari e di dati siano collegati. Se intendi interrompere l'istanza RDS, assicurati che l'istanza EC2 sottostante sia prima nello stato AVAILABLE, quindi usa la console RDS o la CLI per interrompere l'istanza RDS.	

Codice evento	Area di configurazione	Messaggio dell'evento RDS	Processo di convalida
SP-S1002	Stato dell'istanza EC2	Lo stato dell'istanza DB RDS Custom è impostato su [Configurazione non supportata] perché: Lo stato dell'istanza DB RDS è impostato su STOPPED ma l'istanza EC2 sottostante %s è stata avviata. Puoi risolvere questo problema interrompendo l'istanza EC2 sottostante. Se intendi avviare l'istanza RDS, utilizza la console o la CLI.	Utilizzate il seguente AWS CLI comando per controllare lo stato di un'istanza DB: `aws rds describe-db-instances \ --db-instance-identifier *db-instance-name* |grep DBInstanceStatus` Puoi anche controllare lo stato dell'istanza EC2 utilizzando la console EC2. Per avviare un'istanza DB, usa la console o esegui il seguente AWS CLI comando: `aws rds start-db-instance \ --db-instance-identifier *db-instance-name*`

Codice evento	Area di configurazione	Messaggio dell'evento RDS	Processo di convalida
SP-S1003	Classe di istanza EC2	Lo stato dell'istanza DB personalizzata RDS è impostato su [Configurazione non supportata] a causa di: C'è una mancata corrispondenza tra la classe di istanza DB prevista e quella configurata dell'host EC2. Puoi risolvere questo problema modificando la classe dell'istanza DB riportandola al tipo di classe originale.	Usa il seguente comando CLI per controllare la classe di istanza DB prevista: `aws rds describe-db-instances \ --db-instance-identifier db-instance-name |grep DBInstanceClass`
SP-S1004	Volume di archiviazione EBS non accessibile	Lo stato dell'istanza DB personalizzata RDS è impostato su [Configurazione non supportata] perché: il volume di archiviazione EBS originale %s associato all'istanza EC2 non è attualmente accessibile.	

Codice evento	Area di configurazione	Messaggio dell'evento RDS	Processo di convalida
SP-S1005	Volume di storage EBS distaccato	Lo stato dell'istanza DB personalizzata RDS è impostato su [Configurazione non supportata] perché: il volume di archiviazione EBS originale «volume-id» non è collegato. Puoi risolvere questo problema collegando il volume EBS associato all'istanza EC2.	Dopo aver ricollegato il volume EBS, utilizza i seguenti comandi CLI per verificare se il volume EBS 'volume-id' è collegato correttamente all'istanza RDS: `aws ec2 describe-volumes \ --volume-ids volume-id |grep InstanceId`

Codice evento	Area di configurazione	Messaggio dell'evento RDS	Processo di convalida
SP-S1006	Dimensione del volume di archiviazione EBS	Lo stato dell'istanza DB personalizzata RDS è impostato su [Configurazione non supportata] a causa di: C'è una mancata corrispondenza tra le impostazioni previste e quelle configurate del volume di archiviazione EBS «volume-id». La dimensione del volume è stata modificata manualmente a livello EC2 rispetto ai valori originali di [%s]. Per risolvere questo problema, crea una richiesta di supporto.	Utilizza il seguente comando CLI per confrontare la dimensione del volume dei dettagli 'volume-id' del volume EBS e i dettagli dell'istanza RDS: `aws rds describe-db-instances \ --db-instance-identifier *db-instance-name* |grep AllocatedStorage` Utilizzate il seguente comando CLI per visualizzare la dimensione effettiva del volume allocato: `aws ec2 describe-volumes \ --volume-ids |grep Size`

Codice evento	Area di configurazione	Messaggio dell'evento RDS	Processo di convalida
SP-S1007	Configurazione del volume di archiviazione EBS	Lo stato dell'istanza DB personalizzata RDS è impostato su [Configurazione non supportata] a causa di: C'è una mancata corrispondenza tra le impostazioni previste e quelle configurate del volume di archiviazione EBS «volume-id». Puoi risolvere questo problema modificando la configurazione del volume di archiviazione EBS [IOPS, Throughput, Volume type] ai valori originali di [IOPS: %s, Throughput: %s, tipo di volume: %s] a livello EC2. Per future modifiche allo storage, utilizza la console RDS o la CLI. La dimensione del volume è stata inoltre modificata manualmente a livello di EC2 rispetto ai valori	Utilizza il seguente comando CLI per confrontare il tipo di volume dei dettagli 'volume-id' del volume EBS e i dettagli dell'istanza RDS. Assicurati che i valori a livello EBS corrispondano ai valori a livello RDS: `aws rds describe-db-instances \ --db-instance-identifier db-instance-name |grep StorageType` Per ottenere il valore previsto per Storage Throughput a livello RDS: `aws rds describe-db-instances \ --db-instance-identifier db-instance-name |grep StorageThroughput` Per ottenere il valore previsto per Volume IOPS a livello RDS: `aws rds describe-db-instances \ --db-instance-identifier db-instance-name |grep Iops` Per ottenere il tipo di storage corrente a livello EC2: `aws ec2 describe-volumes \`

Codice evento	Area di configura zione	Messaggio dell'even to RDS	Processo di convalida
		originali di [%s]. Per risolvere questo problema, crea una richiesta di supporto.	`--volume-ids |grep VolumeTyp e` Per ottenere il valore corrente per il throughput di storage a livello EC2: `aws ec2 describe-volumes \ --volume-ids |grep Throughpu t` Per ottenere il valore corrente per Volume IOPS a livello EC2: `aws ec2 describe-volumes \ --volume-ids |grep Iops`

Codice evento	Area di configurazione	Messaggio dell'evento RDS	Processo di convalida
SP-S1008	Dimensioni e configurazione del volume di storage EBS	Lo stato dell'istanza DB personalizzata RDS è impostato su [Configurazione non supportata] a causa di: C'è una mancata corrispondenza tra le impostazioni previste e quelle configurate del volume di archiviazione EBS «volume-id». Puoi risolvere questo problema modificando la configurazione del volume di archiviazione EBS [IOPS, Throughput, Volume type] ai valori originali di [IOPS: %s, Throughput: %s, tipo di volume: %s] a livello EC2. Per future modifiche allo storage, utilizza la console RDS o la CLI. La dimensione del volume è stata inoltre modificata manualmente a livello di EC2 rispetto ai valori	Utilizza il seguente comando CLI per confrontare il tipo di volume dei dettagli 'volume-id' del volume EBS e i dettagli dell'istanza RDS. Assicurati che i valori a livello EBS corrispondano ai valori a livello RDS: `aws rds describe-db-instances \ --db-instance-identifier db-instance-name |grep StorageType` Per ottenere il valore previsto per Storage Throughput a livello RDS: `aws rds describe-db-instances \ --db-instance-identifier db-instance-name |grep StorageThroughput` Per ottenere il valore previsto per Volume IOPS a livello RDS: `aws rds describe-db-instances \ --db-instance-identifier db-instance-name |grep Iops` Per ottenere il tipo di storage corrente a livello EC2: `aws ec2 describe-volumes \`

Codice evento	Area di configurazione	Messaggio dell'evento RDS	Processo di convalida
		originali di [%s]. Per risolvere questo problema, crea una richiesta di supporto.	`--volume-ids |grep VolumeType` Per ottenere il valore corrente per il throughput di storage a livello EC2: `aws ec2 describe-volumes \ --volume-ids |grep Throughput` Per ottenere il valore corrente per Volume IOPS a livello EC2: `aws ec2 describe-volumes \ --volume-ids |grep Iops` Per ottenere la dimensione prevista del volume allocato: `aws rds describe-db-instances \ --db-instance-identifier *db-instance-name* |grep AllocatedStorage` Per ottenere la dimensione effettiva del volume allocato: `aws ec2 describe-volumes \ --volume-ids |grep Size`

Codice evento	Area di configurazione	Messaggio dell'evento RDS	Processo di convalida
SP-S1009	Autorizzazioni SQS	Lo stato dell'istanza RDS Custom DB è impostato su [Configurazione non supportata] a causa di: mancano le autorizzazioni di Amazon Simple Queue Service (SQS) per il profilo dell'istanza IAM. Puoi risolvere questo problema assicurandoti che il profilo IAM associato all'host disponga delle seguenti autorizzazioni: ["SQS: «, "SQS: SendMessage «, "SQS: «, "SQS: ReceiveMessage «]. DeleteMessage GetQueueUrl	

Codice evento	Area di configurazione	Messaggio dell'evento RDS	Processo di convalida
SP-S1010	Endpoint VPC SQS	Lo stato dell'istanza DB personalizzata RDS è impostato su [Configurazione non supportata] per: Una policy degli endpoint VPC blocca le operazioni di Amazon Simple Queue Service (SQS). Puoi risolvere questo problema modificando la policy degli endpoint VPC per consentire le azioni SQS richieste.	
Sistema operativo			

Codice evento	Area di configurazione	Messaggio dell'evento RDS	Processo di convalida
SP-S2001	Stato del servizio SQL	Lo stato dell'istanza DB personalizzata RDS è impostato su [Configurazione non supportata] a causa di: Il servizio SQL Server non è avviato. È possibile risolvere questo problema riavviando il servizio SQL Server sull'host. Se l'istanza DB è un'istanza DB Multi-AZ e il riavvio non riesce, interrompi e avvia l'host per avviare un failover.	

Codice evento	Area di configurazione	Messaggio dell'evento RDS	Processo di convalida
SP-S2002	Stato dell'agente personalizzato RDS	Lo stato dell'istanza RDS Custom DB è impostato su [Configurazione non supportata] per: Il servizio RDS Custom Agent non è installato o non può essere avviato. Puoi risolvere questo problema esaminando il registro eventi di Windows per determinare il motivo per cui il servizio non si avvia e adottando le misure appropriate per risolvere il problema. Per ulteriore assistenza, crea una richiesta di supporto.	

Codice evento	Area di configurazione	Messaggio dell'evento RDS	Processo di convalida
SP-S1009	Autorizzazioni SQS	Lo stato dell'istanza RDS Custom DB è impostato su [Configurazione non supportata] a causa di: mancano le autorizzazioni di Amazon Simple Queue Service (SQS) per il profilo dell'istanza IAM. Puoi risolvere questo problema assicurandoti che il profilo IAM associato all'host disponga delle seguenti autorizzazioni: ["SQS: «, "SQS: SendMessage «, "SQS: «, "SQS: ReceiveMessage «]. DeleteMessage GetQueueUrl	

Codice evento	Area di configurazione	Messaggio dell'evento RDS	Processo di convalida
SP-S1010	Endpoint VPC SQS	Lo stato dell'istanza DB personalizzata RDS è impostato su [Configurazione non supportata] per: Una policy degli endpoint VPC blocca le operazioni di Amazon Simple Queue Service (SQS). Puoi risolvere questo problema modificando la policy degli endpoint VPC per consentire le azioni SQS richieste.	
Sistema operativo			

Codice evento	Area di configurazione	Messaggio dell'evento RDS	Processo di convalida
SP-S2001	Stato del servizio SQL	Lo stato dell'istanza DB personalizzata RDS è impostato su [Configurazione non supportata] a causa di: Il servizio SQL Server non è avviato. È possibile risolvere questo problema riavviando il servizio SQL Server sull'host. Se l'istanza DB è un'istanza DB Multi-AZ e il riavvio non riesce, interrompi e avvia l'host per avviare un failover.	

Codice evento	Area di configurazione	Messaggio dell'evento RDS	Processo di convalida
SP-S2002	Stato dell'agente personalizzato RDS	Lo stato dell'istanza RDS Custom DB è impostato su [Configurazione non supportata] per: Il servizio RDS Custom Agent non è installato o non può essere avviato. Puoi risolvere questo problema esaminando il registro eventi di Windows per determinare il motivo per cui il servizio non si avvia e adottando le misure appropriate per risolvere il problema. Per ulteriore assistenza, crea una richiesta di supporto.	Accedere all'host e assicurarsi che l'agente RDS Custom sia in esecuzione. È possibile utilizzare i seguenti comandi per visualizzare lo stato dell'agente. `$name = "RDSCustomAgent"` `$service = Get-Service $name` `Write-Host $service.Status` Se lo stato non è `Running`, è possibile avviare il servizio con il comando seguente: `Start-Service $name` Se l'agente non riesce ad avviarsi, controlla gli eventi di Windows per scoprire perché non può avviarsi. L'agente richiede un utente Windows per avviare il servizio. Assicurati che un utente Windows esista e disponga dei privilegi per eseguire il servizio.

Codice evento	Area di configurazione	Messaggio dell'evento RDS	Processo di convalida
SP-S2003	Stato dell'agente SSM	Lo stato dell'istanza RDS Custom DB è impostato su [Configurazione non supportata] a causa di: Il servizio Amazon SSM Agent non è raggiungibile. Puoi risolvere questo problema controllando lo stato del servizio con il `Get-Service AmazonSSMAgent` PowerShell comando o avviando il servizio con. `Start-Service AmazonSSMAgent` Assicurati che il traffico HTTPS (porta 443) in uscita verso gli endpoint regionali ssm, ssmmessages ed ec2messages sia consentito.	Per ulteriori informazioni, consulta Risoluzione dei problemi relativi all'SSM Agent. Per risolvere i problemi relativi agli endpoint SSM, consulta Impossibile connettersi agli endpoint SSM e Utilizzare ssm-cli per risolvere i problemi relativi alla disponibilità dei nodi gestiti.

Codice evento	Area di configurazione	Messaggio dell'evento RDS	Processo di convalida
SP-S2004	Accesso all'agente personalizzato RDS	SP-S2004Lo stato dell'istanza RDS Custom DB è impostato su [Configurazione non supportata] a causa di: Si è verificato un problema imprevisto con l'accesso SQL. "$HOSTNAME/RDSAgent" Per risolvere questo problema, crea un caso di supporto.	

Codice evento	Area di configurazione	Messaggio dell'evento RDS	Processo di convalida
SP-S2005	Fuso orario	Lo stato dell'istanza DB personalizzata RDS è impostato su [Configurazione non supportata] a causa di: Il fuso orario sull'istanza Amazon EC2 [%s] è stato modificato. Puoi risolvere questo problema modificando il fuso orario riportandolo all'impostazione specificata durante la creazione dell'istanza. Se desideri creare un'istanza con un fuso orario specifico, consulta la documentazione personalizzata di RDS.	Esegui il `Get-Timezone` PowerShell comando per confermare il fuso orario. Per ulteriori informazioni, consulta Fuso orario locale per le istanze database di RDS Custom for SQL Server.

Codice evento	Area di configurazione	Messaggio dell'evento RDS	Processo di convalida
SP-S2006	Versione della soluzione software ad alta disponibilità	Lo stato dell'istanza DB personalizzata RDS è impostato su [Configurazione non supportata] perché: La soluzione software ad alta disponibilità dell'istanza corrente è diversa dalla versione prevista. Per risolvere questo problema, crea un caso di supporto.	

Codice evento	Area di configurazione	Messaggio dell'evento RDS	Processo di convalida
SP-S2007	Configurazione della soluzione software ad alta disponibilità	Lo stato dell'istanza DB personalizzata RDS è impostato su [Configurazione non supportata] perché: le impostazioni di configurazione della soluzione software ad alta disponibilità sono state modificate in valori imprevisti sull'istanza %s. Per risolvere questo problema, riavvia l'istanza EC2. Quando riavvii l'istanza EC2, aggiorna automaticamente le impostazioni alla configurazione richiesta per la soluzione software ad alta disponibilità.	
Database			

Codice evento	Area di configurazione	Messaggio dell'evento RDS	Processo di convalida
SP-S3001	Protocollo di memoria condivisa SQL Server	Lo stato dell'istanza DB personalizzata RDS è impostato su [Configurazione non supportata] perché: il protocollo di memoria condivisa di SQL Server è disabilitato. È possibile risolvere questo problema abilitando il protocollo di memoria condivisa in SQL Server Configuration Manager.	È possibile convalidarlo selezionando: SQL Server Configuration Manager > Configurazione di rete SQL Server > Protocolli per MSSQLSERVER> Memoria condivisa abilitata. Dopo aver abilitato il protocollo, riavviare il processo di SQL Server.
SP-S3002	Chiave principale del servizio	Lo stato dell'istanza RDS Custom DB è impostato su [Configurazione non supportata] perché: RDS Automation non è in grado di eseguire il backup di Service Master Key (SMK) come parte della nuova generazione SMK. Per risolvere questo problema, crea un caso di supporto.	

Codice evento	Area di configurazione	Messaggio dell'evento RDS	Processo di convalida
SP-S3003	Chiave principale del servizio	Lo stato dell'istanza RDS Custom DB è impostato su [Configurazione non supportata] perché: I metadati relativi alla Service Master Key (SMK) sono mancanti o incompleti. Per risolvere questo problema, crea un caso di supporto.	

Codice evento	Area di configurazione	Messaggio dell'evento RDS	Processo di convalida
SP-S3004	Versione ed edizione di DB Engine	C'è una mancata corrispondenza tra la versione e l'edizione di SQL Server previste e installate. La modifica dell'edizione di SQL Server non è supportata in RDS Custom for SQL Server. Inoltre, la modifica manuale della versione di SQL Server sull'istanza EC2 personalizzata di RDS non è supportata. Per risolvere questo problema, crea una richiesta di supporto.	Esegui la seguente query per ottenere la versione SQL: `select @@version` Esegui il AWS CLI comando seguente per ottenere la versione e l'edizione del motore SQL RDS: `aws rds describe-db-instances \ --db-instance-identifier db-instance-name |grep EngineVersion` `aws rds describe-db-instances \ --db-instance-identifier db-instance-name |grep Engine` Per ulteriori informazioni, consulta Modifica di un'istanza database RDS Custom per SQL Server e Aggiornamento della versione del motore di un'istanza database.

Codice evento	Area di configurazione	Messaggio dell'evento RDS	Processo di convalida
SP-S3005	Edizione DB Engine	L'edizione corrente di SQL Server non corrisponde all'edizione di SQL Server prevista [%s]. La modifica dell'edizione di SQL Server non è supportata in RDS Custom for SQL Server. Per risolvere questo problema, crea una richiesta di supporto.	Esegui la seguente query per ottenere l'edizione SQL: Example `select @@version` Esegui il AWS CLI comando seguente per ottenere l'edizione del motore SQL RDS: `aws rds describe-db-instances \ --db-instance-identifier *db-instance-name* |grep Engine`

Codice evento	Area di configurazione	Messaggio dell'evento RDS	Processo di convalida
SP-S3006	DB Engine Version (Versione motore DB)	La versione corrente di SQL Server non corrisponde alla versione prevista di SQL Server [%s]. Non è possibile modificare manualmente la versione di SQL Server sull'istanza EC2 personalizzata di RDS. Per risolvere questo problema, crea una richiesta di supporto. Per eventuali modifiche future alla versione di SQL Server, puoi modificare l'istanza dalla console AWS RDS o tramite il comando CLI modify-db-instance .	Esegui la seguente query per ottenere la versione SQL: Example `select @@version` Esegui il AWS CLI comando seguente per ottenere la versione del motore SQL RDS: `aws rds describe-db-instances \ --db-instance-identifier db-instance-name |grep EngineVersion` Per ulteriori informazioni, consulta Modifica di un'istanza database RDS Custom per SQL Server e Aggiornamento della versione del motore di un'istanza database.

Codice evento	Area di configurazione	Messaggio dell'evento RDS	Processo di convalida
SP-S3007	Posizione del file del database	Lo stato dell'istanza RDS Custom DB è impostato su [Configurazione non supportata] perché: I file di database sono configurati all'esterno dell'unità D:\. Puoi risolvere questo problema assicurandoti che tutti i file di database, inclusi ROW, LOG, FILESTREAM, ecc... siano archiviati nell'unità D:\.	Esegui la seguente query per elencare la posizione dei file di database che non si trovano nel percorso predefinito: `USE master;` `SELECT physical_name as files_not_in_default_path` `FROM sys.master_files` `WHERE SUBSTRING(physical_name,1,3)!='D:\';`

Risoluzione dei problemi **`Storage-Full`** in RDS Custom for SQL Server

RDS Custom monitora lo spazio disponibile sui volumi root (C) e data (D:)) di un'istanza DB di RDS Custom for SQL Server. RDS Custom sposta lo stato dell'istanza allo `Storage-Full` stato quando uno dei due volumi ha meno di 500 MiB di spazio su disco disponibile. Per scalare lo spazio di archiviazione dell'istanza, vedere. Modifica dell'archiviazione per un'istanza database RDS Custom per Oracle

> **Note**
>
> La risoluzione delle istanze in ingresso `Storage-Full` può richiedere fino a 30 minuti dopo il ridimensionamento dello storage.

Lavorare con Amazon RDS su AWS Outposts

Amazon RDS on AWS Outposts estende RDS per SQL Server, RDS per MySQL e RDS per PostgreSQL agli ambienti. AWS Outposts AWS Outposts utilizza lo stesso hardware utilizzato in pubblico per portare AWS servizi, infrastrutture e modelli Regioni AWS operativi in locale. Con RDS in Outposts, è possibile eseguire il provisioning di istanze DB gestite vicino alle applicazioni aziendali che devono essere eseguite in locale. Per ulteriori informazioni su AWS Outposts, vedere AWS Outposts.

Utilizzi la stessa AWS Management Console API e RDS per il provisioning e la gestione di RDS locali su istanze DB Outposts così come per le istanze DB RDS in esecuzione in. AWS CLI Cloud AWS RDS su Outposts automatizza le attività, come il provisioning del database, l'applicazione di patch del sistema operativo e del database, il backup e l'archiviazione a lungo termine in Amazon S3.

RDS in Outposts supporta i backup automatizzato delle istanze database. La connettività di rete tra Outpost e le tue Regione AWS è necessaria per il backup e il ripristino delle istanze DB. Tutte le istantanee DB e i log delle transazioni di un Outpost vengono archiviati nel tuo. Regione AWS Dalla regione AWS è possibile ripristinare un'istanza database da uno snapshot DB su un Outpost diverso. Per ulteriori informazioni, consulta Introduzione ai backup.

RDS in Outposts supporta la manutenzione e gli aggiornamenti automatici delle istanze database. Per ulteriori informazioni, consulta Manutenzione di un'istanza database.

RDS su Outposts utilizza la crittografia a riposo per le istanze database e gli snapshot DB mediante la AWS KMS key. Per ulteriori informazioni sulla crittografia dei dati inattivi, consultare Crittografia delle risorse Amazon RDS.

Per impostazione predefinita, le istanze EC2 nelle sottoreti di Outposts possono utilizzare il servizio DNS Amazon Route 53 per risolvere i nomi dominio in indirizzi IP. Potresti riscontrare tempi di risoluzione DNS più lunghi con Route 53, a seconda della latenza del percorso tra l'Outpost e la Regione AWS. In questi casi, è possibile utilizzare i server DNS installati in locale nell'ambiente locale. Per ulteriori informazioni, consulta DNS nella Guida per l'utente AWS Outposts .

Quando la connettività di rete a non Regione AWS è disponibile, l'istanza DB continua a funzionare localmente. Puoi continuare ad accedere alle istanze database utilizzando la risoluzione dei nomi DNS configurando un server DNS locale come server secondario. Tuttavia, non è possibile creare nuove istanze DB o modificare istanze DB esistenti. I backup automatici non si verificano quando non c'è connettività. Se si verifica un errore di istanza DB, l'istanza DB non viene sostituita

automaticamente fino al ripristino della connettività. Si consiglia di ripristinare la connettività di rete il prima possibile.

Argomenti

Prerequisiti per Amazon RDS su AWS Outposts

Di seguito sono indicati i prerequisiti per l'utilizzo di Amazon RDS su AWS Outposts:

- Installa AWS Outposts nel tuo data center locale. Per ulteriori informazioni su AWS Outposts, consulta AWS Outposts.
- Assicurarsi di avere almeno una sottorete disponibile per RDS in Outposts. È possibile utilizzare la stessa sottorete per altri carichi di lavoro.
- È necessario disporre di una connessione di rete affidabile tra Outpost e una regione AWS .

Supporto Amazon RDS su AWS Outposts per le funzionalità Amazon RDS

La seguente tabella descrive le funzionalità Amazon RDS supportate da Amazon RDS su AWS Outposts.

Funzionalità	Supportato	Note	Ulteriori informazioni
Provisioning istanza DB	Sì	È possibile creare solo istanze database per i motori di database RDS per SQL Server, RDS for MySQL e RDS per PostgreSQL. Sono supportate le seguenti versioni: • Microsoft SQL Server: • 15.00.4043.16.v1 o versioni successive 2019 • 14.00.3294.2.v1 o versioni successive 2017 • 13.00.5820.21.v1 o versioni successive 2016 • Versione MySQL 8.0.28 e versioni successive a MySQL 8.0 • Tutte le versioni di PostgreSQL 16 e 15 e 14 e 13 e PostgreSQL versione 12.5 e	Creazione delle istanze database per Amazon RDS su AWS Outposts

Funzionalità	Supportato	Note	Ulteriori informazioni
		successive di PostgreSQL 12	
È possibile utilizzare Microsoft SQL Server Management Studio per connettersi a un'istanza database di Microsoft SQL Server.	Sì	Alcune versioni TLS e cifrari di crittografia potrebbero non essere sicuri. Per disattivarli, seguire le istruzioni descritte in Configurazione dei protocolli di protezione e dei cifrari.	Connessione a un'istanza database che esegua il motore di database di Microsoft SQL Server
Modifica della password dell'utente master	Sì	—	Modifica di un'istanza database Amazon RDS
Ridenominazione di un'istanza database	Sì	—	Modifica di un'istanza database Amazon RDS
Riavvio di un'istanza database	Sì	—	Riavvio di un'istanza database
Arresto di un'istanza database	Sì	—	Arresto temporaneo di un'istanza database Amazon RDS
Avvio di un'istanza database	Sì	—	Avvio di un'istanza database Amazon RDS arrestata in precedenza

Funzionalità	Supportato	Note	Ulteriori informazioni
Implementazioni Multi-AZ	Sì	Le implementazioni Multi-AZ sono supportate su istanze database MySQL e PostgreSQL. Le implementazioni multi-AZ non supportano l'instradamento VPC diretto (DVR).implementazione multi-AZ	Creazione delle istanze database per Amazon RDS su AWS Outposts Configurazione e gestione di un'implementazione multi-AZ
Gruppi di parametri database	Sì	—	Utilizzo di gruppi di parametri
Repliche di lettura	Sì	Le repliche di lettura sono supportate per le istanze database MySQL e PostgreSQL. Le repliche di lettura non supportano l'instradamento VPC diretto (DVR).	Creazione di repliche di lettura per Amazon RDS su AWS Outposts
Crittografia a riposo	Sì	RDS in Outposts non supporta istanze DB non crittografate.	Crittografia delle risorse Amazon RDS
AWS Identity and Access Management Autenticazione del database (IAM)	No	—	Autenticazione del database IAM per MariaDB, MySQL e PostgreSQL

Funzionalità	Supportato	Note	Ulteriori informazioni
Associare un ruolo IAM a un'istanza DB	No	—	add-role-to-dbAWS CLIcomando -instance AddRoleToFunzionamento dell'API RDS DBInstance
Autenticazione Kerberos	No	—	Autenticazione Kerberos
Tagging delle risorse Amazon RDS	Sì	—	Tagging delle risorse Amazon RDS
Gruppi di opzioni	Sì	—	Uso di gruppi di opzioni
Modifica della finestra di manutenzione	Sì	—	Manutenzione di un'istanza database
Aggiornamenti automatici a versioni secondarie	Sì	—	Aggiornamento automatico della versione secondaria del motore
Modifica della finestra di backup	Sì	—	Introduzione ai backup Modifica di un'istanza database Amazon RDS
Modifica della classe di istanza database	Sì	—	Modifica di un'istanza database Amazon RDS
Modifica dello storage allocato	Sì	—	Modifica di un'istanza database Amazon RDS

Funzionalità	Supportato	Note	Ulteriori informazioni
Storage autoscaling (Auto Scaling dello storage)	Sì	—	Gestione della capacità automaticamente con Auto Scaling dello storage Amazon RDS
Istantanee di istanza DB manuali e automatiche	Sì	Puoi archiviare backup automatici e snapshot manuali nella tua Regione AWS o localmente nel tuo Outpost. I backup locali sono supportati su istanze database MySQL e PostgreSQL. Per archiviare i backup sul proprio Outpost, assicurar si di avere Amazon S3 configurato su Outposts. I backup locali non sono supportati nelle implement azioni multi-AZ delle istanze.	Creazione delle istanze database per Amazon RDS su AWS Outposts Amazon S3 su Outposts Creazione di uno snapshot DB per un'istanza DB Single-AZ
Ripristino da uno snapshot DB	Sì	Puoi archiviare backup automatici e snapshot manuali per l'istanza database ripristinata nella Regione AWS madre o localmente sul tuo Outpost.	Considerazioni per il ripristino delle istanze database di Amazon RDS in AWS Outposts Ripristino da uno snapshot database

Funzionalità	Supportato	Note	Ulteriori informazioni
Ripristino di un'istanza database da Amazon S3	No	—	Ripristino di un backup in un'istanza database MySQL
Esportazione dei dati snapshot in Amazon S3	No	—	Esportazione dei dati dello snapshot DB in Simple Storage Service (Amazon S3)
Ripristino del PC oint-in-time	Sì	Puoi archiviare backup automatici e snapshot manuali per l'istanza database ripristinata nella Regione AWS madre o localmente sul tuo Outpost, con una sola eccezione.	Considerazioni per il ripristino delle istanze database di Amazon RDS in AWS Outposts Ripristino a un'ora specifica per un'istanza database
Enhanced monitoring (Monitoraggio avanzato)	No	—	Monitoraggio dei parametri del sistema operativo con il monitoraggio avanzato
CloudWatch Monitoraggio Amazon	Sì	Puoi visualizzare lo stesso insieme di metriche disponibili per i tuoi database nella Regione AWS.	Monitoraggio dei parametri di Amazon RDS con Amazon CloudWatch
Pubblicazione dei log del motore di database su Logs CloudWatch	Sì	—	Pubblicazione di log di database su Amazon CloudWatch Logs

Funzionalità	Supportato	Note	Ulteriori informazioni
Notifiche di eventi	Sì	—	Utilizzo della notifica degli eventi di Amazon RDS
Performance Insights di Amazon RDS	No	—	Monitoraggio del carico DB con Performance Insights su Amazon RDS
Visualizzazione o download dei log del database	No	RDS in Outposts non supporta la visualizzazione dei log del database utilizzando la console o la descrizione dei log del database tramite l'API AWS CLI o RDS. RDS in Outposts non supporta il download dei log del database tramite la console o il download dei log del database tramite l'API AWS CLI o RDS.	Monitoraggio dei file di log di Amazon RDS
Proxy Amazon RDS	No	—	Utilizzo di Server proxy per Amazon RDS
Procedure archiviate per Amazon RDS for MySQL	Sì	—	Riferimento delle stored procedure RDS per MySQL
Replica con database esterni per RDS for MySQL	No	—	Configurazione della replica della posizione del file di log binario con un'istanza di origine esterna.

Funzionalità	Supportato	Note	Ulteriori informazioni
Backup e ripristino nativi per Amazon RDS for Microsoft SQL Server	Sì	—	Importazione ed esportazione di database SQL Server mediante backup e ripristino nativi

Classi di istanza database supportate per Amazon RDS su AWS Outposts

Amazon RDS su AWS Outposts supporta le classi di istanza database indicate di seguito:

- Classi di istanza database di scopo generico
  - db.m5.24xlarge
  - db.m5.12xlarge
  - db.m5.4xlarge
  - db.m5.2xlarge
  - db.m5.xlarge
  - db.m5.large
- Classi di istanza database con memoria ottimizzata
  - db.r5.24xlarge
  - db.r5.12xlarge
  - db.r5.4xlarge
  - db.r5.2xlarge
  - db.r5.xlarge
  - db.r5.large

A seconda di come hai configurato il tuo Outpost, è possibile che non siano disponibili tutte queste classi. Ad esempio, se non hai acquistato le classi db.r5 per il tuo Outpost, non puoi usarle con RDS in Outposts.

Per le istanze database RDS in Outposts è supportata solo l’archiviazione SSD per scopi generici. Per altre informazioni sulle classi di istanza database, consulta Classi di istanze database.

Amazon RDS gestisce la manutenzione e il ripristino per le istanze database e richiede capacità attiva nell'Outpost per farlo. Si consiglia di configurare N+1 istanze EC2 per ogni classe di istanza database negli ambienti di produzione. RDS su Outposts può utilizzare la capacità supplementare di queste istanze EC2 per le operazioni di manutenzione e riparazione. Ad esempio, se gli ambienti di produzione dispongono di 3 classi di istanza database db.m5.large e 5 classi db.r5.xlarge, è consigliabile disporre di almeno 4 istanze EC2 m5.large e 6 istanze EC2 r5.xlarge. Per ulteriori informazioni, consulta Resilienza in AWS Outposts nella Guida per l'utente di AWS Outposts.

Indirizzi IP di proprietà del cliente per Amazon RDS in AWS Outposts

Amazon RDS in AWS Outposts utilizza le informazioni che fornisci sulla tua rete on-premise per creare un pool di indirizzi. Questo pool è noto come pool di indirizzi IP di proprietà del cliente (pool CoIP). Gli indirizzi IP di proprietà del cliente (CoIPs) forniscono connettività locale o esterna alle risorse nelle sottoreti Outpost attraverso la rete locale. Per ulteriori informazioni sui CoIP, consulta Indirizzi IP di proprietà del cliente nella Guida per l'utente di AWS Outposts.

Ogni istanza databaseRDS in Outposts dispone di un indirizzo IP privato per il traffico all'interno del proprio cloud privato virtuale (VPC). Questo indirizzo IP privato non è accessibile pubblicamente. Puoi utilizzare l'opzione Public (Pubblico) per indicare se l'istanza database dispone anche di un indirizzo IP pubblico oltre all'indirizzo IP privato. L'uso dell'indirizzo IP pubblico per le connessioni le instrada attraverso Internet e in alcuni casi può causare latenze elevate.

Invece di utilizzare questi indirizzi IP privati e pubblici, RDS in Outposts supporta l'utilizzo dei CoIP per le istanze database tramite le sottoreti. Quando utilizzi un CoIP per un'istanza database RDS in Outposts, ti connetti all'istanza database con l'endpoint dell'istanza database. RDS in Outposts utilizza quindi automaticamente il CoIP per tutte le connessioni sia dall'interno che dall'esterno del VPC.

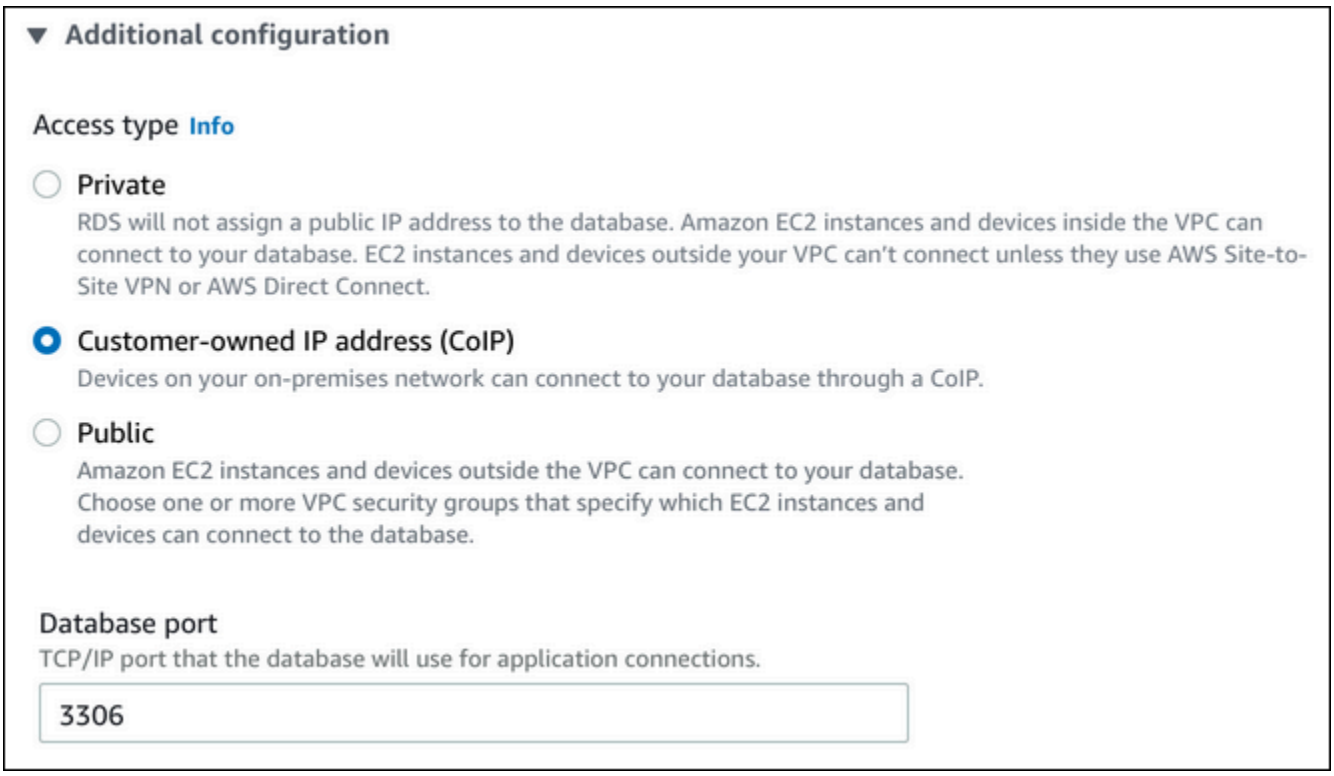
I CoIP possono fornire i seguenti vantaggi per le istanze DB RDS in Outposts:

- Minore latenza di connessione
- Sicurezza migliorata

Utilizzo dei CoIP

Puoi attivare o disattivare i CoIP per un'istanza database RDS in Outposts utilizzando la AWS Management Console, AWS CLI o l'API RDS:

- Nella AWS Management Console, scegli l'impostazione Customer-owned IP address (CoIP) (Indirizzo IP di proprietà del cliente (CoIP)) in Access type (Tipo di accesso) per abilitare un CoIP. Scegli una delle altre impostazioni per disattivarle.



- Con AWS CLI, utilizza l'opzione `--enable-customer-owned-ip | --no-enable-customer-owned-ip`.
- Con l'API RDS, utilizza il `EnableCustomerOwnedIp` parametro.

Puoi attivare o disattivare i CoIP con una delle seguenti operazioni:

- Creare un'Istanza database.

  Per ulteriori informazioni, consulta Creazione delle istanze database per Amazon RDS su AWS Outposts.
- Modificare un'istanza database

  Per ulteriori informazioni, consulta Modifica di un'istanza database Amazon RDS.
- Creare una replica di lettura

  Per ulteriori informazioni, consulta Creazione di repliche di lettura per Amazon RDS su AWS Outposts.
- Ripristinare un'istanza database da uno snapshot

  Per ulteriori informazioni, consulta Ripristino da uno snapshot database.

- Ripristinare un'istanza database a un'ora specifica

  Per ulteriori informazioni, consulta Ripristino a un'ora specifica per un'istanza database.

> **Note**
>
> In alcuni casi, puoi attivare i CoIP per un'istanza database, ma Amazon RDS non è in grado di allocare un CoIP per l'istanza database. In questi casi, lo stato dell'istanza database viene modificato in incompatible-network. Per ulteriori informazioni sullo stato delle istanze DB, consulta Visualizzazione dello stato dell'istanza database di Amazon RDS .

Restrizioni

Le seguenti limitazioni si applicano al supporto CoIP per le istanze DB RDS in Outposts:

- Quando utilizzi un CoIP per un'istanza database, assicurati che l'accessibilità pubblica sia disattivata per l'istanza database.
- Assicurati che le regole in entrata per i tuoi gruppi di sicurezza VPC includano l'intervallo di indirizzi CoIP (blocco CIDR). Per ulteriori informazioni sulla configurazione di un gruppo di sicurezza, consulta Fornisci accesso alla istanza database nel VPC creando un gruppo di sicurezza.
- Non è possibile assegnare un CoIP da un pool CoIP a un'istanza database. Quando utilizzi un CoIP per un'istanza database, Amazon RDS assegna automaticamente all'istanza database un CoIP da un pool di CoIP.
- È necessario utilizzare l'Account AWS che possiede le risorse Outpost (proprietario) o condividere le risorse riportate di seguito con altri Account AWS (consumer) nella stessa organizzazione.
  - L'Outpost
  - La tabella di route del gateway locale (LGW) per il VPC dell'istanza database
  - Il pool o i pool CoIP per la tabella di route LGW

  Per ulteriori informazioni, consulta Utilizzo delle risorse AWS Outposts condivise nella Guida per l'utente di AWS Outposts.

Operare con le implementazioni Multi-AZ per Amazon RDS su AWS Outposts

Per eseguire le implementazioni multi-AZ, Amazon RDS crea un'istanza database primaria in un AWS outpost. RDS replica in modo sincrono i dati in un'istanza database in standby in un Outpost diverso.

Le implementazioni Multi-AZ su AWS Outposts e nelle Regioni AWS seguono la stessa modalità, ma presentano le seguenti differenze:

- Richiedono una connessione locale tra due o più outpost.
- Richiedono pool IP (CoIP) di proprietà del cliente. Per ulteriori informazioni, consulta Indirizzi IP di proprietà del cliente per Amazon RDS in AWS Outposts.
- La replica viene eseguita sulla rete locale.

Multi-AZ è disponibile su AWS Outposts per tutte le versioni supportate di MySQL e PostgreSQL in RDS su Outposts. I backup locali non sono supportati per le implementazioni Multi-AZ. Per ulteriori informazioni, consulta Creazione delle istanze database per Amazon RDS su AWS Outposts.

Operare con il modello di responsabilità condivisa

Sebbene AWS si impegni in modo commercialmente ragionevole a fornire istanze database configurate per l'alta disponibilità, tale disponibilità si basa su un modello di responsabilità condivisa. Affinché RDS su Outposts possa eseguire il failover e riparare le istanze database, è necessario che ciascuno degli outpost sia collegato alla propria Regione AWS.

Inoltre, per poter eseguire la replica sincrona, RDS su Outposts richiede che ci sia connettività tra l'outpost che ospita l'istanza database primaria e quello che ospita l'istanza database in standby. Qualsiasi cosa ostacoli questa connessione può impedire l'esecuzione del failover da parte di RDS su Outposts.

Le implementazioni standard dell'istanza database risultanti dalla replica sincrona dei dati presentano spesso una latenza elevata. La larghezza di banda e la latenza della connessione tra l'outpost che ospita l'istanza database primaria e quello che ospita l'istanza database in standby influiscono direttamente sulle latenze. Per ulteriori informazioni, consulta Prerequisiti.

Miglioramento della disponibilità

Per migliorare la disponibilità, ti consigliamo di eseguire queste azioni:

- Assegna una capacità aggiuntiva sufficiente per le applicazioni mission-critical, al fine di consentire il ripristino e il failover in caso di problemi con l'host sottostante. Ciò vale per tutti gli outpost aventi sottoreti nel tuo gruppo di sottoreti database. Per ulteriori informazioni, consulta Resilienza in AWS Outposts.
- Fornisci una connettività di rete ridondante agli outpost in uso.
- Utilizza più di due outpost. Se Amazon RDS ha a disposizione più di due Outpost, può recuperare un'istanza database. RDS esegue questo ripristino spostando l'istanza database in un altro Outpost se si verifica un errore nell'Outpost corrente.
- Fornisci all'outpost una doppia sorgente di alimentazione e una connettività di rete ridondante.

Per le reti locali, ti consigliamo quanto segue:

- La latenza dovuta al tempo di round trip (RTT) tra l'outpost che ospita l'istanza database primaria e quello che ospita l'istanza database in standby, influisce direttamente sulla latenza di scrittura. Mantieni la latenza RTT tra gli outpost AWS al di sotto dei 10 millesecondi. Idealmente non più di 5 millisecondi, ma i requisiti possono variare.

  Per verificare l'impatto netto sulla latenza di rete, cerca `WriteLatency` tra i parametri di Amazon CloudWatch. Per ulteriori informazioni, consulta CloudWatch Parametri Amazon per Amazon RDS.
- La presenza di connessione tra gli outpost influisce sulla disponibilità complessiva delle istanze database. Connettività di rete ridondante tra gli outpost.

Prerequisiti

I prerequisiti per le implementazioni multi-AZ in RDS su Outposts sono i seguenti:

- Due outpost connessi tra loro tramite connessione locale e collegati a diverse zone di disponibilità in una Regione AWS.
- Verifica che i gruppi di sottorete database contengano i seguenti elementi:
  - Minimo due sottoreti in almeno due zone di disponibilità in una determinata Regione AWS.
  - Sottoreti localizzate esclusivamente in outpost.
  - Minimo due sottoreti in almeno due outpost all'interno dello stesso cloud privato virtuale (VPC).

- Associa il VPC dell'istanza database a tutte le tabelle di routing del gateway locale. Questa associazione è necessaria perché la replica viene eseguita sulla rete locale utilizzando i gateway locali Outpost.

  Ad esempio, supponiamo che il tuo VPC contenga la sottorete-A nell'outpost-A e la sottorete-B nell'outpost-B. L'outpost-A utilizza il GatewayLocale-A (LGW-A) e l'outpost-B utilizza il GatewayLocale-B (LGW-B). All'LGW-A è assegnata la TabellaDiRouting-A e all'LGW-B la TabellaDiRouting-B. Vuoi utilizzare sia la TabellaDiRouting-A sia la TabellaDiRouting-B per il traffico di replica. Per fare ciò, dovrai associare il tuo VPC sia alla TabellaDiRouting-A sia alla TabellaDiRouting-B.

  Per ulteriori informazioni su come creare un'associazione, consulta il comando della AWS CLI Amazon EC2 create-local-gateway-route-table-vpc-association.
- Assicurati che gli outpost utilizzino l'IP routing (CoIP) di proprietà del cliente. Ogni tabella di routing deve inoltre avere almeno un pool di indirizzi. Amazon RDS assegna un indirizzo IP aggiuntivo alle istanze database primarie e uno alle istanze database in standby per la sincronizzazione dei dati.
- Assicurati che l'Account AWS proprietario delle istanze database RDS sia anche il proprietario delle tabelle di routing del gateway locale e dei pool CoIP. In alternativa assicurati che faccia parte di una condivisione Resource Access Manager con accesso alle tabelle di routing del gateway locale e ai pool CoIP.
- Assicurati che gli indirizzi IP dei pool CoIP possano essere instradati da un gateway locale Outpost all'altro.
- Assicurati che i blocchi CIDR del VPC (ad esempio, 10.0.0.0/4) e i blocchi CIDR del pool CoIP non contengano indirizzi IP di classe E (240.0.0.0/4). RDS utilizza questi indirizzi IP internamente.
- Assicurati di aver impostato correttamente il traffico correlato in entrata e in uscita.

  RDS su Outposts stabilisce una connessione di rete privata virtuale (VPN) tra le istanze database primarie e quelle in standby. Affinché la connessione funzioni correttamente, la rete locale deve consentire il traffico correlato in entrata e in uscita per Internet Security Association and Key Management Protocol (ISAKMP). Per fare ciò, utilizza la porta UDP (User Datagram Protocol) 500 e l'attraversamento NAT-T (Network Address Translation Traversal) con protezione IP Security (IPSec), che utilizza la porta UDP 4500.

Per ulteriori informazioni sui CoIP, consulta Indirizzi IP di proprietà del cliente per Amazon RDS in AWS Outposts in questa guida e Indirizzi IP di proprietà del cliente nella Guida per l'utente di AWS Outposts.

Utilizzo delle operazioni API per ottenere autorizzazioni Amazon EC2

RDS richiede l'accesso alle risorse del pool CoIP indipendentemente dal fatto che utilizzi o meno dei CoIP per la tua istanza database su AWS Outposts. Durante l'esecuzione di un'implementazione Multi-AZ, RDS può chiamare le seguenti operazioni API per richiedere per tuo conto le autorizzazioni EC2 relative ai COIP:

- `CreateCoipPoolPermission` - quando crei un'istanza database Multi-AZ in RDS su Outposts
- `DeleteCoipPoolPermission` - quando elimini un'istanza database Multi-AZ in RDS su Outposts

Queste operazioni API concedono o rimuovono dagli account RDS interni l'autorizzazione ad assegnare indirizzi IP elastici provenienti dal pool CoIP specificato dall'autorizzazione. È possibile visualizzare questi indirizzi IP utilizzando l'operazione API `DescribeCoipPoolUsage`. Per ulteriori informazioni sui CoIP, consulta Indirizzi IP di proprietà del cliente per Amazon RDS in AWS Outposts e Indirizzi IP di proprietà del cliente nella Guida per l'utente di AWS Outposts

Durante le implementazioni Multi-AZ, RDS può anche chiamare le seguenti operazioni API per richiedere per tuo conto le autorizzazioni EC2 relative alle tabelle di routing gateway locali:

- `CreateLocalGatewayRouteTablePermission` - quando crei un'istanza database Multi-AZ in RDS su Outposts
- `DeleteLocalGatewayRouteTablePermission` - quando elimini un'istanza database Multi-AZ in RDS su Outposts

Queste operazioni API concedono o rimuovono dagli account RDS interni l'autorizzazione ad associare i VPC RDS interni con le tue tabelle di routing gateway locali. È possibile visualizzare queste associazioni tra tabella di routing e VPC utilizzando le operazioni API `DescribeLocalGatewayRouteTableVpcAssociations`.

Creazione delle istanze database per Amazon RDS su AWS Outposts

La creazione di un'istanza database Amazon RDS su AWS Outposts è simile alla creazione di un'istanza database Amazon RDS in AWS Cloud. Tuttavia, è necessario specificare un gruppo di sottoreti database associato a Outpost.

Un Virtual Private Cloud (VPC) basato sul servizio Amazon VPC può estendersi su tutte le zone di disponibilità in una Regione AWS. Puoi estendere qualsiasi VPC nella Regione AWS al tuo Outpost aggiungendo una sottorete Outpost. Per aggiungere una sottorete Outpost a un VPC, specificare l'Amazon Resource Name (ARN) dell'outpost quando si crea la sottorete.

Prima di creare un'istanza database RDS in Outposts, è possibile creare un gruppo di sottoreti database che include una sottorete associata a Outpost. Quando si crea un'istanza database RDS in Outposts, specificare questo gruppo di sottoreti database. È inoltre possibile scegliere di creare un nuovo gruppo di sottoreti DB quando si crea l'istanza DB.

Per informazioni sulla configurazione AWS Outposts, vedere la Guida dell'utente AWS Outposts.

Console

Per creare un gruppo di sottoreti del database

Creare un gruppo di sottoreti database con una sottorete associata a Outpost.

È inoltre possibile scegliere di creare un nuovo gruppo di sottoreti DB quando si crea l'istanza DB. In tal caso, saltare questa procedura.

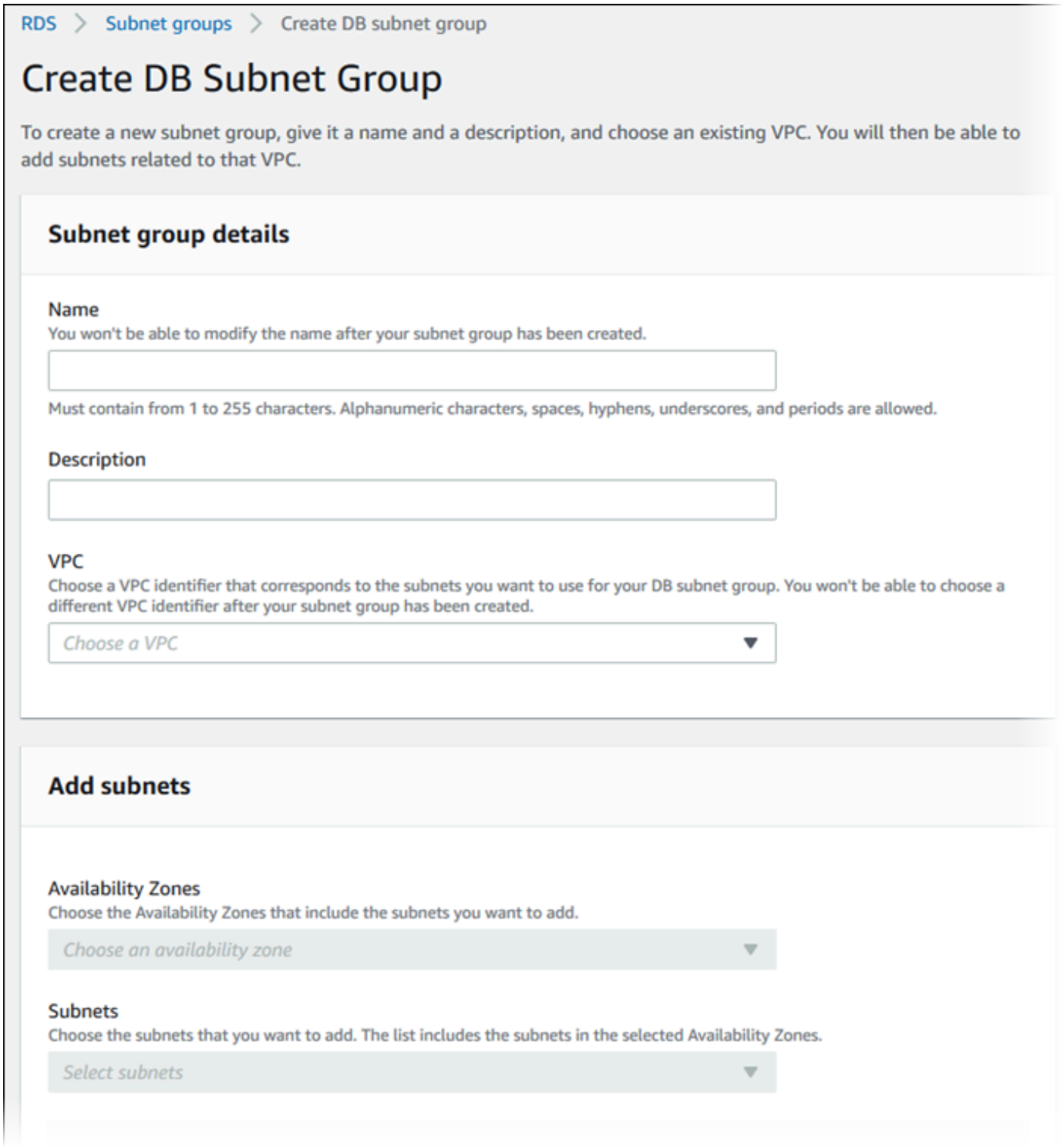
> **Note**
>
> Per creare un gruppo di sottoreti database per Cloud AWS, è necessario specificare almeno due sottoreti.

Per creare un gruppo di sottoreti database per il proprio Outpost

1. Accedi alla AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nell'angolo in alto a destra della console Amazon RDS, scegli la Regione AWS in cui vuoi creare il gruppo di sottoreti di database.

3. Scegliere Subnet groups (Gruppi di sottoreti), quindi fare clic su Create DB Subnet Group (Crea gruppo di sottoreti database).

   Viene visualizzata la pagina Create DB Subnet Group (Crea gruppo di sottoreti database).



4. Per Name (Nome), digitare il nome del gruppo di sottoreti database.

5. Per Description (Descrizione), scegliere una descrizione del gruppo di sottoreti database.
6. Per VPC, scegliere il VPC per il quale si sta creando il gruppo di sottoreti database.
7. In Zona di disponibilità, selezionare la zona di disponibilità per l'Outpost.
8. Per Sottorete, scegliere la sottorete da utilizzare con RDS in Outposts.
9. Scegliere Create (Crea) per creare il gruppo di sottoreti database.

Creazione di RDS sull'istanza database Outposts

Creare l'istanza DB e scegliere l'outpost per l'istanza DB.

Per creare un'istanza database RDS in Outposts tramite la console

1. Accedi alla AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nell'angolo in alto a destra della console Amazon RDS, scegli la Regione AWS a cui è collegato l'Outpost in cui desideri creare l'istanza database.
3. Nel riquadro di navigazione, scegliere Databases (Database).
4. Scegliere Create database (Crea database).

   AWS Management Console rileva gli outpost disponibili configurati e presenta l'opzione Locale nella sezione Percorso database .

   > **Note**
   >
   > Se non sono stati configurati gli outpost, la sezione Posizione database non viene visualizzata oppure l'opzione RDS in Outposts non è disponibile nella sezione Scegliere un metodo di creazione locale.

5. Per Ubicazione del database, scegliere On-premise.
6. Per Metodo di creazione locale, scegliere RDS in Outposts.
7. Specificare le impostazioni per Connettività degli Outposts. Queste impostazioni sono per l'Outpost che utilizza il VPC con il gruppo di sottoreti database per l'istanza database. Il VPC deve essere basato sul servizio Amazon VPC.

   a. Per Cloud privato virtuale (VPC), selezionare il VPC che contiene il gruppo di sottoreti database per l'istanza database.

b. Per Gruppo di sicurezza VPC, selezionare il gruppo di sicurezza Amazon VPC per l'istanza database.

c. In DB subnet group (Gruppo di sottoreti DB), seleziona il gruppo di sottoreti DB per l'istanza database.

   Puoi scegliere un gruppo di sottoreti di database esistente associato a Outpost, ad esempio se hai eseguito la procedura in Per creare un gruppo di sottoreti del database.

   È inoltre possibile creare un nuovo gruppo di sottoreti database per l'Outpost.

8. Per implementazione Multi-AZ, scegli Create a standby instance (recommended for production usage) (Crea un'istanza in standby (opzione consigliata per l'utilizzo di produzione)) per creare un'istanza database in standby in un altro Outpost.

> **Note**
>
> Questa opzione non è disponibile per Microsoft SQL Server.
> Se scegli di creare un'implementazione Multi-AZ, non puoi archiviare i backup nel tuo Outpost.

9. In Backup, eseguire le seguenti operazioni:

   a. Per Backup target, scegliere in uno dei seguenti modi:

      - Cloud AWS per archiviare backup automatici e istantanee manuali nella Regione AWS madre.
      - Outposts (locale) per creare backup locali.

> **Note**
>
> Per archiviare i backup sul proprio Outpost, l'Outpost deve disporre della funzionalità Amazon S3. Per ulteriori informazioni, consultare Amazon S3 su Outposts.
> I backup locali non sono supportati per le implementazioni multi-AZ o le repliche di lettura.

   b. Scegli Abilita backup automatici per creare point-in-time istantanee della tua istanza DB.

Se attivi i backup automatici, puoi scegliere i valori per Backup retention period (Periodo di conservazione dei backup) e Backup window (Finestra di backup) oppure lasciare i valori di default.

10. Specificare le altre impostazioni dell'istanza database nel modo necessario.

    Per informazioni su ciascuna impostazione durante la creazione di un'istanza database, consulta Impostazioni per istanze database.

11. Scegliere Create database (Crea database).

    Verrà visualizzato il riquadro Databases (Database). Un banner indica che l'istanza database è in fase di creazione e visualizza il pulsante Visualizza i dettagli delle credenziali.

Visualizzazione di dettagli istanza database

Dopo aver creato l'istanza database è possibile visualizzarne le credenziali e altri dettagli.

Per visualizzare i dettagli dell'istanza:

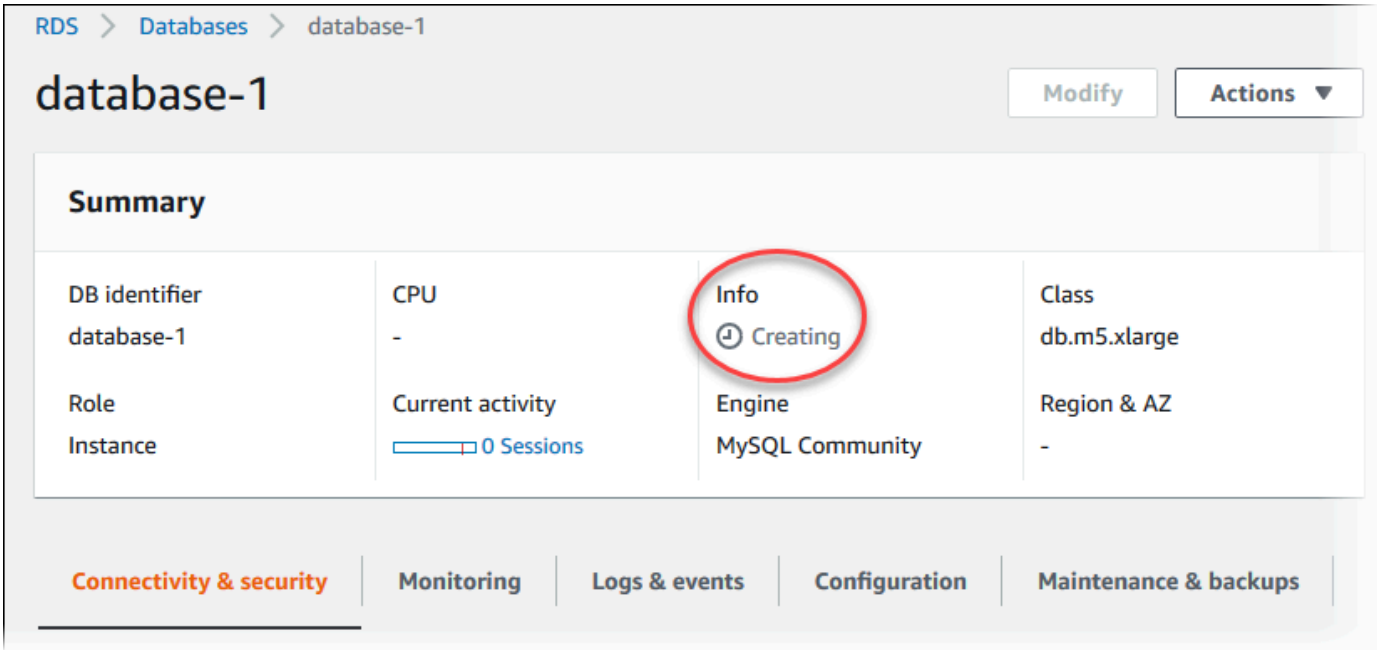
1. Per vedere nome utente e password principali per l'istanza database, seleziona View credential details (Visualizza i dettagli delle credenziali) sulla pagina Databases.

   È possibile connettersi all'istanza database come utente master utilizzando queste credenziali.

   > **Important**
   >
   > Non potrai visualizzare di nuovo la password dell'utente principale. Se non la registri, potresti doverla modificare. Per modificare la password dell'utente principale dopo che l'istanza database è disponibile, modificare l'istanza database. Per ulteriori informazioni sulla modifica di un'istanza database , consulta Modifica di un'istanza database Amazon RDS.

2. Selezionare il nome della nuova istanza database sulla pagina Database.

   Nella console RDS vengono visualizzati i dettagli per la nuova istanza database. L'istanza database rimane nello stato Creating (In creazione) fino a quando non viene creata ed è pronta per l'uso. Quando lo stato cambia in Available (Disponibile) è possibile connettersi all'istanza database. A seconda della classe di istanza database e dello storage allocato, potrebbero trascorrere diversi minuti prima che la nuova istanza database sia disponibile.



Dopo che l'istanza database è diventata disponibile, è possibile gestirla in modo analogo a come si gestiscono le istanze database RDS in Cloud AWS.

AWS CLI

Prima di creare una nuova istanza database in un Outpost con AWS CLI, creare innanzitutto un gruppo di sottoreti database da utilizzare con RDS su Outposts.

Per creare un gruppo di sottoreti database per il proprio Outpost

- Utilizza il comando create-db-subnet-group. Per `--subnet-ids`, specificare il gruppo di sottoreti nell'Outpost per l'utilizzo da parte di RDS in Outposts.

  PerLinux, omacOS: Unix

```
aws rds create-db-subnet-group \
    --db-subnet-group-name myoutpostdbsubnetgr \
    --db-subnet-group-description "DB subnet group for RDS on Outposts" \
    --subnet-ids subnet-abc123
```

  Per Windows:

```
aws rds create-db-subnet-group ^
    --db-subnet-group-name myoutpostdbsubnetgr ^
```

```
    --db-subnet-group-description "DB subnet group for RDS on Outposts" ^
    --subnet-ids subnet-abc123
```

Per creare un'istanza database RDS in Outposts tramite AWS CLI

- Utilizza il comando create-db-instance. Specificare una zona di disponibilità per l'Outpost, un gruppo di sicurezza Amazon VPC associato all'outpost e il gruppo di sottoreti DB creato per l'outpost. È possibile includere le seguenti opzioni:
  - `--db-instance-identifier`
  - `--db-instance-class`
  - `--engine` – Il motore di database. Utilizza uno dei seguenti valori:
    - MySQL - Specificare `mysql`.
    - PostgreSQL - Specificare`postgres`.
    - Microsoft SQL Server - Specifica `sqlserver-ee`, `sqlserver-se` o `sqlserver-web`.
  - `--availability-zone`
  - `--vpc-security-group-ids`
  - `--db-subnet-group-name`
  - `--allocated-storage`
  - `--max-allocated-storage`
  - `--master-username`
  - `--master-user-password`
  - `--multi-az | --no-multi-az`— (Facoltativo) Indica se creare un'istanza database in standby in una zona di disponibilità diversa. Il valore predefinito è `--no-multi-az`.

    L'opzione `--multi-az` non è disponibile per SQL Server.
  - `--backup-retention-period`
  - `--backup-target`— (Facoltativo) Dove archiviare backup automatici e istantanee manuali. Utilizza uno dei seguenti valori:
    - `outposts`— Conservarli localmente sul proprio Outpost.
    - `region`— Conservarli nella Regione AWS madre. Si tratta del valore di default.

Se utilizzi l'opzione `--multi-az`, non è possibile utilizzare `outposts` per `--backup-target`. Inoltre, l'istanza database non può avere repliche di lettura se si utilizza `outposts` per `--backup-target`.

- `--storage-encrypted`
- `--kms-key-id`

Example

Nell'esempio seguente viene creata un'istanza database MySQL denominata `myoutpostdbinstance` con backup memorizzati sul proprio Outpost.

Per LinuxmacOS, oUnix:

```
aws rds create-db-instance \
    --db-instance-identifier myoutpostdbinstance \
    --engine-version 8.0.17 \
    --db-instance-class db.m5.large \
    --engine mysql \
    --availability-zone us-east-1d \
    --vpc-security-group-ids outpost-sg \
    --db-subnet-group-name myoutpostdbsubnetgr \
    --allocated-storage 100 \
    --max-allocated-storage 1000 \
    --master-username masterawsuser \
    --manage-master-user-password \
    --backup-retention-period 3 \
    --backup-target outposts \
    --storage-encrypted \
    --kms-key-id mykey
```

Per Windows:

```
aws rds create-db-instance ^
    --db-instance-identifier myoutpostdbinstance ^
    --engine-version 8.0.17 ^
    --db-instance-class db.m5.large ^
    --engine mysql ^
    --availability-zone us-east-1d ^
    --vpc-security-group-ids outpost-sg ^
    --db-subnet-group-name myoutpostdbsubnetgr ^
```

```
    --allocated-storage 100 ^
    --max-allocated-storage 1000 ^
    --master-username masterawsuser ^
    --manage-master-user-password ^
    --backup-retention-period 3 ^
    --backup-target outposts ^
    --storage-encrypted ^
    --kms-key-id mykey
```

Per informazioni su ciascuna impostazione durante la creazione di un'istanza database, consulta Impostazioni per istanze database.

API RDS

Per creare una nuova istanza DB in un Outpost con l'API RDS, crea prima un gruppo di sottoreti DB da utilizzare da RDS su Outposts chiamando l'operazione CreateDB. SubnetGroup Per `SubnetIds`, specificare il gruppo di sottoreti nell'Outpost per l'utilizzo da parte di RDS in Outposts.

Chiama quindi l'operazione CreateDBInstance con i parametri riportati di seguito. Specificare una zona di disponibilità per l'Outpost, un gruppo di sicurezza Amazon VPC associato all'outpost e il gruppo di sottoreti DB creato per l'outpost.

- `AllocatedStorage`
- `AvailabilityZone`
- `BackupRetentionPeriod`
- `BackupTarget`

  Se stai creando un'implementazione di istanza database multi-AZ, non puoi utilizzare `outposts` per `BackupTarget`. Inoltre, l'istanza database non può avere repliche di lettura se si utilizza `outposts` per `BackupTarget`.
- `DBInstanceClass`
- `DBInstanceIdentifier`
- `VpcSecurityGroupIds`
- `DBSubnetGroupName`
- `Engine`
- `EngineVersion`
- `MasterUsername`

- `MasterUserPassword`
- `MaxAllocatedStorage` (facoltativo)
- `MultiAZ` (facoltativo)
- `StorageEncrypted`
- `KmsKeyID`

Per informazioni su ciascuna impostazione durante la creazione di un'istanza database, consulta Impostazioni per istanze database.

Creazione di repliche di lettura per Amazon RDS su AWS Outposts

Amazon RDS on AWS Outposts utilizza la funzionalità di replica integrata dei motori DB MySQL e PostgreSQL per creare una replica di lettura da un'istanza DB di origine. L'istanza DB di origine diventa l'istanza DB primaria. Gli aggiornamenti applicati all'istanza database primaria vengono copiati in modo asincrono nella replica di lettura. Puoi ridurre il carico sull'istanza database di database primaria instradando le query di lettura dalle applicazioni alla replica di lettura. Tramite le repliche di lettura puoi dimensionare in modo elastico la capacità oltre i vincoli di una singola istanza database per carichi di lavoro di database particolarmente gravosi in lettura.

Quando crei una replica di lettura da un'istanza database RDS su Outposts, la replica di lettura utilizza un indirizzo IP di proprietà del cliente (CoIP). Per ulteriori informazioni, consulta Indirizzi IP di proprietà del cliente per Amazon RDS in AWS Outposts.

Le repliche di lettura di RDS su Outposts includono le seguenti limitazioni:

- Non puoi creare repliche di lettura di istanze database Amazon RDS per SQL Server in RDS su Outposts.
- Le repliche di lettura tra regioni non sono supportate in RDS su Outposts.
- Le repliche di lettura a cascata non sono supportate in RDS su Outposts.
- L'istanza database di origine RDS su Outposts non può avere backup locali. La destinazione di backup per l'istanza database di origine deve essere la tua Regione AWS.
- Le repliche di lettura richiedono pool IP (CoIP) di proprietà del cliente. Per ulteriori informazioni, consulta Indirizzi IP di proprietà del cliente per Amazon RDS in AWS Outposts.
- Le repliche di lettura su RDS on Outposts possono essere create solo nello stesso cloud privato virtuale (VPC) dell'istanza DB di origine.
- Le repliche di lettura su RDS on Outposts possono trovarsi sullo stesso Outpost o su un altro Outpost nello stesso VPC dell'istanza DB di origine.

È possibile creare una replica di lettura da un'istanza DB RDS on Outposts utilizzando l'API AWS Management Console AWS CLI, o RDS. Per ulteriori informazioni sulle repliche di lettura, consultare Uso delle repliche di lettura dell'istanza database.

Console

Per creare una replica di lettura da un'istanza database di origine

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, scegliere Databases (Database).
3. Scegli l'istanza database da usare come origine per la replica di lettura.
4. Per Actions (Operazioni), scegliere Create read replica (Crea replica di lettura).
5. Per DB instance identifier (Identificatore istanze DB) inserire un nome per la replica di lettura.
6. Specificare le impostazioni per Connettività degli Outposts. Queste impostazioni sono per l'Outpost che utilizza il cloud privato virtuale (VPC) con il gruppo di sottoreti database per l'istanza database. Il VPC deve essere basato sul servizio Amazon VPC.
7. Scegli la classe di istanza database. Consigliamo di usare la stessa classe di istanza database o più grande e lo stesso tipo di archiviazione dell'istanza database di origine per la replica di lettura.
8. Per Multi-AZ deployment (Implementazione multi-AZ), scegli Create a standby instance (recommended for production usage) (Crea un'istanza in standby (opzione consigliata per l'utilizzo di produzione)) per creare un'istanza database in standby in un'altra zona di disponibilità.

   La creazione della replica di lettura come un'istanza database Multi-AZ non dipende dal fatto che il database di origine sia un'istanza database Multi-AZ.
9. (Facoltativo) In Connectivity (Connettività), imposta i valori per il gruppo di sottorete e la zona di disponibilità.

   Se si specificano valori sia per il gruppo di sottorete che per la zona di disponibilità, la replica di lettura viene creata su un Outpost associato alla zona di disponibilità nel gruppo di sottoreti database.

   Se si specifica un valore per il gruppo di sottoreti e nessuna preferenza per la zona di disponibilità, la replica di lettura viene creata su un Outpost casuale nel gruppo di sottoreti database.
10. Per AWS KMS key, scegli l' AWS KMS key identificatore della chiave KMS.

    La replica di lettura deve essere crittografata.

11. Scegli altre opzioni in base alle esigenze.
12. Scegliere Create read replica (Crea replica di lettura).

Dopo aver creato la replica di lettura, è possibile visualizzarla nella pagina Databases (Database) della console RDS. Mostra Replica nella colonna Role (Ruolo).

AWS CLI

Per creare una replica di lettura da un'istanza database MySQL o PostgreSQL di origine, usa il comando -replica. AWS CLI create-db-instance-read

Puoi controllare dove viene creata la replica di lettura specificando le opzioni`--db-subnet-group-name` e `--availability-zone`:

- Se si specificano entrambe le opzioni `--db-subnet-group-name` e `--availability-zone`, la replica di lettura viene creata su un Outpost associato alla zona di disponibilità nel gruppo di sottoreti database.
- Se si specifica l'opzione `--db-subnet-group-name` e non l'opzione `--availability-zone`, la replica di lettura viene creata su un Outpost casuale nel gruppo di sottoreti database.
- Se non si specifica nessuna delle due opzioni, la replica di lettura viene creata sullo stesso Outpost dell'istanza database di origine RDS su Outposts.

Nell'esempio seguente viene creata una replica e specificata la posizione della replica di lettura includendo le opzioni `--db-subnet-group-name` e `--availability-zone`.

Example

UnixPer, o: Linux macOS

```
aws rds create-db-instance-read-replica \
    --db-instance-identifier myreadreplica \
    --source-db-instance-identifier mydbinstance \
    --db-subnet-group-name myoutpostdbsubnetgr \
    --availability-zone us-west-2a
```

Per Windows:

```
aws rds create-db-instance-read-replica ^
    --db-instance-identifier myreadreplica ^
```

```
    --source-db-instance-identifier mydbinstance ^
    --db-subnet-group-name myoutpostdbsubnetgr ^
    --availability-zone us-west-2a
```

API RDS

Per creare una replica di lettura da un'istanza database MySQL o PostgreSQL di origine, chiama l'operazione Amazon RDS API CreateDB con i seguenti parametri obbligatori: InstanceReadReplica

- `DBInstanceIdentifier`
- `SourceDBInstanceIdentifier`

Puoi controllare dove viene creata la replica di lettura specificando i parametri `DBSubnetGroupName` e `AvailabilityZone`:

- Se si specificano entrambi i parametri `DBSubnetGroupName` e `AvailabilityZone`, la replica di lettura viene creata su un Outpost associato alla zona di disponibilità nel gruppo di sottoreti database.
- Se si specifica il parametro `DBSubnetGroupName` e non il parametro `AvailabilityZone`, la replica di lettura viene creata su un Outpost casuale nel gruppo di sottoreti database.
- Se non si specifica nessuno dei due parametri, la replica di lettura viene creata sullo stesso Outpost dell'istanza database di origine RDS su Outposts.

Considerazioni per il ripristino delle istanze database di Amazon RDS in AWS Outposts

Nel caso di ripristino di un'istanza database di Amazon RDS in AWS Outposts, in genere è possibile scegliere la posizione di archiviazione per i backup automatici e gli snapshot manuali dell'istanza database ripristinata.

- Per il ripristino da uno snapshot DB manuale, è possibile memorizzare i backup sia nella Regione AWS madre o localmente sul tuo Outpost.
- Nel caso di ripristino da un backup automatico (ripristino point-in-time), si hanno meno possibilità di scelta:
  - Se si ripristina dalla Regione AWS madre, è possibile archiviare i backup nella Regione AWS o sul tuo Outpost.

- Se si esegue il ripristino dal proprio Outpost, è possibile memorizzare i backup solo sul proprio Outpost.

Utilizzo di Server proxy per Amazon RDS

Con Amazon RDS Proxy, puoi consentire alle tue applicazioni di eseguire il pool e condividere connessioni di database per migliorare la loro capacità di dimensionamento. RDS Proxy rende le applicazioni più resilienti agli errori del database connettendosi automaticamente a un'istanza database di standby, mantenendo al contempo le connessioni delle applicazioni. Utilizzando RDS Proxy, puoi anche applicare l'autenticazione AWS Identity and Access Management (IAM) per i database e archiviare in modo sicuro le credenziali in. AWS Secrets Manager

Con Server proxy per RDS, puoi gestire picchi imprevedibili nel traffico del database In caso contrario, questi picchi potrebbero causare problemi a causa di un numero eccessivo di connessioni o della creazione di nuove connessioni a una velocità elevata. Server proxy per RDS stabilisce un pool di connessioni al database e riutilizza le connessioni di questo pool. Questo approccio evita il sovraccarico della memoria e della CPU per aprire ogni volta una nuova connessione al database. Per proteggere un database dall'eccesso di sottoscrizioni, è possibile controllare il numero di connessioni al database che vengono create.

Il proxy RDS mette in coda o limita le connessioni alle applicazioni che non possono essere servite immediatamente dal pool di connessioni. Sebbene le latenze possano aumentare, l'applicazione può continuare a dimensionare senza errori improvvisi o senza sovraccaricare il database. Se le richieste di connessione superano i limiti specificati, RDS Proxy rifiuta le connessioni dell'applicazione (genera il carico). Allo stesso tempo, mantiene prestazioni prevedibili per il carico che RDS può gestire con la capacità disponibile.

Puoi ridurre il sovraccarico per elaborare le credenziali e stabilire una connessione sicura per ogni nuova connessione. RDS Proxy può gestire alcune di queste operazioni per conto del database.

RDS Proxy è compatibile con le versioni di motore supportate. È possibile abilitare RDS Proxy per la maggior parte delle applicazioni senza modifiche al codice.

Argomenti

- Gestire un RDS Proxy
- Utilizzo degli endpoint Amazon RDS Proxy
- Monitoraggio dei parametri del proxy RDS con Amazon CloudWatch
- Utilizzo degli eventi RDS Proxy
- Esempi della riga di comando per RDS Proxy
- Risoluzione dei problemi per RDS Proxy
- Utilizzo di RDS Proxy con AWS CloudFormation

Disponibilità di regioni e versioni

Il supporto varia a seconda delle versioni specifiche di ciascun motore di database e a seconda delle Regioni AWS. Per ulteriori informazioni sulla disponibilità di versioni e regioni per Amazon RDS con il proxy RDS, consulta Regioni e motori DB supportati per Amazon RDS Proxy.

Quote e limiti per RDS Proxy

A RDS Proxy si applicano i seguenti limiti:

- Puoi avere fino a 20 proxy per ogni AWS ID di account. Se la tua applicazione richiede più proxy, puoi richiederne altri aprendo un ticket presso l'organizzazione Support. AWS
- Ogni proxy può avere fino a 200 segreti Secrets Manager associati. Pertanto, ogni proxy può connettersi a un massimo di 200 account utente diversi in qualsiasi momento.
- Ogni proxy ha un endpoint predefinito. Puoi anche aggiungere fino a 20 endpoint proxy per ogni proxy. È possibile creare, visualizzare, modificare ed eliminare questi endpoint.
- Per le istanze database RDS nelle configurazioni di replica, puoi associare un proxy solo all'istanza database del writer, non a una replica di lettura.
- RDS Proxy deve essere nello stesso virtual private cloud (VPC) del database. Sebbene il database non sia accessibile pubblicamente, il proxy può esserlo. Ad esempio, se state prototipando il database su un host locale, non potete connettervi al proxy a meno che non impostiate i requisiti di rete necessari per consentire la connessione al proxy. Questo perché l'host locale si trova all'esterno del VPC del proxy.
- Non è possibile utilizzare RDS Proxy con un VPC con tenancy impostato a `dedicated`.
- Se utilizzi RDS Proxy con un DB di istanze RDS con autenticazione IAM abilitata, verifica l'autenticazione dell'utente. Gli utenti che si connettono tramite un proxy devono eseguire

l'autenticazione con le credenziali di accesso. Per ulteriori informazioni sul supporto di Secrets Manager e IAM in Server proxy per Amazon RDS, consulta Configurazione delle credenziali del database in AWS Secrets Manager e Configurazione delle politiche AWS Identity and Access Management (IAM).

- Non puoi utilizzare RDS Proxy con DNS personalizzati quando utilizzi la convalida del nome host SSL.
- Ogni proxy può essere associato a una singola istanza database di destinazione. Tuttavia, è possibile associare più proxy alla stessa istanza .
- Qualsiasi istruzione con una dimensione del testo maggiore di 16 KB fa sì che il proxy effettui il pinning della sessione nella connessione corrente.
- Per alcune regioni sono presenti restrizioni relative alla zona di disponibilità (AZ) da considerare durante la creazione del proxy. La Regione Stati Uniti orientali (Virginia settentrionale) non supporta Server proxy per RDS nella zona di disponibilità `use1-az3`. La Regione Stati Uniti occidentali (California settentrionale) non supporta Server proxy per RDS nella zona di disponibilità `usw1-az2`. Quando selezioni le sottoreti durante la creazione del proxy, assicurati di non scegliere sottoreti nelle zone di disponibilità sopra menzionate.
- Attualmente, RDS Proxy non supporta alcuna chiave di contesto di condizione globale.

  Per ulteriori informazioni sulle chiavi di contesto delle condizioni globali, consulta Chiavi di contesto delle condizioni globali AWS nella Guida per l'utente di IAM.

Per le altre limitazioni di ciascun motore di database, consulta le sezioni riportate di seguito:

- Limitazioni aggiuntive per RDS per MariaDB
- Limitazioni aggiuntive per RDS per Microsoft SQL Server
- Limitazioni aggiuntive per RDS per MySQL
- Limitazioni aggiuntive per RDS per PostgreSQL

Limitazioni aggiuntive per RDS per MariaDB

Le seguenti limitazioni aggiuntive si applicano a Server proxy per RDS con database RDS per MariaDB:

- Attualmente, tutti i proxy sono in ascolto di MariaDB sulla porta 3306. I proxy si connettono ancora al database utilizzando la porta specificata nelle impostazioni del database.

- Non puoi utilizzare Server proxy per RDS con database MariaDB autogestiti nelle istanze Amazon EC2.
- Non puoi utilizzare il proxy RDS con un'istanza database di RDS per MariaDB con il parametro `read_only` nel suo gruppo di parametri DB impostato su 1.
- Il proxy RDS non supporta la modalità compressa MariadB. Ad esempio, non supporta la compressione utilizzata dalle opzioni `--compress` o `-C` del comando `mysql`.
- Alcune istruzioni e funzioni SQL possono modificare lo stato della connessione senza causare l'aggiunta. Per il comportamento del pinning più aggiornato, consulta Evitare il pinning.
- Il proxy RDS non supporta il plugin `auth_ed25519` MariaDB.
- Il proxy RDS non supporta Transport Layer Security (TLS) versione 1.3 per i database MariaDB.
- Le connessioni al database che elaborano un comando `GET DIAGNOSTIC` potrebbero restituire informazioni imprecise quando Server proxy per RDS riutilizza la stessa connessione al database per eseguire un'altra query. Questo può accadere quando Server proxy per RDS crea multiplex delle connessioni al database. Per ulteriori informazioni, consulta Panoramica dei concetti RDS Proxy.

> **Important**
>
> Per i proxy associati ai database MariaDB, non impostare il parametro di configurazione `sql_auto_is_null` su `true` o un valore diverso da zero nella query di inizializzazione. Ciò potrebbe causare un comportamento non corretto dell'applicazione.

Limitazioni aggiuntive per RDS per Microsoft SQL Server

Le seguenti limitazioni aggiuntive si applicano a Server proxy per RDS con database RDS per Microsoft SQL Server:

- Il numero di segreti di Gestione dei segreti da creare per un proxy dipende dal confronto utilizzato dall'istanza database. Supponi, ad esempio, che l'istanza database utilizzi il confronto con distinzione tra lettere maiuscole e minuscole. Se l'applicazione accetta sia "Admin" che "admin", il proxy necessita di due segreti separati. Per ulteriori informazioni sul confronto in SQL Server, consulta la documentazione di Microsoft SQL Server.
- Server proxy per RDS non supporta le connessioni che utilizzano Active Directory.

- Non puoi usare l'autenticazione IAM con i client che non supportano le proprietà dei token. Per ulteriori informazioni, consulta Considerazioni per la connessione a un proxy con Microsoft SQL Server.
- I risultati di `@@IDENTITY`, `@@ROWCOUNT` e `SCOPE_IDENTITY` non sono sempre accurati. Per ovviare al problema, recupera i valori nella stessa istruzione di sessione per assicurarti che restituiscano le informazioni corrette.
- Se la connessione utilizza Multiple Active Result Set (MARS), Server proxy per RDS non esegue le query di inizializzazione. Per informazioni su MARS, consulta la documentazione di Microsoft SQL Server.
- Attualmente, RDS Proxy non supporta RDS per le istanze DB di SQL Server eseguite sulla versione principale di SQL Server 2022.
- RDS Proxy non supporta RDS per le istanze DB di SQL Server eseguite sulla versione principale di SQL Server 2014.
- RDS Proxy non supporta le applicazioni client che non sono in grado di gestire più messaggi di risposta in un unico record TLS.

Limitazioni aggiuntive per RDS per MySQL

Le seguenti limitazioni aggiuntive si applicano a Server proxy per RDS con database RDS per MySQL:

- RDS Proxy non supporta i plugin di autenticazione MySQL `sha256_password` e `caching_sha2_password`. Questi plugin implementano l'hashing SHA-256 per le password dell'account utente.
- Attualmente, tutti i proxy sono in ascolto di MySQL sulla porta 3306. I proxy si connettono ancora al database utilizzando la porta specificata nelle impostazioni del database.
- Non puoi utilizzare RDS Proxy con database MySQL gestiti dal cliente nelle istanze EC2.
- Non è possibile utilizzare RDS Proxy con un'istanza database di RDS per MySQL con il parametro `read_only` nel suo gruppo di parametri database impostato su `1`.
- RDS Proxy non supporta la modalità compressa MySQL. Ad esempio, non supporta la compressione utilizzata dalle opzioni `--compress` o `-C` del comando `mysql`.
- Le connessioni al database che elaborano un comando `GET DIAGNOSTIC` potrebbero restituire informazioni imprecise quando Server proxy per RDS riutilizza la stessa connessione al database per eseguire un'altra query. Questo può accadere quando Server proxy per RDS crea multiplex delle connessioni al database.

- Alcune istruzioni e funzioni SQL, ad esempio, `SET LOCAL` possono modificare lo stato della connessione senza causare il pinning. Per il comportamento del pinning più aggiornato, consulta Evitare il pinning.
- L'utilizzo della `ROW_COUNT()` funzione in una query con più istruzioni non è supportato.
- RDS Proxy non supporta le applicazioni client che non sono in grado di gestire più messaggi di risposta in un unico record TLS.

> **Important**
>
> Per i proxy associati ai database MySQL, non impostare il parametro `sql_auto_is_null` di configurazione su `true` o un valore diverso da zero nella query di inizializzazione. Ciò potrebbe causare un comportamento non corretto dell'applicazione.

Limitazioni aggiuntive per RDS per PostgreSQL

Le seguenti limitazioni aggiuntive si applicano a Server proxy per RDS con database RDS per PostgreSQL:

- RDS Proxy non supporta i filtri di pinning della sessione per PostgreSQL.
- Attualmente, tutti i proxy sono in ascolto di PostgreSQL sulla porta 5432.
- Per PostgreSQL, RDS Proxy attualmente non supporta l'annullamento di una query da un client tramite l'emissione di una `CancelRequest`. Questo è il caso, ad esempio, di quando si annulla una query a esecuzione prolungata in una sessione psql interattiva utilizzando Ctrl+C.
- I risultati della funzione PostgreSQL lastval non sono sempre accurati. Per risolvere il problema, utilizzare l'istruzione INSERT con la clausola `RETURNING`.
- Server proxy per RDS attualmente non supporta la modalità di replica in streaming.
- Con RDS per PostgreSQL 16, le modifiche al valore influiscono esclusivamente sul processo di autenticazione `scram_iterations` tra il proxy e il database. In particolare, se si configura `ClientPasswordAuthType` to`scram-sha-256`, eventuali personalizzazioni apportate al `scram_iterations` valore non influiscono sull'autenticazione della password. client-to-proxy Invece, il valore di iterazione per l'autenticazione client-to-proxy tramite password è fissato a 4096.

> **Important**
>
> Per i proxy esistenti con database PostgreSQL, se si modifica l'autenticazione del database in modo da utilizzare solo `SCRAM`, il proxy diventa non disponibile per un massimo di 60 secondi. Per evitare il problema, procedi in uno dei seguenti modi:
>
> - Assicurati che il database consenta entrambe le autenticazioni `SCRAM` e `MD5`.
> - Per utilizzare solo l'autenticazione `SCRAM`, crea un nuovo proxy, esegui la migrazione del traffico dell'applicazione sul nuovo proxy, quindi elimina il proxy precedentemente associato al database.

Pianificazione sull'utilizzo di RDS Proxy

Puoi determinare quali tra le istanze DB, i cluster e le applicazioni possono trarre maggior vantaggio dall'utilizzo di RDS Proxy. Per fare ciò, considera questi fattori:

- Qualsiasi istanza database che genera errori del tipo "Troppe connessioni" è un buon candidato per l'associazione con un proxy. Questo è spesso caratterizzato da un valore elevato della `ConnectionAttempts` CloudWatch metrica. Il proxy consente alle applicazioni di aprire molte connessioni client, mentre il proxy gestisce un numero minore di connessioni di lunga durata all'istanza database.
- Per le istanze DB, che utilizzano classi di AWS istanze più piccole, come T2 o T3, l'uso di un proxy può aiutare a evitare condizioni. out-of-memory Può anche contribuire a ridurre il sovraccarico della CPU per stabilire connessioni. Queste condizioni possono verificarsi quando vi è un numero elevato di connessioni.
- Puoi monitorare determinati CloudWatch parametri di Amazon per determinare se un di istanze DB si avvicina a determinati tipi di limite. Questi limiti riguardano il numero di connessioni e la memoria associata alla gestione delle connessioni. Puoi anche monitorare determinati CloudWatch parametri per determinare se un di istanze DB gestisce molte connessioni di breve durata. L'apertura e la chiusura di tali connessioni possono determinare un sovraccarico delle prestazioni sul database. Per informazioni sui parametri da monitorare, consulta Monitoraggio dei parametri del proxy RDS con Amazon CloudWatch.
- AWS LambdaAnche le funzioni possono essere ben utilizzate con un proxy. Queste funzioni effettuano frequenti connessioni del database brevi che beneficiano del pool di connessioni

offerto da RDS Proxy. Puoi usufruire di qualsiasi autenticazione IAM già disponibile per le funzioni Lambda, invece di gestire le credenziali del database nel codice Lambda dell'applicazione.

- Le applicazioni che in genere aprono e chiudono un numero elevato di connessioni al database e non dispongono di meccanismi di pooling delle connessioni incorporati sono ottimi candidati per l'utilizzo di un proxy.
- Le applicazioni che mantengono un numero elevato di connessioni aperte per lunghi periodi sono in genere buoni candidati per l'utilizzo con un proxy. Applicazioni in ambiti come software as a service (SaaS) o e-commerce spesso riducono al minimo la latenza per le richieste del database lasciando aperte le connessioni.
- Potrebbe non essere stata adottata l'autenticazione IAM e Secrets Manager a causa della complessità di configurazione di tale autenticazione per tutte le istanze database. In tal caso, puoi abbandonare i metodi di autenticazione esistenti e delegare l'autenticazione a un proxy. Il proxy può applicare le policy di autenticazione per le connessioni client per applicazioni particolari. Puoi usufruire di qualsiasi autenticazione IAM già disponibile per le funzioni Lambda, invece di gestire le credenziali del database nel codice Lambda dell'applicazione.
- Server proxy per RDS può contribuire a rendere le applicazioni più resilienti e trasparenti agli errori del database. Server proxy per RDS ignora le cache del sistema dei nomi di dominio (DNS) per ridurre i tempi di failover fino al 66% per le istanze database Amazon RDS Multi-AZ. Server proxy per RDS inoltre instrada automaticamente il traffico a una nuova istanza database, preservando al contempo le connessioni dell'applicazione. In tal modo i failover sono più trasparenti per le applicazioni.

Concetti e terminologia RDS Proxy

Puoi semplificare la gestione delle connessioni per le istanze database Amazon RDS utilizzando Server proxy per RDS.

RDS Proxy gestisce il traffico di rete tra l'applicazione client e il database. Lo fa in modo attivo prima comprendendo il protocollo del database. Quindi regola il suo comportamento in base alle operazioni SQL dell'applicazione e ai set di risultati dal database.

RDS Proxy riduce il sovraccarico di memoria e CPU per la gestione delle connessioni nel database. Il database richiede meno memoria e risorse della CPU quando le applicazioni aprono molte connessioni simultanee. Inoltre, non richiede alcuna logica nelle applicazioni al fine di chiudere e riaprire le connessioni che rimangono inattive per un lungo periodo di tempo. Allo stesso modo,

richiede meno operazioni logiche dell'applicazione per ristabilire le connessioni in caso di problemi di database.

RDS Proxy è altamente disponibile e distribuito su più zone di disponibilità (AZ). Questa separazione consente di ridurre il sovraccarico sui server di database, in modo da poter dedicare le risorse al servizio dei carichi di lavoro del database. Le risorse di calcolo RDS Proxy sono serverless e vengono scalate automaticamente in base al carico di lavoro del database.

Argomenti

Panoramica dei concetti RDS Proxy

RDS Proxy gestisce l'infrastruttura per eseguire il pool di connessioni e le altre funzionalità descritte nelle sezioni seguenti. I proxy rappresentati nella console RDS vengono visualizzati nella pagina Proxy.

Ogni proxy gestisce le connessioni a un cluster DB a singola istanza RDS DB. Il proxy determina l'istanza di scrittura corrente per il cluster o l'istanza database Multi-AZ RDS.

Le connessioni che un proxy mantiene aperte e disponibili per le applicazioni di database da utilizzare costituiscono il pool di connessioni.

Per impostazione predefinita, RDS Proxy può riutilizzare una connessione dopo ogni transazione nella sessione. Questo riutilizzo a livello di transazione viene definito multiplexing. Quando RDS Proxy rimuove temporaneamente una connessione dal pool di connessioni per riutilizzarla, tale operazione viene chiamata prestito della connessione. Quando è sicuro farlo, RDS Proxy restituisce tale connessione al pool di connessioni.

In alcuni casi, per RDS Proxy non è possibile avere la certezza di riutilizzare una connessione al database al di fuori della sessione corrente. In questi casi, mantiene la sessione sulla stessa connessione fino al termine della sessione. Questo comportamento di fallback viene definito pinning.

Un proxy ha un endpoint predefinito. Ti connetti a questo endpoint quando lavori con un'istanza database RDS invece di connetterti all'endpoint di lettura-scrittura che si connette direttamente all'istanza . Per RDS DB, puoi anche creare endpoint di lettura/scrittura e di sola lettura aggiuntivi. Per ulteriori informazioni, consulta Panoramica degli endpoint proxy.

Ad esempio, puoi comunque connetterti all'endpoint del cluster per le connessioni di lettura-scrittura senza pool di connessioni. Puoi comunque connetterti all'endpoint di lettura per le connessioni di sola lettura con bilanciamento del carico. Puoi comunque connetterti agli endpoint dell'istanza per la diagnosi e la risoluzione dei problemi di istanze database specifiche all'interno di un cluster. Se utilizzi altri AWS servizi, ad esempio AWS Lambda per connetterti ai database RDS, modifica le relative impostazioni di connessione per utilizzare l'endpoint proxy. Ad esempio, se specifichi l'endpoint proxy per consentire alle funzioni Lambda di accedere al database sfruttando al contempo le funzionalità RDS Proxy.

Ogni proxy contiene un gruppo di destinazione. Questo gruppo target include il a cui il proxy può connettersi. Il cluster DB dell'istanza RDS associato a un proxy viene chiamato target di tale proxy. Per comodità, quando crei un proxy attraverso la console, RDS Proxy crea anche il gruppo di destinazione corrispondente e registra automaticamente le destinazioni associate.

Una famiglia di motori è un insieme correlato di motori di database che utilizzano lo stesso protocollo di DB. Scegli la famiglia di motori per ogni proxy creato.

Pooling di connessioni

Ogni proxy esegue il pool di connessioni per l'istanza di scrittura del relativo database RDS associato. Il pool di connessioni è un'ottimizzazione che riduce il sovraccarico associato all'apertura e alla chiusura delle connessioni e mantiene aperte contemporaneamente molte connessioni. Questo sovraccarico include la memoria necessaria per gestire ogni nuova connessione. Ciò implica un sovraccarico della CPU anche per chiudere la connessione e aprirne una nuova. Gli esempi includono l'handshaking Transport Layer Security/Secure Sockets Layer (TLS/SSL), l'autenticazione, le capacità di negoziazione e così via. Il pool di connessioni semplifica la logica dell'applicazione. Non devi scrivere codice dell'applicazione per ridurre al minimo il numero di connessioni aperte simultanee.

Ogni proxy esegue anche il multiplexing delle connessioni, noto anche come riutilizzo della connessione. Con il multiplexing, Server proxy per RDS esegue tutte le operazioni per una transazione utilizzando una connessione al database sottostante, quindi può utilizzare una connessione diversa per la transazione successiva. Puoi aprire molte connessioni simultanee al proxy e il proxy mantiene un numero minore di connessioni aperte all'istanza database o al cluster.

In questo modo si riduce ulteriormente il sovraccarico di memoria per le connessioni sul server di database. Questa tecnica riduce anche la possibilità di errori del tipo «troppe connessioni».

Sicurezza di RDS Proxy

RDS Proxy utilizza i meccanismi di sicurezza RDS esistenti come TLS/SSL e AWS Identity and Access Management (IAM). Per informazioni generali sulle funzionalità di sicurezza, consulta Sicurezza in Amazon RDS. Inoltre, assicurati di familiarizzare con il modo in cui RDS utilizza l'autenticazione, l'autorizzazione e altri metodi di protezione.

RDS Proxy può fungere da ulteriore livello di sicurezza tra le applicazioni client e il database sottostante. Ad esempio, è possibile connettersi al proxy utilizzando TLS 1.3, anche se l'istanza DB sottostante supporta una versione precedente di TLS. Puoi connetterti al proxy utilizzando un ruolo IAM, anche se il proxy si connette al database utilizzando il metodo di autenticazione utente/password nativo. Utilizzando questa tecnica, puoi applicare requisiti di autenticazione avanzata per le applicazioni di database senza un costoso sforzo di migrazione per le istanze DB medesime.

Le credenziali del database utilizzate da RDS Proxy vengono archiviate in. AWS Secrets Manager Ogni utente del database per il cluster DB dell'istanza RDS a cui accede un proxy deve disporre di un segreto corrispondente in Secrets Manager. Puoi inoltre impostare l'autenticazione IAM per gli utenti di RDS Proxy. In questo modo, puoi applicare l'autenticazione IAM per l'accesso al database anche se i database utilizzano ancora l'autenticazione con password nativa. È consigliabile utilizzare queste funzionalità di protezione anziché incorporare le credenziali del database nel codice dell'applicazione.

Utilizzo di TLS/SSL con RDS Proxy

È possibile connettersi a RDS Proxy utilizzando il protocollo TLS/SSL.

> **Note**
>
> RDS Proxy utilizza i certificati di AWS Certificate Manager (ACM). Se si utilizza RDS Proxy, non è necessario scaricare certificati Amazon RDS o aggiornare applicazioni che utilizzano connessioni RDS Proxy.

Per applicare il TLS per tutte le connessioni tra il proxy e il database, è possibile specificare un'impostazione Require Transport Layer Security quando si crea o si modifica un proxy in. AWS Management Console

RDS Proxy può garantire che la sessione utilizzi TLS/SSL tra il client e l'endpoint RDS Proxy. Per fare in modo che RDS Proxy proceda, specificare il requisito sul lato client. Le variabili di sessione SSL non sono configurate per le connessioni SSL a un database che utilizza RDS Proxy.

- Per RDS per MySQL, specifica il requisito sul lato client con il parametro `--ssl-mode` quando esegui il comando `mysql`.
- Per Amazon RDS PostgreSQL, specifica `sslmode=require` come parte della stringa `conninfo` quando esegui il comando `psql`.

RDS Proxy supporta le versioni 1.0, 1.1, 1.2 e 1.3 del protocollo TLS. È possibile connettersi al proxy utilizzando una versione superiore di TLS rispetto a quella utilizzata nel database sottostante.

Per impostazione predefinita, i programmi client stabiliscono una connessione crittografata con RDS Proxy, con controllo aggiuntivo disponibile tramite l'opzione `--ssl-mode`. Dal lato client, RDS Proxy supporta tutte le modalità SSL.

Per il client, le modalità SSL sono le seguenti:

PREFERRED

SSL è la prima scelta, ma non obbligatoria.

DISABLED

Nessun SSL è abilitato.

REQUIRED

Applica SSL.

VERIFY_CA

Applicare SSL e verificare l'autorità di certificazione (CA).

VERIFY_IDENTITY

Applica SSL e verifica CA e nome host CA.

Quando si utilizza un client con `--ssl-mode` `VERIFY_CA` o `VERIFY_IDENTITY`, specificare l'opzione `--ssl-ca` puntando a una CA in formato `.pem`. Per il file `.pem` da usare, scarica tutti i PEM CA root da Amazon Trust Services e inseriscili in un singolo file `.pem`.

RDS Proxy utilizza certificati wildcard, che si applicano sia a un dominio che ai relativi sottodomini. Se utilizzi il client `mysql` per eseguire la connessione in modalità SSL `VERIFY_IDENTITY`, al momento devi utilizzare il comando compatibile `mysql` con MySQL 8.0.

Failover

Il failover è una funzionalità ad alta disponibilità che sostituisce un'istanza di database con un'altra quando l'istanza originale diventa non disponibile. Un failover potrebbe verificarsi a causa di un problema con un'istanza di database. Potrebbe anche essere parte delle normali procedure di manutenzione, ad esempio durante un aggiornamento del database. Il failover si applica alle istanze database RDS in una configurazione Multi-AZ.

La connessione tramite un proxy rende le applicazioni più resistenti ai failover del database. Quando l'istanza database originale diventa non disponibile, RDS Proxy si connette al database di standby senza far cadere le connessioni dell'applicazione inattiva. Ciò consente di velocizzare e semplificare il processo di failover. Ciò comporta meno interruzioni per l'applicazione rispetto a un tipico problema di riavvio o di database.

Senza RDS Proxy, un failover comporta una breve interruzione. Durante l'interruzione, non è possibile eseguire operazioni di scrittura sul database in caso di failover. Tutte le connessioni al database esistenti vengono interrotte e l'applicazione deve riaprirle. Il database diventa disponibile per le nuove connessioni e le operazioni di scrittura quando un'istanza database di sola lettura viene promossa al posto di quella non disponibile.

Durante i failover DB, RDS Proxy continua ad accettare connessioni allo stesso indirizzo IP e indirizza automaticamente le connessioni alla nuova istanza database primaria. I client che si connettono tramite RDS Proxy non sono soggetti a quanto segue:

- Ritardi di propagazione DNS (Domain Name System) durante il failover.
- Cache DNS locale.
- Timeout di connessione.
- Incertezza su quale istanza database è il writer corrente.
- In attesa di una risposta di query da un precedente writer che è diventato non disponibile senza chiudere le connessioni.

Per le applicazioni che mantengono il proprio pool di connessioni, passare attraverso RDS Proxy significa che la maggior parte delle connessioni rimane attiva durante i failover o altre interruzioni.

Vengono annullate solo le connessioni che si trovano nel mezzo di una transazione o istruzione SQL. RDS Proxy accetta immediatamente nuove connessioni. Quando l'istanza di scrittura del database non è disponibile, RDS Proxy accoda le richieste in arrivo.

Per le applicazioni che non mantengono i propri pool di connessioni, RDS Proxy offre velocità di connessione più rapide e connessioni più aperte. Si evita il costoso sovraccarico dovuto a frequenti riconnessioni dal database. Ciò avviene riutilizzando le connessioni al database mantenute nel pool di connessioni del RDS Proxy. Questo approccio è particolarmente importante per le connessioni TLS, dove i costi di installazione sono significativi.

Transazioni

Tutte le istruzioni all'interno di una singola transazione utilizzano sempre la stessa connessione al database sottostante. La connessione diventa disponibile per l'utilizzo da parte di una sessione diversa al termine della transazione. L'utilizzo della transazione come unità di granularità ha le seguenti conseguenze:

- Il riutilizzo della connessione può avvenire dopo ogni singola istruzione quando l'impostazione RDS per MySQL `autocommit` è attivata.
- Al contrario, quando l'impostazione `autocommit` è disattivata, la prima istruzione che viene emessa in una sessione inizia una nuova transazione. Ad esempio, supponi di immettere una sequenza di `SELECT`, `INSERT`, `UPDATE` e altre istruzioni DML (Data Manipulation Language). In questo caso, il riutilizzo della connessione non avviene fino a quando non invii un'istruzione come `COMMIT` o `ROLLBACK` per terminare la transazione.
- L'immissione di un'istruzione DDL (Data Definition Language) fa terminare la transazione dopo il completamento dell'istruzione.

RDS Proxy rileva quando una transazione termina attraverso il protocollo di rete utilizzato dall'applicazione client del database. Il rilevamento delle transazioni non si basa su parole chiave come `COMMIT` o `ROLLBACK` che appaiono nel testo dell'istruzione SQL.

In alcuni casi, RDS Proxy potrebbe rilevare una richiesta di database che rende impossibile spostare la sessione a una connessione diversa. In questi casi, disattiva il multiplexing per quella connessione al resto della sessione. La stessa regola si applica se RDS Proxy non può avere la certezza che il multiplexing sia praticabile per la sessione. Questa operazione è chiamata pinning. Per informazioni su come rilevare e ridurre al minimo il pinning, consulta Evitare il pinning.

Nozioni di base su RDS Proxy

Nelle sezioni seguenti, puoi scoprire come configurare e gestire RDS Proxy. Puoi anche scoprire come impostare le opzioni di sicurezza correlate che Queste opzioni controllano chi può accedere a ciascun proxy e in che modo ogni proxy si connette alle istanze DB.

Argomenti

Configurazione dei prerequisiti di rete

L'utilizzo di RDS Proxy richiede la disponibilità di un cloud privato virtuale (VPC) comune tra l'istanza RDS DB, il cluster DB e il proxy RDS. Questo VPC deve avere un minimo di due sottoreti che si trovano in zone di disponibilità diverse. Il tuo account può possedere queste sottoreti o condividerle con altri account. Per ulteriori informazioni sui VPC condivisi, consultare Utilizzo dei VPC condivisi.

Le risorse dell'applicazione client come Amazon EC2, Lambda o Amazon ECS possono trovarsi nello stesso VPC del proxy. In alternativa, possono trovarsi in un VPC separato dal proxy. Se hai effettuato correttamente la connessione a un'istanza database RDS , sono già disponibili le risorse di rete necessarie.

Argomenti

Recupero delle informazioni sulle sottoreti

Per creare un proxy, è necessario fornire le sottoreti e il VPC in cui opera il proxy. Il seguente esempio di Linux mostra AWS CLI i comandi che esaminano i VPC e le sottoreti di proprietà dell'utente. Account AWS In particolare, si passano gli ID delle sottoreti come parametri quando si crea un proxy utilizzando la CLI.

```
aws ec2 describe-vpcs
aws ec2 describe-internet-gateways
aws ec2 describe-subnets --query '*[].[VpcId,SubnetId]' --output text | sort
```

Il seguente esempio di Linux mostra AWS CLI i comandi per determinare gli ID di sottorete corrispondenti a una specifica istanza RDS DB del cluster DB. Trova l'ID VPC per l'istanza DB. Esamina il VPC per trovarne le sottoreti. Il seguente esempio di Linux mostra come.

```
$ #From the DB instance, trace through the DBSubnetGroup and Subnets to find the subnet
 IDs.
$ aws rds describe-db-instances --db-instance-identifier my_instance_id --query '*[].
[DBSubnetGroup]||[0]||[0]||[Subnets]||[0]||[*].SubnetIdentifier' --output text
```

```
subnet_id_1
subnet_id_2
subnet_id_3
...
```

```
$ #From the DB instance, find the VPC.
$ aws rds describe-db-instances --db-instance-identifier my_instance_id --query '*[].
[DBSubnetGroup]||[0]||[0].VpcId' --output text
```

```
my_vpc_id
```

```
$ aws ec2 describe-subnets --filters Name=vpc-id,Values=my_vpc_id --query '*[].
[SubnetId]' --output text
```

```
subnet_id_1
subnet_id_2
subnet_id_3
subnet_id_4
subnet_id_5
subnet_id_6
```

Pianificazione della capacità degli indirizzi IP

Server proxy per RDS regola automaticamente la sua capacità secondo le necessità in base alle dimensioni e al numero di istanze database registrate. Alcune operazioni potrebbero richiedere anche

una capacità proxy aggiuntiva, ad esempio l'aumento delle dimensioni di un database registrato o operazioni di manutenzione interne del proxy RDS. Durante queste operazioni, il proxy può aver bisogno di più indirizzi IP per fornire la capacità aggiuntiva. Questi indirizzi aggiuntivi consentono al proxy di dimensionare senza interessare il carico di lavoro. La mancanza di indirizzi IP liberi nelle sottoreti impedisce al proxy di aumentare le dimensioni, causando eventuali latenze elevate nell'esecuzione di query o errori di connessione del client. RDS ti avvisa tramite l'evento `RDS-EVENT-0243` quando non ci sono abbastanza indirizzi IP liberi nelle tue sottoreti. Per informazioni su questo evento, consulta Utilizzo degli eventi RDS Proxy.

Di seguito sono riportati i numeri minimi consigliati di indirizzi IP da lasciare liberi nelle sottoreti per il proxy in base alle dimensioni delle classi delle istanze DB.

DB instance class (Classe istanza database)	Numero minimo di indirizzi IP liberi
db.*.xlarge o più piccola	10
db.*.2xlarge	15
db.*.4xlarge	25
db.*.8xlarge	45
db.*.12xlarge	60
db.*.16xlarge	75
db.*.24xlarge	110

Questi numeri consigliati di indirizzi IP sono stime per un proxy con solo l'endpoint predefinito. Un proxy con endpoint aggiuntivi o repliche di lettura potrebbe richiedere più indirizzi IP liberi. Per ogni endpoint aggiuntivo, ti consigliamo di riservare altri tre indirizzi IP. Per ogni replica di lettura, ti consigliamo di riservare gli indirizzi IP aggiuntivi specificati nella tabella in base alle dimensioni della replica di lettura.

> **Note**
>
> Il proxy RDS non supporta più di 215 indirizzi IP in un VPC.

Configurazione delle credenziali del database in AWS Secrets Manager

Per ogni proxy creato, utilizza innanzitutto il servizio Secrets Manager per memorizzare set di credenziali composti da nome utente e password. Si crea un segreto Secrets Manager separato per ogni account utente del database a cui il proxy si connette sul cluster DB dell'istanza RDS DB.

Nella console Secrets Manager, crei questi segreti con valori per i `password` campi `username` e. In questo modo il proxy può connettersi agli utenti del database corrispondenti su un cluster associato al proxy. A tale scopo, puoi utilizzare l'impostazione Credentials for other database (Credenziali per altri database), Credentials for RDS database (Credenziali per database RDS) o Other type of secrets (Altro tipo di segreti). Inserisci i valori appropriati per i campi Nome utente e Password e i valori per tutti gli altri campi obbligatori. Il proxy ignora altri campi, ad esempio Host e Port (Porta) se sono presenti nel segreto. Tali dettagli sono forniti automaticamente dal proxy.

Puoi anche scegliere Other type of secrets (Altro tipo di segreti). In questo caso, crei il segreto con le chiavi denominate `username` e `password`.

Poiché i segreti usati dal proxy non sono legati a un server di database specifico, puoi riutilizzare un segreto in più proxy. Per farlo, devi utilizzare le stesse credenziali nei diversi server di database. Ad esempio, è possibile utilizzare le stesse credenziali su server di sviluppo e test.

Per connetterti tramite il proxy come utente specifico del database, assicurati che la password associata a un segreto corrisponda alla password del database di quell'utente. In caso di mancata corrispondenza, è possibile aggiornare il segreto associato in Secrets Manager. In questo caso, è comunque possibile connettersi ad altri account in cui le credenziali segrete e le password del database corrispondono.

> **Note**
>
> Per RDS per SQL Server, RDS Proxy necessita di un segreto in Secrets Manager che faccia distinzione tra maiuscole e minuscole per il codice dell'applicazione indipendentemente dalle impostazioni di confronto delle istanze DB. Ad esempio, se l'applicazione può utilizzare entrambi i nomi utente «Admin» o «admin», configura il proxy con segreti sia per «Admin» che per «admin». RDS Proxy non supporta la distinzione tra maiuscole e minuscole per nome utente nel processo di autenticazione tra client e proxy.
> Per ulteriori informazioni sul confronto in SQL Server, consulta la documentazione di Microsoft SQL Server.

Quando crei un proxy tramite l'API AWS CLI o RDS, specifichi gli Amazon Resource Names (ARN) dei segreti corrispondenti. per tutti gli account utente del database a cui il proxy può accedere. In AWS Management Console, scegli i segreti in base ai loro nomi descrittivi.

Per istruzioni sulla creazione di segreti in Secrets Manager, consulta la pagina Creazione di un segreto nella documentazione di Secrets Manager. Puoi utilizzare una delle seguenti tecniche:

- Utilizza Secrets Manager nella console.
- Per utilizzare la CLI al fine di creare un segreto Secrets Manager da utilizzare con RDS Proxy, utilizza un comando come il seguente.

```
aws secretsmanager create-secret
  --name "secret_name"
  --description "secret_description"
  --region region_name
  --secret-string '{"username":"db_user","password":"db_user_password"}'
```

- Puoi anche creare una chiave personalizzata per cifrare il tuo segreto di Secrets Manager. Il comando seguente crea una chiave di esempio.

```
PREFIX=my_identifier
aws kms create-key --description "$PREFIX-test-key" --policy '{
  "Id":"$PREFIX-kms-policy",
  "Version":"2012-10-17",
  "Statement":
    [
      {
        "Sid":"Enable IAM User Permissions",
        "Effect":"Allow",
        "Principal":{"AWS":"arn:aws:iam::account_id:root"},
        "Action":"kms:*","Resource":"*"
      },
      {
        "Sid":"Allow access for Key Administrators",
        "Effect":"Allow",
        "Principal":
          {
            "AWS":
              ["$USER_ARN","arn:aws:iam:account_id::role/Admin"]
          },
        "Action":
          [
```

```
            "kms:Create*",
            "kms:Describe*",
            "kms:Enable*",
            "kms:List*",
            "kms:Put*",
            "kms:Update*",
            "kms:Revoke*",
            "kms:Disable*",
            "kms:Get*",
            "kms:Delete*",
            "kms:TagResource",
            "kms:UntagResource",
            "kms:ScheduleKeyDeletion",
            "kms:CancelKeyDeletion"
          ],
        "Resource":"*"
      },
      {
        "Sid":"Allow use of the key",
        "Effect":"Allow",
        "Principal":{"AWS":"$ROLE_ARN"},
        "Action":["kms:Decrypt","kms:DescribeKey"],
        "Resource":"*"
      }
    ]
}'
```

Ad esempio, i seguenti comandi creano segreti di Secrets Manager per due utenti del database:

```
aws secretsmanager create-secret \
  --name secret_name_1 --description "db admin user" \
  --secret-string '{"username":"admin","password":"choose_your_own_password"}'

aws secretsmanager create-secret \
  --name secret_name_2 --description "application user" \
  --secret-string '{"username":"app-user","password":"choose_your_own_password"}'
```

Per creare questi segreti crittografati con la tua AWS KMS chiave personalizzata, usa i seguenti comandi:

```
aws secretsmanager create-secret \
  --name secret_name_1 --description "db admin user" \
```

```
  --secret-string '{"username":"admin","password":"choose_your_own_password"}'
  --kms-key-id arn:aws:kms:us-east-2:account_id:key/key_id

aws secretsmanager create-secret \
  --name secret_name_2 --description "application user" \
  --secret-string '{"username":"app-user","password":"choose_your_own_password"}'
  --kms-key-id arn:aws:kms:us-east-2:account_id:key/key_id
```

Per visualizzare i segreti di proprietà del tuo AWS account, usa un comando come il seguente.

```
aws secretsmanager list-secrets
```

Quando crei un proxy utilizzando la CLI, invii gli ARN (Amazon Resource Names) di uno o più segreti al parametro `--auth`. Il seguente esempio di Linux mostra come preparare un rapporto con solo il nome e l'ARN di ogni segreto di proprietà dell'account AWS . In questo esempio viene utilizzato il parametro `--output table` disponibile in AWS CLI versione 2. Se stai usando la AWS CLI versione 1, usa `--output text` invece.

```
aws secretsmanager list-secrets --query '*[].[Name,ARN]' --output table
```

Per verificare di aver memorizzato le credenziali corrette e nel formato corretto in un segreto, utilizza un comando come il seguente. Sostituisci il nome breve o l'ARN del segreto con *your_secret_name*.

```
aws secretsmanager get-secret-value --secret-id your_secret_name
```

L'output dovrebbe includere una riga che visualizza un valore codificato JSON come il seguente.

```
"SecretString": "{\"username\":\"your_username\",\"password\":\"your_password\"}",
```

Configurazione delle politiche AWS Identity and Access Management (IAM)

Dopo aver creato i segreti in Secrets Manager, puoi creare una policy IAM in grado di accedere a tali segreti. Per informazioni generali sull'utilizzo di IAM, consulta Gestione accessi e identità per Amazon RDS.

> **Tip**
>
> La procedura seguente si applica se si utilizza la console IAM. Se utilizzi la AWS Management Console per RDS, RDS può creare automaticamente la policy IAM per te. In tal caso, è possibile ignorare la seguente procedura.

Per creare una policy IAM che acceda ai segreti Secrets Manager da utilizzare con il proxy

1. Accedere alla console IAM. Segui il processo di creazione del ruolo, come descritto in Creazione di ruoli IAM, scegliendo Creazione di un ruolo per delegare le autorizzazioni a un servizio. AWS

   Scegli Servizio AWS in Tipo di entità attendibile. In Caso d'uso, seleziona RDS nell'elenco a discesa Casi d'uso per altri servizi AWS . Scegli RDS – Aggiungi ruolo al database.

2. Per il nuovo ruolo, esegui il passaggio Add inline policy (Aggiungi policy inline) . Utilizzare le stesse procedure generali di Modifica dei criteri IAM. Incollare il seguente JSON nella casella di testo JSON. Sostituire l'ID account. Sostituisci la tua regione con. AWS `us-east-2` Sostituisci il nome della risorsa Amazon (ARN) dei segreti creati, consulta Specificazione delle chiavi KMS nelle istruzioni della policy IAM. Per l'`kms:Decrypt`azione, sostituisci l'ARN AWS KMS key predefinito o la tua chiave KMS. La scelta dipende da quale hai usato per crittografare i segreti di Gestione dei segreti.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": "secretsmanager:GetSecretValue",
            "Resource": [
                "arn:aws:secretsmanager:us-east-2:account_id:secret:secret_name_1",
                "arn:aws:secretsmanager:us-east-2:account_id:secret:secret_name_2"
            ]
        },
        {
            "Sid": "VisualEditor1",
            "Effect": "Allow",
            "Action": "kms:Decrypt",
            "Resource": "arn:aws:kms:us-east-2:account_id:key/key_id",
            "Condition": {
```

```
                "StringEquals": {
                    "kms:ViaService": "secretsmanager.us-east-2.amazonaws.com"
                }
            }
        }
    ]
}
```

3. Modifica la policy di attendibilità per questo ruolo IAM. Incollare il seguente JSON nella casella di testo JSON.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "rds.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Il seguente comando esegue la stessa operazione tramite AWS CLI.

```
PREFIX=my_identifier
USER_ARN=$(aws sts get-caller-identity --query "Arn" --output text)

aws iam create-role --role-name my_role_name \
  --assume-role-policy-document '{"Version":"2012-10-17","Statement":
[{"Effect":"Allow","Principal":{"Service":
["rds.amazonaws.com"]},"Action":"sts:AssumeRole"}]}'

ROLE_ARN=arn:aws:iam::account_id:role/my_role_name

aws iam put-role-policy --role-name my_role_name \
  --policy-name $PREFIX-secret-reader-policy --policy-document '{
    "Version": "2012-10-17",
    "Statement": [
        {
```

```
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": "secretsmanager:GetSecretValue",
            "Resource": [
                "arn:aws:secretsmanager:us-east-2:account_id:secret:secret_name_1",
                "arn:aws:secretsmanager:us-east-2:account_id:secret:secret_name_2"
            ]
        },
        {
            "Sid": "VisualEditor1",
            "Effect": "Allow",
            "Action": "kms:Decrypt",
            "Resource": "arn:aws:kms:us-east-2:account_id:key/key_id",
            "Condition": {
                "StringEquals": {
                    "kms:ViaService": "secretsmanager.us-east-2.amazonaws.com"
                }
            }
        }
    ]
}
```

Creazione di un RDS Proxy

Per gestire le connessioni per un set specificato di istanze DB, puoi creare un proxy. Puoi associare un proxy a un'istanza database RDS per MariaDB, RDS per Microsoft SQL Server, RDS per MySQL o RDS per PostgreSQL.

AWS Management Console

Per creare un proxy

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione scegli Proxies (Proxy).
3. Scegli Create proxy (Crea proxy).
4. Scegli tutte le impostazioni per il tuo proxy.

   Per la configurazione del proxy, fornire informazioni per quanto segue:

- Famiglia di motori. Questa impostazione determina il protocollo di rete del database riconosciuto dal proxy quando interpreta il traffico di rete verso e dal database. Per RDS per MariaDB o RDS per MySQL, scegli MariaDB and MySQL (MariaDB e MySQL). Per RDS per PostgreSQL, scegli PostgreSQL. Per RDS per SQL Server, scegli SQL Server.
- Identificatore proxy. Specificane un nome univoco all'interno dell'ID AWS dell'account e AWS della regione corrente.
- Timeout della connessione client per inattività. Scegli un periodo di tempo in cui una connessione client può rimanere inattiva prima che il proxy la chiuda. Il valore predefinito è 1.800 secondi (30 minuti). Una connessione client è considerata inattiva quando l'applicazione non invia una nuova richiesta entro il tempo specificato dopo il completamento della richiesta precedente. La connessione al database sottostante rimane aperta e viene restituita al pool di connessioni. Pertanto, è disponibile per essere riutilizzata per nuove connessioni client.

  Per fare in modo che il proxy rimuova in modo proattivo le connessioni obsolete, riducete il timeout della connessione del client inattivo. Quando il carico di lavoro aumenta, per risparmiare sui costi di creazione delle connessioni, aumenta il timeout della connessione dei client inattivi».

Per la configurazione del gruppo di destinazione, fornire informazioni per quanto segue:

- Database. Scegli un'istanza RDS DB cluster a cui accedere tramite questo proxy. L'elenco include solo istanze DB e cluster con motori di database compatibili, versioni del motore e altre impostazioni. Se l'elenco è vuoto, crea una nuova istanza database o un cluster compatibile con RDS Proxy. A tale scopo, segui la procedura in Creazione di un'istanza database Amazon RDS. Quindi prova a creare nuovamente il proxy.
- Connessioni massime del pool di connessioni. Specificare un valore compreso tra 1 e 100. Questa impostazione rappresenta la percentuale del valore `max_connections` che RDS Proxy può utilizzare per le relative connessioni. Se intendi utilizzare un solo proxy con questa istanza database o cluster, puoi impostarlo su 100. Per informazioni dettagliate su come RDS Proxy utilizza questa impostazione, consulta MaxConnectionsPercent.
- Filtri di pinning della sessione. (Facoltativo) Questa opzione consente di forzare Server proxy per Amazon RDS a non eseguire il pin per determinati tipi di stati di sessione rilevati. Ciò elude le misure di sicurezza predefinite per il multiplexing delle connessioni al database tra connessioni client. Attualmente, l'impostazione non è supportata per PostgreSQL. L'unica scelta è. `EXCLUDE_VARIABLE_SETS`

L'attivazione di questa impostazione può far sì che le variabili di sessione di una connessione influiscano sulle altre connessioni. Ciò può causare errori o problemi di correttezza se le query dipendono dai valori delle variabili di sessione impostati al di fuori della transazione corrente. Valuta l'utilizzo di questa opzione dopo aver verificato che sia sicuro per le applicazioni condividere le connessioni al database tra connessioni client.

I seguenti modelli possono essere considerati sicuri:

- Istruzioni SET in cui non viene apportata alcuna modifica al valore della variabile di sessione effettiva, ovvero non viene apportata alcuna modifica alla variabile di sessione.
- Modifichi il valore della variabile di sessione ed esegui un'istruzione nella stessa transazione.

Per ulteriori informazioni, consulta Evitare il pinning.

- Timeout del prestito di connessione. In alcuni casi, è possibile che il proxy utilizzi talvolta tutte le connessioni di database disponibili. In questi casi, è possibile specificare per quanto tempo il proxy attende che una connessione di database diventi disponibile prima di restituire un errore di timeout. È possibile specificare un periodo fino a un massimo di cinque minuti. Questa impostazione si applica solo quando il proxy ha il numero massimo di connessioni aperte e tutte le connessioni sono già in uso.
- Query di inizializzazione. (Opzionale) Puoi specificare una o più istruzioni SQL per l'esecuzione del proxy all'apertura di ogni nuova connessione al database. L'impostazione viene in genere utilizzata con SET le istruzioni per assicurarsi che ogni connessione abbia impostazioni identiche, come il fuso orario e i set di caratteri. Per più istruzioni, utilizzare il punto e virgola come separatore. È inoltre possibile includere più variabili in una singola istruzione SET, ad esempio `SET x=1, y=2`.

Per l'autenticazione, fornisci informazioni per quanto segue:

- Ruolo IAM. Scegli un ruolo IAM che disponga dell'autorizzazione per accedere ai segreti Secrets Manager scelti in precedenza. In alternativa, puoi creare un nuovo ruolo IAM da AWS Management Console.
- Segreti di Secrets Manager. Scegli almeno un segreto di Secrets Manager che contenga le credenziali utente del database che consentano al proxy di accedere al cluster DB dell'istanza RDS DB.

- Tipo di autenticazione client. Scegli il tipo di autenticazione utilizzato dal proxy per le connessioni dei client. La tua scelta si applica a tutti i segreti di Secrets Manager che associ a questo proxy. Se devi specificare un tipo di autenticazione client diverso per ogni segreto, crea il proxy utilizzando invece l'API AWS CLI o l'API.
- Autenticazione IAM. Scegli se richiedere, consentire o non consentire l'autenticazione IAM per le connessioni al proxy. L'opzione per consentire è valida solo per i proxy per RDS per SQL Server. La tua scelta si applica a tutti i segreti di Secrets Manager che associ a questo proxy. Se devi specificare un'autenticazione IAM diversa per ogni segreto, crea il proxy utilizzando invece l'API AWS CLI o l'API.

Per Connettività, fornire informazioni per quanto segue:

- Richiedi Transport Layer Security. Scegli questa impostazione se desideri che il proxy applichi TLS/SSL per tutte le connessioni client. Per una connessione crittografata o non crittografata a un proxy, il proxy utilizza la stessa impostazione di crittografia quando effettua la connessione al database sottostante.
- Sottoreti. Questo campo è precompilato con tutte le sottoreti associate al VPC. Rimuovere le sottoreti non necessarie per questo proxy. Devi lasciare almeno due sottoreti.

Configurazione di connettività aggiuntiva:

- Gruppo di sicurezza VPC. Scegli un gruppo di sicurezza VPC esistente. In alternativa, puoi creare un nuovo gruppo di sicurezza da AWS Management Console. È necessario configurare le regole in entrata per consentire alle applicazioni di accedere al proxy. È inoltre necessario configurare le regole in uscita per consentire il traffico proveniente dalle destinazioni del database.

> **Note**
>
> Questo gruppo di sicurezza deve consentire le connessioni dal proxy al database. Lo stesso gruppo di protezione viene utilizzato per l'ingresso dalle applicazioni al proxy e per l'uscita dal proxy al database. Si supponga, ad esempio, di utilizzare lo stesso gruppo di protezione per il database e il proxy. In questo caso, assicurarsi di specificare che le risorse di tale gruppo di protezione possono comunicare con altre risorse dello stesso gruppo di protezione.

Quando si utilizza un VPC condiviso, non è possibile utilizzare il gruppo di sicurezza predefinito per il VPC o uno appartenente a un altro account. Scegli un gruppo di sicurezza appartenente all'account. Se non esiste, creane uno. Per ulteriori informazioni su questa limitazione, consulta Utilizzo di VPC condivisi.

RDS implementa un proxy su più zone di disponibilità per garantire una disponibilità elevata. Per abilitare la comunicazione tra zone di disponibilità per un proxy di questo tipo, la lista di controllo degli accessi (ACL) alla rete della sottorete proxy deve consentire l'uscita specifica della porta del motore e l'ingresso di tutte le porte. Per ulteriori informazioni sulle ACL di rete, consulta Come controllare il traffico verso le sottoreti utilizzando le liste di controllo degli accessi di rete. Se l'ACL di rete per il proxy e la destinazione sono identici, devi aggiungere una regola di ingresso del protocollo TCP in cui Origine è impostata sul VPC CIDR. È inoltre necessario aggiungere una regola di uscita del protocollo TCP specifica per la porta del motore in cui la destinazione è impostata sul VPC CIDR.

(Facoltativo) Fornire una configurazione avanzata:

- Enable enhanced logging (Abilita registrazione avanzata. Puoi attivare questa impostazione per risolvere problemi di compatibilità o prestazioni del proxy.

  Quando questa impostazione è abilitata, RDS Proxy include informazioni dettagliate sulle prestazioni del proxy nei suoi registri. Queste informazioni consentono di eseguire il debug di problemi relativi al comportamento SQL o alle prestazioni e alla scalabilità delle connessioni proxy. Pertanto, abilita questa impostazione solo per il debug e quando sono state adottate misure di sicurezza per salvaguardare le informazioni sensibili che appaiono nei registri.

  Per ridurre al minimo il sovraccarico associato al proxy, RDS Proxy disattiva automaticamente questa impostazione 24 ore dopo l'attivazione. Abilitare temporaneamente la risoluzione di un problema specifico.

5. Scegli Create Proxy (Crea Proxy).

AWS CLI

Per creare un proxy utilizzando il AWS CLI, chiamate il create-db-proxycomando con i seguenti parametri obbligatori:

- `--db-proxy-name`
- `--engine-family`
- `--role-arn`
- `--auth`
- `--vpc-subnet-ids`

Il valore `--engine-family` prevede la distinzione tra lettere maiuscole e minuscole.

Example

Per LinuxmacOS, oUnix:

```
aws rds create-db-proxy \
    --db-proxy-name proxy_name \
    --engine-family { MYSQL | POSTGRESQL | SQLSERVER } \
    --auth ProxyAuthenticationConfig_JSON_string \
    --role-arn iam_role \
    --vpc-subnet-ids space_separated_list \
    [--vpc-security-group-ids space_separated_list] \
    [--require-tls | --no-require-tls] \
    [--idle-client-timeout value] \
    [--debug-logging | --no-debug-logging] \
    [--tags comma_separated_list]
```

Per Windows:

```
aws rds create-db-proxy ^
    --db-proxy-name proxy_name ^
    --engine-family { MYSQL | POSTGRESQL | SQLSERVER } ^
    --auth ProxyAuthenticationConfig_JSON_string ^
    --role-arn iam_role ^
    --vpc-subnet-ids space_separated_list ^
    [--vpc-security-group-ids space_separated_list] ^
    [--require-tls | --no-require-tls] ^
    [--idle-client-timeout value] ^
    [--debug-logging | --no-debug-logging] ^
    [--tags comma_separated_list]
```

Di seguito è riportato un esempio di valore JSON per l'opzione `--auth`. Questo esempio applica un tipo di autenticazione client diverso a ciascun segreto.

```
[
  {
    "Description": "proxy description 1",
    "AuthScheme": "SECRETS",
    "SecretArn": "arn:aws:secretsmanager:us-
west-2:123456789123:secret/1234abcd-12ab-34cd-56ef-1234567890ab",
    "IAMAuth": "DISABLED",
    "ClientPasswordAuthType": "POSTGRES_SCRAM_SHA_256"
  },

  {
    "Description": "proxy description 2",
    "AuthScheme": "SECRETS",
    "SecretArn": "arn:aws:secretsmanager:us-
west-2:111122223333:seret/1234abcd-12ab-34cd-56ef-1234567890cd",
    "IAMAuth": "DISABLED",
    "ClientPasswordAuthType": "POSTGRES_MD5"

  },

  {
    "Description": "proxy description 3",
    "AuthScheme": "SECRETS",
    "SecretArn": "arn:aws:secretsmanager:us-
west-2:111122221111:secret/1234abcd-12ab-34cd-56ef-1234567890ef",
    "IAMAuth": "REQUIRED"
  }

]
```

Tip

Se non conosci già gli ID delle sottoreti da utilizzare per il parametro `--vpc-subnet-ids`, consulta Configurazione dei prerequisiti di rete per esempi su come trovarli.

Note

Il gruppo di protezione deve consentire l'accesso al database a cui si connette il proxy. Lo stesso gruppo di protezione viene utilizzato per l'ingresso dalle applicazioni al proxy e per l'uscita dal proxy al database. Si supponga, ad esempio, di utilizzare lo stesso gruppo di

protezione per il database e il proxy. In questo caso, assicurarsi di specificare che le risorse di tale gruppo di protezione possono comunicare con altre risorse dello stesso gruppo di protezione.
Quando si utilizza un VPC condiviso, non è possibile utilizzare il gruppo di sicurezza predefinito per il VPC o uno appartenente a un altro account. Scegli un gruppo di sicurezza appartenente all'account. Se non esiste, creane uno. Per ulteriori informazioni su questa limitazione, consulta Utilizzo di VPC condivisi.

Per creare le associazioni corrette per il proxy, si usa anche il register-db-proxy-targetscomando. Specificare il tipo di gruppo di destinazione `default` RDS Proxy crea automaticamente un gruppo di destinazione con questo nome quando si crea ogni proxy.

```
aws rds register-db-proxy-targets
    --db-proxy-name value
    [--target-group-name target_group_name]
    [--db-instance-identifiers space_separated_list]  # rds db instances, or
    [--db-cluster-identifiers cluster_id]        # rds db cluster (all instances)
```

API RDS

Per creare un proxy RDS, chiamare l'operazione Amazon RDS API CreateDBProxy. Si passa un parametro con la struttura AuthConfigdei dati.

RDS Proxy crea automaticamente un gruppo di destinazione denominato `default` quando si crea ogni proxy. È possibile associare un cluster . ProxyTargets

Visualizzazione di un RDS Proxy

Dopo aver creato uno o più proxy RDS, puoi visualizzarli. In questo modo puoi esaminare i dettagli di configurazione e scegliere quali modificare, eliminare e così via.

Affinché le applicazioni di database utilizzino un proxy, è necessario fornire l'endpoint proxy nella stringa di connessione.

AWS Management Console

Per visualizzare il proxy

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.

2. Nell'angolo in alto a destra di AWS Management Console, scegli la AWS regione in cui hai creato il proxy RDS.
3. Nel riquadro di navigazione scegli Proxies (Proxy).
4. Scegli il nome di un proxy RDS per visualizzarne i dettagli.
5. Nella pagina dei dettagli, la sezione Target groups mostra come il proxy è associato a un'istanza RDS DB cluster DB specifica. Puoi seguire il collegamento alla pagina di default del gruppo di destinazione per visualizzare ulteriori dettagli sull'associazione tra il proxy e il database. In questa pagina vengono visualizzate le impostazioni specificate durante la creazione del proxy. Includono la percentuale massima di connessione, il timeout del prestito di connessione, la famiglia di motori e i filtri di associazione delle sessioni.

CLI

Per visualizzare il proxy utilizzando la CLI, utilizzare il describe-db-proxiescomando. Per impostazione predefinita, mostra tutti i proxy di proprietà del tuo account. AWS Per visualizzare i dettagli di un singolo proxy, specificarne il nome con il parametro `--db-proxy-name`.

```
aws rds describe-db-proxies [--db-proxy-name proxy_name]
```

Per visualizzare le altre informazioni associate al proxy, utilizza questi comandi:

```
aws rds describe-db-proxy-target-groups  --db-proxy-name proxy_name

aws rds describe-db-proxy-targets --db-proxy-name proxy_name
```

Utilizza la seguente sequenza di comandi per visualizzare ulteriori dettagli sugli elementi associati al proxy:

1. Per ottenere un elenco di proxy, esegui. describe-db-proxies
2. Per mostrare i parametri di connessione, come la percentuale massima di connessioni che il proxy può utilizzare, describe-db-proxy-target esegui -groups. `--db-proxy-name` Utilizza il nome del proxy come valore del parametro.
3. Per visualizzare i dettagli del cluster associato al gruppo di destinazione restituito, esegui. describe-db-proxy-targets

API RDS

Per visualizzare i proxy utilizzando l'API RDS, utilizza l'operazione DescribedBProxies . Restituisce valori del tipo di dati DbProxy .

Per visualizzare i dettagli delle impostazioni di connessione per il proxy, utilizza gli identificatori proxy di questo valore restituito con l'operazione DescribeDB. ProxyTargetGroups Restituisce valori del tipo di dati DB. ProxyTargetGroup

Per visualizzare l'istanza RDS o il cluster Aurora DB associato al proxy, utilizzare l'ProxyTargetsoperazione DescribeDB. Restituisce valori del tipo di dati DB. ProxyTarget

Connessione a un database tramite RDS Proxy

Il modo per connettersi a un'istanza DB RDS tramite un proxy o collegandosi al database è generalmente lo stesso. Per ulteriori informazioni, consulta Panoramica degli endpoint proxy.

Argomenti

- Connessione a un proxy utilizzando l'autenticazione nativa
- Connessione a un proxy mediante autenticazione IAM
- Considerazioni per la connessione a un proxy con Microsoft SQL Server
- Considerazioni per la connessione a un proxy con PostgreSQL

Connessione a un proxy utilizzando l'autenticazione nativa

Utilizza i seguenti passaggi per connetterti a un proxy utilizzando l'autenticazione nativa:

1. Trova l'endpoint del proxy. In AWS Management Console, puoi trovare l'endpoint nella pagina dei dettagli del proxy corrispondente. Con AWS CLI, è possibile utilizzare il describe-db-proxiescomando. L'esempio seguente mostra come.

```
# Add --output text to get output as a simple tab-separated list.
$ aws rds describe-db-proxies --query '*[*].
{DBProxyName:DBProxyName,Endpoint:Endpoint}'
[
    [
        {
            "Endpoint": "the-proxy.proxy-demo.us-east-1.rds.amazonaws.com",
            "DBProxyName": "the-proxy"
```

```
        },
        {
            "Endpoint": "the-proxy-other-secret.proxy-demo.us-
east-1.rds.amazonaws.com",
            "DBProxyName": "the-proxy-other-secret"
        },
        {
            "Endpoint": "the-proxy-rds-secret.proxy-demo.us-
east-1.rds.amazonaws.com",
            "DBProxyName": "the-proxy-rds-secret"
        },
        {
            "Endpoint": "the-proxy-t3.proxy-demo.us-east-1.rds.amazonaws.com",
            "DBProxyName": "the-proxy-t3"
        }
    ]
]
```

2. Specificate l'endpoint come parametro host nella stringa di connessione per l'applicazione client. Ad esempio, specificare l'endpoint proxy come valore per l'opzione `mysql -h` o l'opzione `psql -h`.
3. Fornisci lo stesso nome utente e la stessa password del database come fai normalmente.

Connessione a un proxy mediante autenticazione IAM

Quando si utilizza l'autenticazione IAM con RDS Proxy, impostare gli utenti del database per l'autenticazione con nomi utente e password regolari. L'autenticazione IAM si applica al recupero RDS Proxy delle credenziali del nome utente e della password da Secrets Manager. La connessione da RDS Proxy al database sottostante non passa tramite IAM.

Per connetterti al proxy RDS utilizzando l'autenticazione IAM, utilizza la stessa procedura di connessione generale utilizzata per l'autenticazione IAM con un cluster DB di istanze RDS DB. Per informazioni generali sull'utilizzo di IAM, consulta Sicurezza in Amazon RDS.

Le principali differenze nell'utilizzo di IAM per RDS Proxy includono le seguenti:

- Non si configura ogni singolo utente del database con un plugin di autorizzazione. Gli utenti del database hanno ancora nomi utente e password regolari all'interno del database. Si impostano i segreti Secrets Manager contenenti questi nomi utente e password e si autorizza RDS Proxy a recuperare le credenziali da Secrets Manager.

L'autenticazione IAM si applica alla connessione tra il programma client e il proxy. Il proxy esegue quindi l'autenticazione nel database utilizzando il nome utente e le credenziali della password recuperate da Secrets Manager.

- Invece dell'istanza, del cluster o dell'endpoint dell'istanza di lettura, specifica l'endpoint del proxy. Per informazioni dettagliate sull'endpoint proxy, vedere Connessione all'istanza tramite l'autenticazione IAM.
- Nel caso di autenticazione diretta di database IAM, scegli selettivamente gli utenti del database e configurali per essere identificati con uno speciale plug-in di autenticazione. È quindi possibile connettersi a tali utenti utilizzando l'autenticazione IAM.

  Nel caso d'uso del proxy, è necessario fornire al proxy segreti che contengono nome utente e password di alcuni utenti (autenticazione nativa). Quindi ci si connette al proxy utilizzando l'autenticazione IAM. Qui, si esegue questa operazione generando un token di autenticazione con l'endpoint proxy, non l'endpoint del database. Si utilizza anche un nome utente che corrisponde a uno dei nomi utente per i segreti forniti.
- Quando ci si connette a un proxy utilizzando l'autenticazione IAM è necessario utilizzare Transport Layer Security (TLS)/Secure Sockets Layer (SSL).

È possibile concedere a un utente specifico l'accesso al proxy modificando la policy IAM. Di seguito è riportato un esempio.

```
"Resource": "arn:aws:rds-db:us-east-2:1234567890:dbuser:prx-ABCDEFGHIJKL01234/db_user"
```

Considerazioni per la connessione a un proxy con Microsoft SQL Server

Per connettere un proxy con l'autenticazione IAM, non utilizzare il campo della password. Fornisci invece la proprietà token appropriata per ogni tipo di driver di database nel campo del token. Ad esempio, utilizza la proprietà `accessToken` per JDBC, la proprietà `sql_copt_ss_access_token` per ODBC Oppure usa la `AccessToken` proprietà per il driver.NET. SqlClient Non puoi usare l'autenticazione IAM con i client che non supportano le proprietà dei token.

In alcune condizioni, il proxy non può condividere una connessione al database e associa la connessione dall'applicazione client al proxy a una connessione al database dedicata. Per ulteriori informazioni su queste condizioni, consulta Evitare il pinning.

Considerazioni per la connessione a un proxy con PostgreSQL

Per PostgreSQL, quando un client avvia una connessione a un database PostgreSQL, invia un messaggio di avvio che include coppie di stringhe di nome parametro e valore. Per i dettagli, vedere i `StartupMessage` in Formati dei messaggi PostgreSQL nella documentazione di PostgreSQL.

Quando si effettua la connessione tramite un proxy RDS, il messaggio di avvio può includere i seguenti parametri attualmente riconosciuti:

- `user`
- `database`

Il messaggio di avvio può includere anche i seguenti parametri di runtime aggiuntivi:

- `application_name`
- `client_encoding`
- `DateStyle`
- `TimeZone`
- `extra_float_digits`
- `search_path`

Per ulteriori informazioni sulla messaggistica PostgreSQL, vedere Frontend/Backend Protocol nella documentazione di PostgreSQL.

Per PostgreSQL, se usi JDBC, ti consigliamo quanto segue per evitare il pinning:

- Impostare il parametro `assumeMinServerVersion` di connessione JDBC su almeno `9.0` per evitare il pinning. Ciò impedisce al driver JDBC di eseguire un ulteriore round trip durante l'avvio della connessione quando è in esecuzione. `SET extra_float_digits = 3`
- Impostare il parametro di connessione JDBC `ApplicationName` su *`any/your-application-name`* per evitare il pinning. In questo modo si impedisce al driver JDBC di eseguire un ulteriore round trip durante l'avvio della connessione quando viene eseguito `SET application_name = "PostgreSQL JDBC Driver"`. Si noti che il parametro JDBC è `ApplicationName` ma il parametro `StartupMessage` PostgreSQL è `application_name`.

Per ulteriori informazioni, consulta Evitare il pinning. Per ulteriori informazioni sulla connessione tramite JDBC, vedere Connessione al database nella documentazione di PostgreSQL.

Gestire un RDS Proxy

Questa sezione fornisce informazioni su come gestire il funzionamento e la configurazione del proxy RDS. Queste procedure consentono all'applicazione di utilizzare in modo più efficiente le connessioni al database e di ottenere il massimo riutilizzo della connessione. Più sfrutti il riutilizzo della connessione, maggiore sarà il risparmio in termini di sovraccarico di CPU e memoria. Questo a sua volta riduce la latenza per l'applicazione e consente al database di dedicare più risorse all'elaborazione delle richieste dell'applicazione.

Argomenti

Modifica di un RDS Proxy

Puoi modificare specifiche impostazioni associate a un proxy dopo aver creato il proxy. Esegui questa operazione modificando il proxy stesso, il suo gruppo di destinazione associato o entrambi. Ogni proxy ha un gruppo di destinazione associato.

AWS Management Console

> **Important**
>
> I valori nei campi Client authentication type (Tipo di autenticazione client) e IAM authentication (autenticazione IAM) si applicano a tutti i segreti di Secrets Manager associati al proxy. Per specificare valori diversi per ogni segreto, modifica il proxy utilizzando invece l'API AWS CLI o.

Per modificare le impostazioni di un proxy

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione scegli Proxies (Proxy).
3. Nell'elenco dei proxy, scegli il proxy di cui desideri modificare le impostazioni o passa alla relativa pagina dei dettagli.
4. Per Actions (Operazioni), scegliere Modify (Modifica).
5. Inserisci o scegli le proprietà da modificare. È possibile modificare le seguenti:
   - L'identificatore Proxy: rinominare il proxy immettendo un nuovo identificatore.
   - Timeout della connessione client per inattività: immettere un periodo di tempo per il timeout della connessione client inattiva.
   - Ruolo IAM: modificare il ruolo IAM utilizzato per recuperare i segreti da Secrets Manager.
   - Segreti di Secrets Manager: aggiungere o rimuovere segreti di Secrets Manager. Questi segreti corrispondono a nomi utente e password del database.
   - Tipo di autenticazione client: (solo PostgreSQL) cambia il tipo di autenticazione per le connessioni client al proxy.
   - Autenticazione IAM: richiedi o disabilita l'autenticazione IAM per le connessioni al proxy.
   - Richiedi Transport Layer Security: attivare o disattivare il requisito per Transport Layer Security (TLS).
   - Gruppo di sicurezza VPC: aggiungere o rimuovere gruppi di sicurezza VPC da utilizzare per il proxy.
   - Abilitazione della registrazione avanzata: abilitare o disabilitare la registrazione avanzata.
6. Scegliere Modify (Modifica).

Se non sono state trovate le impostazioni elencate che si desidera modificare, attenersi alla procedura seguente per aggiornare il gruppo di destinazione per il proxy. Il gruppo di destinazione associato a un proxy controlla le impostazioni relative alle connessioni del database fisico. Ogni proxy ha un gruppo di destinazione associato denominato `default`, che viene creato automaticamente insieme al proxy.

Puoi modificare il gruppo di destinazione solo dalla pagina dei dettagli del proxy, non dall'elenco nella pagina Proxies (Proxy) .

Per modificare le impostazioni di un gruppo di destinazione proxy

1. Dalla pagina Proxy, passa alla pagina dei dettagli di un proxy.
2. Per Target groups (Gruppi di destinazione), scegli il collegamento di `default`. Attualmente, tutti i proxy hanno un singolo gruppo di destinazione denominato `default`.
3. Nella pagina dei dettagli del gruppo di destinazione di default, scegli Modify (Modifica).
4. Scegli nuove impostazioni per le proprietà che è possibile modificare:

   - Database: puoi scegliere un diverso cluster o una diversa istanza database RDS.
   - Connessioni massime del pool di connessioni: puoi modificare la percentuale delle connessioni massime disponibili che il proxy può utilizzare.
   - Filtri di pinning della sessione: (opzionale) scegliere un filtro di pinning della sessione. Ciò elude le misure di sicurezza predefinite per il multiplexing delle connessioni al database tra connessioni client. Attualmente, l'impostazione non è supportata per PostgreSQL. L'unica scelta è. `EXCLUDE_VARIABLE_SETS`

     L'attivazione di questa impostazione può far sì che le variabili di sessione di una connessione influiscano sulle altre connessioni. Ciò può causare errori o problemi di correttezza se le query dipendono dai valori delle variabili di sessione impostati al di fuori della transazione corrente. Valuta l'utilizzo di questa opzione dopo aver verificato che sia sicuro per le applicazioni condividere le connessioni al database tra connessioni client.

     I seguenti modelli possono essere considerati sicuri:

     - Istruzioni `SET` in cui non viene apportata alcuna modifica al valore della variabile di sessione effettiva, ovvero non viene apportata alcuna modifica alla variabile di sessione.
     - Modifichi il valore della variabile di sessione ed esegui un'istruzione nella stessa transazione.

     Per ulteriori informazioni, consulta Evitare il pinning.
   - Timeout del prestito di connessione: puoi regolare l'intervallo di timeout del prestito di connessione. Questa impostazione si applica quando il numero massimo di connessioni è già in uso per il proxy. In questi casi, è possibile specificare per quanto tempo il proxy attende che una connessione diventi disponibile prima di restituire un errore di timeout.
   - Query di inizializzazione: (opzionale) aggiungere una query di inizializzazione o modificare quella corrente. Puoi specificare una o più istruzioni SQL per l'esecuzione del proxy all'apertura di ogni nuova connessione al database. L'impostazione è in genere utilizzata

con le istruzioni `SET` per assicurarsi che ogni connessione abbia impostazioni identiche, ad esempio fuso orario e set di caratteri. Per più istruzioni, utilizzare il punto e virgola come separatore. È inoltre possibile includere più variabili in una singola istruzione `SET`, ad esempio `SET x=1, y=2`.

Alcune proprietà, ad esempio l'identificatore del gruppo di destinazione e il motore del database, sono fisse.

5. Scegli Modify target group (Modifica gruppo di destinazione).

AWS CLI

Per modificare un proxy utilizzando il AWS CLI, usa i comandi modify-db-proxymodify-db-proxy-target-group deregister-db-proxy-targets, e register-db-proxy-targets.

Con il comando `modify-db-proxy`, è possibile modificare proprietà come le seguenti:

- L'insieme di segreti Secrets Manager usati dal proxy.
- Indica se TLS è necessario.
- Il timeout del client inattivo.
- Indica se registrare ulteriori informazioni dalle istruzioni SQL per il debug.
- Il ruolo IAM utilizzato per recuperare i segreti Secrets Manager.
- I gruppi di sicurezza utilizzati dal proxy.

Nell'esempio seguente viene illustrato come rinominare un proxy esistente.

```
aws rds modify-db-proxy --db-proxy-name the-proxy --new-db-proxy-name the_new_name
```

Con il comando `modify-db-proxy-target-group`, puoi modificare le impostazioni relative alla connessione o rinominare il gruppo di destinazione. Attualmente, tutti i proxy hanno un singolo gruppo di destinazione denominato `default`. Quando lavori con questo gruppo di destinazione, devi specificare il nome del proxy e `default` per il nome del gruppo di destinazione.

Nell'esempio seguente viene illustrato come controllare prima l'impostazione `MaxIdleConnectionsPercent` per un proxy e quindi modificarla utilizzando il gruppo di destinazione.

```
aws rds describe-db-proxy-target-groups --db-proxy-name the-proxy

{
    "TargetGroups": [
        {
            "Status": "available",
            "UpdatedDate": "2019-11-30T16:49:30.342Z",
            "ConnectionPoolConfig": {
                "MaxIdleConnectionsPercent": 50,
                "ConnectionBorrowTimeout": 120,
                "MaxConnectionsPercent": 100,
                "SessionPinningFilters": []
            },
            "TargetGroupName": "default",
            "CreatedDate": "2019-11-30T16:49:27.940Z",
            "DBProxyName": "the-proxy",
            "IsDefault": true
        }
    ]
}

aws rds modify-db-proxy-target-group --db-proxy-name the-proxy --target-group-name
 default --connection-pool-config '
{ "MaxIdleConnectionsPercent": 75 }'

{
    "DBProxyTargetGroup": {
        "Status": "available",
        "UpdatedDate": "2019-12-02T04:09:50.420Z",
        "ConnectionPoolConfig": {
            "MaxIdleConnectionsPercent": 75,
            "ConnectionBorrowTimeout": 120,
            "MaxConnectionsPercent": 100,
            "SessionPinningFilters": []
        },
        "TargetGroupName": "default",
        "CreatedDate": "2019-11-30T16:49:27.940Z",
        "DBProxyName": "the-proxy",
        "IsDefault": true
    }
}
```

Con i comandi `deregister-db-proxy-targets` e `register-db-proxy-targets`, puoi modificare le istanze database RDS a cui il proxy è associato tramite il relativo gruppo di destinazione. Attualmente, ogni proxy può connettersi a un cluster DB di istanze RDS DB. Il gruppo target tiene traccia dei dettagli di connessione per tutte le istanze DB RDS in una configurazione Multi-AZ, cluster Aurora.

L'esempio seguente inizia con un proxy associato a un cluster Aurora MySQL denominato `cluster-56-2020-02-25-1399`. Nell'esempio viene illustrato come modificare il proxy in modo che possa connettersi a un cluster diverso denominato `provisioned-cluster`.

Quando lavori con un'istanza database RDS, puoi specificare l'opzione `--db-instance-identifier`.

L'esempio seguente modifica un proxy Aurora MySQL. Un proxy Aurora PostgreSQL ha la porta 5432.

```
aws rds describe-db-proxy-targets --db-proxy-name the-proxy

{
    "Targets": [
        {
            "Endpoint": "instance-9814.demo.us-east-1.rds.amazonaws.com",
            "Type": "RDS_INSTANCE",
            "Port": 3306,
            "RdsResourceId": "instance-9814"
        },
        {
            "Endpoint": "instance-8898.demo.us-east-1.rds.amazonaws.com",
            "Type": "RDS_INSTANCE",
            "Port": 3306,
            "RdsResourceId": "instance-8898"
        },
        {
            "Endpoint": "instance-1018.demo.us-east-1.rds.amazonaws.com",
            "Type": "RDS_INSTANCE",
            "Port": 3306,
            "RdsResourceId": "instance-1018"
        },
        {
            "Type": "TRACKED_CLUSTER",
            "Port": 0,
            "RdsResourceId": "cluster-56-2020-02-25-1399"
```

```
        },
        {
            "Endpoint": "instance-4330.demo.us-east-1.rds.amazonaws.com",
            "Type": "RDS_INSTANCE",
            "Port": 3306,
            "RdsResourceId": "instance-4330"
        }
    ]
}

aws rds deregister-db-proxy-targets --db-proxy-name the-proxy --db-cluster-identifier
 cluster-56-2020-02-25-1399

aws rds describe-db-proxy-targets --db-proxy-name the-proxy

{
    "Targets": []
}

aws rds register-db-proxy-targets --db-proxy-name the-proxy --db-cluster-identifier
 provisioned-cluster

{
    "DBProxyTargets": [
        {
            "Type": "TRACKED_CLUSTER",
            "Port": 0,
            "RdsResourceId": "provisioned-cluster"
        },
        {
            "Endpoint": "gkldje.demo.us-east-1.rds.amazonaws.com",
            "Type": "RDS_INSTANCE",
            "Port": 3306,
            "RdsResourceId": "gkldje"
        },
        {
            "Endpoint": "provisioned-1.demo.us-east-1.rds.amazonaws.com",
            "Type": "RDS_INSTANCE",
            "Port": 3306,
            "RdsResourceId": "provisioned-1"
        }
    ]
}
```

API RDS

Per modificare un proxy utilizzando l'API RDS, si utilizzano le operazioni ModifyDBProxy, ModifyDB, ProxyTargetGroup deregisterDB e registerDB. ProxyTargets ProxyTargets

Con `ModifyDBProxy`, è possibile modificare proprietà come le seguenti:

- L'insieme di segreti Secrets Manager usati dal proxy.
- Indica se TLS è necessario.
- Il timeout del client inattivo.
- Indica se registrare ulteriori informazioni dalle istruzioni SQL per il debug.
- Il ruolo IAM utilizzato per recuperare i segreti Secrets Manager.
- I gruppi di sicurezza utilizzati dal proxy.

Con `ModifyDBProxyTargetGroup`, puoi modificare le impostazioni relative alla connessione o rinominare il gruppo di destinazione. Attualmente, tutti i proxy hanno un singolo gruppo di destinazione denominato `default`. Quando lavori con questo gruppo di destinazione, devi specificare il nome del proxy e `default` per il nome del gruppo di destinazione.

Con `DeregisterDBProxyTargets` e`RegisterDBProxyTargets`, si modifica l'istanza RDS DB a cui il cluster è associato il proxy tramite il relativo gruppo di destinazione. Attualmente, ogni proxy può connettersi a un'istanza database RDS . Il gruppo di destinazione tiene traccia dei dettagli di connessione per le istanze database RDS in una configurazione Multi-AZ .

Aggiunta di un nuovo utente di database

In alcuni casi, puoi aggiungere un nuovo utente di database a un cluster o un'istanza database RDS associato a un proxy. In tal caso, aggiungi o riutilizza un segreto Secrets Manager per archiviare le credenziali per tale utente. Per fare ciò, scegli una delle seguenti opzioni:

1. Crea un nuovo segreto Secrets Manager, utilizzando la procedura descritta in Configurazione delle credenziali del database in AWS Secrets Manager.
2. Aggiornare il ruolo IAM per consentire a RDS Proxy l'accesso al nuovo segreto Secrets Manager. A tale scopo, aggiornare la sezione Risorse della policy di ruolo IAM.
3. Modifica il proxy RDS per aggiungere il nuovo segreto di Secrets Manager nell'area Segreti Secrets Manager.

4. Se il nuovo utente sostituisce un utente esistente, aggiorna le credenziali memorizzate nel segreto Secrets Manager del proxy per l'utente esistente.

Aggiungere un nuovo utente del database a un database PostgreSQL

Quando aggiungi un nuovo utente al tuo database PostgreSQL, se hai eseguito il seguente comando:

```
REVOKE CONNECT ON DATABASE postgres FROM PUBLIC;
```

All'utente `rdsproxyadmin` concedere il privilegio `CONNECT` in modo che possa monitorare le connessioni sul database di destinazione.

```
GRANT CONNECT ON DATABASE postgres TO rdsproxyadmin;
```

È anche possibile consentire ad altri utenti del database di destinazione di eseguire controlli dell'integrità passando `rdsproxyadmin` all'utente del database nel comando precedente.

Modifica della password per un utente di database

In alcuni casi, puoi modificare la password per un utente di database in un'istanza database RDS associata a un proxy. In tal caso, aggiorna il segreto Secrets Manager corrispondente con la nuova password.

Connessioni client e database

Le connessioni dall'applicazione a Server proxy per RDS sono note come connessioni client. Le connessioni da un proxy al database sono le connessioni database. Quando si utilizza Server proxy per RDS, le connessioni client terminano sul proxy mentre le connessioni database vengono gestite all'interno di Server proxy per RDS.

Il pool di connessioni lato applicazione può offrire il vantaggio di ridurre la creazione di connessioni ricorrenti tra l'applicazione e il proxy RDS.

Considerate i seguenti aspetti di configurazione prima di implementare un pool di connessioni lato applicazione:

- Durata massima della connessione client: RDS Proxy impone una durata massima delle connessioni client di 24 ore. Questo valore non è configurabile. Configura il pool con una durata

massima della connessione inferiore a 24 ore per evitare interruzioni impreviste della connessione del client.

- Timeout di inattività della connessione client: RDS Proxy impone un tempo di inattività massimo per le connessioni client. Configura il pool con un valore di timeout di inattività della connessione inferiore all'impostazione di timeout di inattività della connessione client per Server proxy per RDS per evitare interruzioni impreviste della connessione.

Il numero massimo di connessioni client configurate nel pool di connessioni lato applicazione non deve essere limitato all'impostazione max_connections per RDS Proxy.

Il pooling delle connessioni dei client comporta una maggiore durata della connessione del client. Se si verifica il pinning delle connessioni, il pool delle connessioni client può ridurre l'efficienza del multiplexing. Le connessioni client bloccate ma inattive nel pool di connessioni lato applicazione continuano a mantenere una connessione al database e impediscono che la connessione al database venga riutilizzata da altre connessioni client. Esamina i log del proxy per verificare se le connessioni presentano problemi di blocco.

> **Note**
>
> RDS Proxy chiude le connessioni al database all'incirca dopo 24 ore quando non sono più in uso. Il proxy esegue questa operazione indipendentemente dal valore dell'impostazione massima delle connessioni inattive.

Configurazione delle impostazioni di connessione

Per regolare il pooling di connessioni del proxy RDS, è possibile modificare le seguenti impostazioni:

- IdleClientTimeout
- MaxConnectionsPercent
- MaxIdleConnectionsPercent
- ConnectionBorrowTimeout

IdleClientTimeout

È possibile specificare per quanto tempo una connessione client può rimanere inattiva prima che il proxy la chiuda. Il valore predefinito è 1.800 secondi (30 minuti).

Una connessione client è considerata inattiva quando l'applicazione non invia una nuova richiesta entro il tempo specificato dopo il completamento della richiesta precedente. La connessione al database sottostante rimane aperta e viene restituita al pool di connessioni. Pertanto, è disponibile per essere riutilizzata per nuove connessioni client. Se desideri che il proxy rimuova in modo proattivo le connessioni obsolete, riduci il timeout della connessione del client inattivo. Se il carico di lavoro stabilisce connessioni frequenti con il proxy, aumenta il timeout di connessione del client inattivo per risparmiare sui costi di creazione delle connessioni.

Questa impostazione è rappresentata dal campo di timeout della connessione del client Idle nella console RDS e dall'impostazione in and nell'`IdleClientTimeout`API. AWS CLI Per informazioni su come modificare il valore del campo Idle client connection timeout (Timeout di connessione client inattivo) nella console RDS, consulta AWS Management Console. Per informazioni su come modificare il valore dell'`IdleClientTimeout`impostazione, consulta il comando CLI modify-db-proxyo l'operazione API ModifyDBProxy.

MaxConnectionsPercent

Puoi limitare il numero di connessioni che un proxy RDS può stabilire con il database di destinazione. Puoi specificare il limite come percentuale delle connessioni massime disponibili per il tuo database. Questa impostazione è rappresentata dal campo Connection pool di connessioni massime nella console RDS e dall'`MaxConnectionsPercent`impostazione in and nell'API. AWS CLI

Il valore `MaxConnectionsPercent` viene espresso come percentuale dell'impostazione `max_connections` per l'istanza database RDS usata dal gruppo di destinazione. Il proxy non crea tutte queste connessioni in anticipo. Questa impostazione consente al proxy di stabilire queste connessioni quando il carico di lavoro le richiede.

Ad esempio, per una destinazione di database registrata con il parametro `max_connections` impostato su 1000 e il parametro `MaxConnectionsPercent` impostato su 95, RDS Proxy imposta 950 connessioni come limite massimo per le connessioni simultanee al database di destinazione specificato.

Un effetto collaterale comune del raggiungimento del numero massimo di connessioni al database consentite dal carico di lavoro è un aumento della latenza complessiva delle query e un incremento del valore della metrica `DatabaseConnectionsBorrowLatency`. È possibile monitorare le connessioni al database attualmente utilizzate e il totale consentito confrontando le metriche `DatabaseConnections` e `MaxDatabaseConnectionsAllowed`.

Se imposti questo parametri, segui le best practice riportate di seguito:

- Consenti un margine sufficiente per le connessioni per la modifica dello schema del carico di lavoro. Si consiglia di impostare il parametro su un valore superiore di almeno il 30% rispetto all'utilizzo massimo monitorato. Poiché RDS Proxy ridistribuisce le quote di connessione del database su più nodi, le modifiche alla capacità interna potrebbero richiedere un margine di almeno il 30% per connessioni aggiuntive per evitare l'incremento delle latenze di prestito.
- RDS Proxy riserva un certo numero di connessioni per il monitoraggio attivo per supportare il failover rapido, l'instradamento del traffico e le operazioni interne. Il parametro `MaxDatabaseConnectionsAllowed` non include queste connessioni riservate. Rappresenta il numero di connessioni disponibili per servire il carico di lavoro e può essere inferiore al valore derivato dall'impostazione `MaxConnectionsPercent`.

  Valori `MaxConnectionsPercent` minimi consigliati

  - db.t3.small: 30
  - db.t3.medium o superiore: 20

Per informazioni su come modificare il valore del campo Connection pool maximum connections (Numero massimo di connessioni del pool di connessioni) nella console RDS, consulta AWS Management Console. Per informazioni su come modificare il valore dell'`MaxConnectionsPercent`impostazione, consulta il comando CLI modify-db-proxy-target-group o l'operazione API ModifyDB. ProxyTargetGroup

Per informazioni sui limiti di connessione al database, consulta Numero massimo di connessioni di database.

MaxIdleConnectionsPercent

Puoi controllare il numero di connessioni al database inattive che RDS Proxy può mantenere nel pool di connessione. Per impostazione predefinita, RDS Proxy considera inattiva una connessione al database nel suo pool quando non vi è stata alcuna attività sulla connessione per cinque minuti.

Puoi specificare il limite come percentuale delle connessioni massime disponibili per il tuo database. Il valore predefinito è 50% di `MaxConnectionsPercent` e il limite superiore è il valore di `MaxConnectionsPercent`. Con un valore elevato, il proxy lascia aperta un'alta percentuale di connessioni al database inattive. Con un valore basso, il proxy chiude un'alta percentuale di connessioni al database inattive. Se i carichi di lavoro sono imprevedibili, valuta la possibilità di impostare un valore elevato per `MaxIdleConnectionsPercent`. In tal modo Server proxy per RDS può soddisfare i picchi di attività senza aprire molte nuove connessioni al database.

Questa impostazione è rappresentata dall'`MaxIdleConnectionsPercent`impostazione di `DBProxyTargetGroup` in AWS CLI e nell'API. Per informazioni su come modificare il valore dell'`MaxIdleConnectionsPercent`impostazione, consulta il comando CLI modify-db-proxy-target-group o l'operazione API ModifyDB. ProxyTargetGroup

Per informazioni sui limiti di connessione al database, consulta Numero massimo di connessioni di database.

ConnectionBorrowTimeout

Puoi specificare per quanto tempo RDS Proxy attende che una connessione di database nel pool di connessione diventi disponibile per l'uso prima di restituire un errore di timeout. Il valore predefinito è 120 secondi. Questa impostazione si applica quando il numero di connessioni è pari al massimo e quindi non sono disponibili connessioni nel pool di connessioni. Si applica anche quando non è disponibile un'istanza di database appropriata per gestire la richiesta, ad esempio quando è in corso un'operazione di failover. Utilizzando questa impostazione, è possibile impostare il periodo di attesa migliore per l'applicazione senza modificare il timeout della query nel codice dell'applicazione.

Questa impostazione è rappresentata dal campo Connection borrow timeout nella console RDS o dall'`ConnectionBorrowTimeout`impostazione di `DBProxyTargetGroup` nell'API o. AWS CLI Per informazioni su come modificare il valore del campo Connection borrow timeout (Timeout del prestito della connessione) nella console RDS, consulta AWS Management Console. Per informazioni su come modificare il valore dell'`ConnectionBorrowTimeout`impostazione, consulta il comando CLI modify-db-proxy-target-group o l'operazione API ModifyDB. ProxyTargetGroup

Evitare il pinning

Il multiplexing è più efficiente quando le richieste del database non si basano su informazioni di stato provenienti da richieste precedenti. In tal caso, RDS Proxy può riutilizzare una connessione alla conclusione di ogni transazione. Esempi di tali informazioni sullo stato includono la maggior parte delle variabili e dei parametri di configurazione che puoi modificare attraverso le istruzioni `SET` o `SELECT`. Le transazioni SQL su una connessione client possono eseguire il multiplex tra le connessioni di database sottostanti per impostazione predefinita.

Le connessioni al proxy possono entrare in uno stato noto come pinning. Quando una connessione viene bloccata, ogni transazione successiva utilizza la stessa connessione al database sottostante fino al termine della sessione. Altre connessioni client, inoltre, non possono riutilizzare tale connessione al database fino al termine della sessione. La sessione termina quando viene interrotta la connessione client.

RDS Proxy collega automaticamente una connessione client a una specifica connessione DB quando rileva una modifica dello stato della sessione che non è appropriata per altre sessioni. Il pinning riduce l'efficacia del riutilizzo della connessione. Se tutte o quasi tutte le connessioni riscontrano il pinning, potresti modificare il codice dell'applicazione o il carico di lavoro per ridurre le condizioni che causano il blocco.

Ad esempio, l'applicazione modifica una variabile di sessione o un parametro di configurazione. In questo caso, le istruzioni successive possono basarsi sulla nuova variabile o sul nuovo parametro per essere effettive. Pertanto, quando RDS Proxy elabora le richieste di modifica delle variabili di sessione o delle impostazioni di configurazione, il medesimo effettua il pinning di tale sessione alla connessione DB. In questo modo, lo stato della sessione rimane attivo per tutte le transazioni successive nella stessa sessione.

Per alcuni motori di database, questa regola non si applica a tutti i parametri che puoi impostare. RDS Proxy tiene traccia di determinate istruzioni e variabili. Pertanto, RDS Proxy non blocca la sessione quando la modificate. In tal caso, RDS Proxy riutilizza la connessione solo per altre sessioni con gli stessi valori per tali impostazioni. Per dettagli sulle istruzioni che Server proxy per RDS può tracciare per un motore di database, consulta quanto segue:

- Istruzioni tracciate da Server proxy per RDS per database RDS per SQL Server
- Istruzioni tracciate da Server proxy per RDS per database RDS per MariaDB e RDS per MySQL

Istruzioni tracciate da Server proxy per RDS per database RDS per SQL Server

Di seguito sono riportate le istruzioni SQL Server di cui Server proxy per RDS tiene traccia:

- `USE`
- `SET ANSI_NULLS`
- `SET ANSI_PADDING`
- `SET ANSI_WARNINGS`
- `SET ARITHABORT`
- `SET CONCAT_NULL_YIELDS_NULL`
- `SET CURSOR_CLOSE_ON_COMMIT`
- `SET DATEFIRST`
- `SET DATEFORMAT`

- SET LANGUAGE
- SET LOCK_TIMEOUT
- SET NUMERIC_ROUNDABORT
- SET QUOTED_IDENTIFIER
- SET TEXTSIZE
- SET TRANSACTION ISOLATION LEVEL

Istruzioni tracciate da Server proxy per RDS per database RDS per MariaDB e RDS per MySQL

Di seguito sono riportate le istruzioni MariadB e MySQL che RDS Proxy tiene traccia:

- DROP DATABASE
- DROP SCHEMA
- USE

Di seguito sono riportate le variabili MySQL e MariaDB di cui il proxy RDS tiene traccia:

- AUTOCOMMIT
- AUTO_INCREMENT_INCREMENT
- CHARACTER SET (or CHAR SET)
- CHARACTER_SET_CLIENT
- CHARACTER_SET_DATABASE
- CHARACTER_SET_FILESYSTEM
- CHARACTER_SET_CONNECTION
- CHARACTER_SET_RESULTS
- CHARACTER_SET_SERVER
- COLLATION_CONNECTION
- COLLATION_DATABASE
- COLLATION_SERVER
- INTERACTIVE_TIMEOUT

- `NAMES`
- `NET_WRITE_TIMEOUT`
- `QUERY_CACHE_TYPE`
- `SESSION_TRACK_SCHEMA`
- `SQL_MODE`
- `TIME_ZONE`
- `TRANSACTION_ISOLATION (or TX_ISOLATION)`
- `TRANSACTION_READ_ONLY (or TX_READ_ONLY)`
- `WAIT_TIMEOUT`

Riduzione dell'associazione

L'ottimizzazione delle prestazioni di RDS Proxy comporta il tentativo di massimizzare il riutilizzo della connessione a livello di transazione (multiplexing) riducendo al minimo il pinning.

È possibile ridurre l'associazione effettuando le seguenti operazioni:

- Evitare richieste di database non necessarie che potrebbero causare il pinning.
- Impostare le variabili e le impostazioni di configurazione in modo coerente su tutte le connessioni. In questo modo, le sessioni successive hanno maggiori probabilità di riutilizzare le connessioni con quelle specifiche impostazioni.

  Tuttavia, per l'impostazione di PostgreSQL una variabile porterà al pinning della sessione.
- Per un database della famiglia di motori MySQL, applica un filtro di pinning della sessione al proxy. Puoi esentare determinati tipi di operazioni dal pinning della sessione se sai che tale operazione non influisce sul corretto funzionamento dell'applicazione.
- Scopri con quale frequenza si verifica il pinning monitorando la CloudWatch metrica `DatabaseConnectionsCurrentlySessionPinned` di Amazon. Per informazioni su questa e altre CloudWatch metriche, consulta. Monitoraggio dei parametri del proxy RDS con Amazon CloudWatch
- Se utilizzi istruzioni `SET` per eseguire l'inizializzazione identica per ogni connessione client, puoi farlo pur mantenendo il multiplexing a livello di transazione. In questo caso, sposta le istruzioni che impostano lo stato della sessione iniziale nella query di inizializzazione utilizzata da un proxy. Questa proprietà è una stringa contenente una o più istruzioni SQL, separate da punto e virgola.

Ad esempio, puoi definire una query di inizializzazione per un proxy che imposta determinati parametri di configurazione. Quindi, RDS Proxy applica tali impostazioni ogni volta che imposta una nuova connessione per tale proxy. Puoi rimuovere le istruzioni `SET` corrispondenti dal codice dell'applicazione, in modo che non interferiscano con il multiplexing a livello di transazione.

Per visualizzare i parametri sulla frequenza con cui si verifica il pinning in relazione a un proxy, consulta Monitoraggio dei parametri del proxy RDS con Amazon CloudWatch.

Condizioni che causano il pinning per tutte le famiglie di motori

Il proxy effettua il pinning della sessione alla connessione corrente nelle seguenti situazioni in cui il multiplexing potrebbe causare un comportamento imprevisto:

- Qualsiasi istruzione con una dimensione del testo maggiore di 16 KB fa sì che il proxy effettui il pinning della sessione.

Condizioni che causano l'associazione per RDS per Microsoft SQL Server

Per RDS per SQL Server, anche le seguenti interazioni causano l'associazione:

- L'utilizzo di più set di risultati attivi (MARS). Per informazioni su MARS, consulta la documentazione di SQL Server.
- L'utilizzo della comunicazione con Distributed Transaction Coordinator (DTC).
- La creazione di tabelle temporanee, transazioni, cursori o istruzioni preparate.
- L'utilizzo delle seguenti istruzioni `SET`:
  - `SET ANSI_DEFAULTS`
  - `SET ANSI_NULL_DFLT`
  - `SET ARITHIGNORE`
  - `SET DEADLOCK_PRIORITY`
  - `SET FIPS_FLAGGER`
  - `SET FMTONLY`
  - `SET FORCEPLAN`
  - `SET IDENTITY_INSERT`
  - `SET NOCOUNT`

- `SET NOEXEC`
- `SET OFFSETS`
- `SET PARSEONLY`
- `SET QUERY_GOVERNOR_COST_LIMIT`
- `SET REMOTE_PROC_TRANSACTIONS`
- `SET ROWCOUNT`
- `SET SHOWPLAN_ALL`, `SHOWPLAN_TEXT` e `SHOWPLAN_XML`
- `SET STATISTICS`
- `SET XACT_ABORT`

Condizioni che causano l'associazione per RDS per MariaDB e RDS per MySQL

Per MariadB e MySQL, anche le seguenti interazioni causano il pinning:

- Le dichiarazioni di blocco esplicito delle tabelle `LOCK TABLE`, `LOCK TABLES` o `FLUSH TABLES WITH READ LOCK` causano il pinning della sessione da parte del proxy.
- La creazione di blocchi denominati mediante `GET_LOCK` fa sì che il proxy esegua il pinning della sessione.
- L'impostazione di una variabile utente o di una variabile di sistema (con alcune eccezioni) fa sì che il proxy effettui il pinning della sessione. Se questa situazione riduce eccessivamente il riutilizzo della connessione, scegli le operazioni che non causino il pinning. `SET` Per informazioni su come eseguire questa operazione impostando la proprietà Session pinning filters (Filtri per l'aggiunta di sessioni), consulta Creazione di un RDS Proxy e Modifica di un RDS Proxy.
- La creazione di una tabella temporanea fa sì che il proxy effettui il pinning della sessione. In questo modo, il contenuto della tabella temporanea viene conservato per tutta la sessione indipendentemente dai limiti delle transazioni.
- Le chiamate delle funzioni MySQL `ROW_COUNT`, `FOUND_ROWS` e `LAST_INSERT_ID` talvolta causano il pinning.
- Le istruzioni preparate fanno sì che il proxy effettui il pinning della sessione. Questa regola si applica se l'istruzione preparata utilizza il testo SQL o il protocollo binario.
- RDS Proxy non blocca le connessioni quando si utilizza SET LOCAL.
- Le chiamate di stored procedure e di funzioni archiviate non causano il pinning. RDS Proxy non rileva alcuna modifica dello stato della sessione derivante da tali chiamate. Assicurati che

l'applicazione non modifichi lo stato della sessione all'interno delle routine archiviate se fai affidamento su quello stato della sessione per persistere tra le transazioni. Ad esempio, RDS Proxy non è attualmente compatibile con una stored procedure che crea una tabella temporanea che persiste in tutte le transazioni.

Se disponi di un'approfondita conoscenza del comportamento dell'applicazione, puoi scegliere di ignorare il comportamento del pinning per determinate istruzioni dell'applicazione. A tale scopo, puoi selezionare l'opzione (Filtri di pinning della sessione durante la creazione del proxy. Attualmente, è possibile disattivare l'aggiunta della sessione per l'impostazione delle variabili di sessione e delle impostazioni di configurazione.

Condizioni che causano l'associazione per RDS per PostgreSQL

Per PostgreSQL, le seguenti interazioni causano anche il pinning:

- Utilizzo dei comandi`SET`.
- Utilizzo di`PREPARE`, `DISCARD``DEALLOCATE`, o `EXECUTE` comandi per gestire le istruzioni preparate.
- Creazione di sequenze, tabelle o viste temporanee.
- Dichiarazione dei cursori.
- Eliminare lo stato della sessione.
- Ascolto su un canale di notifica.
- Caricamento di un modulo di libreria come`auto_explain`.
- Manipolazione di sequenze utilizzando funzioni come e. `nextval` `setval`
- Interazione con le serrature utilizzando funzioni come e. `pg_advisory_lock` `pg_try_advisory_lock`

> **Note**
>
> RDS Proxy non prevede blocchi consultivi a livello di transazione, in particolare`pg_advisory_xact_lock`,, e. `pg_advisory_xact_lock_shared` `pg_try_advisory_xact_lock` `pg_try_advisory_xact_lock_shared`

- Impostazione di un parametro o ripristino dei valori predefiniti di un parametro. In particolare, utilizzo dei `set_config` comandi `SET` and per assegnare valori predefiniti alle variabili di sessione.

- Le chiamate di stored procedure e di funzioni archiviate non causano il pinning. RDS Proxy non rileva alcuna modifica dello stato della sessione derivante da tali chiamate. Assicurati che l'applicazione non modifichi lo stato della sessione all'interno delle routine archiviate se fai affidamento su quello stato della sessione per persistere tra le transazioni. Ad esempio, RDS Proxy non è attualmente compatibile con una stored procedure che crea una tabella temporanea che persiste in tutte le transazioni.

Eliminazione di un RDS Proxy

È possibile eliminare un proxy quando non è più necessario. In alternativa, puoi eliminare un proxy se metti fuori servizio l'istanza DB o il cluster ad esso associato.

AWS Management Console

Per eliminare un proxy

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione scegli Proxies (Proxy).
3. Scegli il proxy da eliminare dall'elenco.
4. Scegli Delete Proxy (Elimina proxy).

AWS CLI

Per eliminare un proxy DB, usa il AWS CLI comando delete-db-proxy. Per rimuovere le associazioni correlate, usa anche il deregister-db-proxy-targetscomando.

```
aws rds delete-db-proxy --name proxy_name
```

```
aws rds deregister-db-proxy-targets
    --db-proxy-name proxy_name
    [--target-group-name target_group_name]
    [--target-ids comma_separated_list]       # or
    [--db-instance-identifiers instance_id]       # or
    [--db-cluster-identifiers cluster_id]
```

API RDS

Per eliminare un proxy DB, chiama la funzione API Amazon RDS DeleteDBProxy. Per eliminare elementi e associazioni correlati, chiamate anche le funzioni DeleteDB ProxyTargetGroup e deregisterDB. ProxyTargets

Utilizzo degli endpoint Amazon RDS Proxy

Informazioni sugli endpoint per Server proxy per RDS e su come usarli. Utilizzando gli endpoint proxy, puoi sfruttare le seguenti funzionalità:

- Puoi utilizzare più endpoint con un proxy per monitorare e risolvere i problemi di connessione da diverse applicazioni in modo indipendente.
- Puoi utilizzare un endpoint tra VPC per consentire l'accesso ai database in un VPC da risorse quali istanze Amazon EC2 presenti in un VPC diverso.

Argomenti

Panoramica degli endpoint proxy

L'uso degli endpoint Server proxy per RDS include gli stessi tipi di procedure degli endpoint di istanza RDS. Se non hai familiarità con gli endpoint RDS, puoi consultare Connessione a un'istanza database che esegue il motore del database MySQL e Connessione a un'istanza database che esegue il modulo di gestione di database PostgreSQL.

Per un endpoint proxy creato, puoi inoltre associare l'endpoint a un cloud privato virtuale (VPC) diverso da quello utilizzato dal proxy stesso. In questo modo, puoi connetterti al proxy da un VPC diverso, ad esempio da un VPC utilizzato da un'applicazione diversa all'interno dell'organizzazione.

Per informazioni sui limiti associati agli endpoint proxy, consulta Limitazioni per gli endpoint proxy.

Nei log RDS Proxy, ogni voce è preceduta dal nome dell'endpoint proxy associato. Questo nome può essere quello specificato per un endpoint definito dall'utente. In alternativa, può essere il nome `default` speciale dell'endpoint predefinito di un proxy che esegue richieste di lettura/scrittura.

Ogni endpoint proxy ha il proprio set di metriche. CloudWatch Puoi monitorare i parametri per tutti gli endpoint di un proxy. Puoi inoltre monitorare i parametri per un endpoint specifico o per tutti gli endpoint di lettura/scrittura o di sola lettura di un proxy. Per ulteriori informazioni, consulta Monitoraggio dei parametri del proxy RDS con Amazon CloudWatch.

Un endpoint proxy utilizza lo stesso meccanismo di autenticazione del proxy associato. RDS Proxy imposta automaticamente i permessi e le autorizzazioni per l'endpoint definito dall'utente, coerenti con le proprietà del proxy associato.

Endpoint proxy per cluster di database Multi-AZ

Per impostazione predefinita, l'endpoint a cui ci si connette quando si utilizza Server proxy per RDS con un cluster di database Multi-AZ ha funzionalità di lettura/scrittura. Di conseguenza, questo endpoint invia tutte le richieste all'istanza di scrittura del cluster. Tutte queste connessioni vengono conteggiate nel valore `max_connections` dell'istanza di scrittura. Se il proxy è associato a un cluster di database Multi-AZ, puoi creare ulteriori endpoint di lettura/scrittura o di sola lettura per tale proxy.

Puoi utilizzare un endpoint di sola lettura con il proxy per le query di sola lettura. Puoi farlo allo stesso modo in cui utilizzi l'endpoint di lettura per un cluster di database Multi-AZ. Puoi così sfruttare la scalabilità di lettura di un cluster di database Multi-AZ con una o più istanze database di lettura. Puoi eseguire più query simultanee e creare più connessioni simultanee utilizzando un endpoint di sola lettura e aggiungendo più istanze database di lettura al cluster di database Multi-AZ in base alle necessità. Questi endpoint di lettura consentono di migliorare la scalabilità di lettura delle applicazioni che richiedono un uso intensivo di query. Gli endpoint di lettura consentono inoltre di migliorare la disponibilità delle connessioni se un'istanza database di lettura nel cluster non è disponibile.

Endpoint di lettura per cluster di database Multi-AZ

Con RDS Proxy, puoi creare e utilizzare gli endpoint di lettura. Tuttavia, questi endpoint funzionano solo per i proxy associati a cluster di database Multi-AZ. Se utilizzi la CLI RDS o l'API, è possibile che venga visualizzato l'attributo `TargetRole` con un valore di `READ_ONLY`. È possibile sfruttare tali proxy modificando la destinazione di un proxy da un'istanza DB RDS a un cluster DB Multi-AZ.

Puoi creare e connetterti a endpoint di sola lettura denominati endpoint di lettura quando utilizzi Server proxy per RDS con cluster di database Multi-AZ.

In che modo gli endpoint di lettura aiutano la disponibilità delle applicazioni

In alcuni casi, un'istanza di lettura nel cluster potrebbe non essere disponibile. In questo caso, le connessioni che utilizzano un endpoint di lettura di un proxy DB possono essere ripristinate più rapidamente di quelle che utilizzano l'endpoint di lettura del cluster di database Multi-AZ. Server proxy per RDS instrada le connessioni solo all'istanza del lettore disponibile nel cluster. Non vi è alcun ritardo dovuto alla memorizzazione nella cache DNS quando un'istanza diventa non disponibile.

Se la connessione è multiplexing, Server proxy per RDS indirizza le query successive a un'istanza di lettura diversa senza alcuna interruzione dell'applicazione. Se un'istanza di lettura si trova in uno stato non disponibile, tutte le connessioni client all'endpoint dell'istanza vengono chiuse.

Se la connessione è bloccata, la successiva query sulla connessione restituisce un errore. Tuttavia, l'applicazione può riconnettersi immediatamente allo stesso endpoint proxy. RDS Proxy indirizza la connessione a un'istanza database del lettore diversa che si trova nello stato `available`. Se ti riconnetti manualmente, Server proxy per RDS non controlla il ritardo di replica tra la vecchia e la nuova istanza di lettura.

Se il cluster di database Multi-AZ non dispone di istanze di lettura disponibili, Server proxy per RDS tenta di connettersi a un endpoint di lettura disponibile. Se nessuna istanza di lettura diventa disponibile entro il periodo di timeout di prestito della connessione, il tentativo di connessione ha esito negativo. Se invece un'istanza di lettura diventa disponibile, il tentativo di connessione ha esito positivo.

In che modo gli endpoint di lettura aiutano la scalabilità delle query

Gli endpoint di lettura per un proxy supportano la scalabilità delle query del cluster di database Multi-AZ nei seguenti modi:

- Laddove possibile, RDS Proxy utilizza la stessa istanza database di lettura per tutti i problemi di query utilizzando una particolare connessione all'endpoint di lettura. In questo modo, un insieme di query correlate sulle stesse tabelle può sfruttare la memorizzazione nella cache, l'ottimizzazione del piano e così via, su una particolare istanza database.
- Se un'istanza database di lettura non è disponibile, l'effetto sull'applicazione dipende dal fatto che la sessione sia multiplexing o bloccata. Se la sessione è multiplexing, RDS Proxy indirizza tutte le query successive a un'istanza database di lettura diversa senza richiedere alcuna azione da

parte tua. Se invece la sessione è bloccata, l'applicazione riceve un errore e deve riconnettersi. Puoi riconnetterti immediatamente all'endpoint di lettura e RDS Proxy indirizza la connessione a un'istanza database di lettura disponibile. Per ulteriori informazioni sul multiplexing e sul blocco per le sessioni proxy, consulta Panoramica dei concetti RDS Proxy.

Accesso ai database RDS su VPC

Per impostazione predefinita, i componenti dello stack di tecnologia di RDS si trovano tutti nello stesso Amazon VPC. Supponi, ad esempio, che un'applicazione in esecuzione su un'istanza Amazon EC2 si connetta a un'istanza database Amazon RDS. In questo caso, il server delle applicazioni e il database devono trovarsi entrambi all'interno dello stesso VPC.

Con RDS Proxy, puoi configurare l'accesso a un'istanza Amazon RDS DB DB del in un VPC dalle risorse di un altro VPC, come le istanze EC2. Ad esempio, l'organizzazione potrebbe avere più applicazioni che accedono alle stesse risorse del database. Ogni applicazione potrebbe trovarsi nel proprio VPC.

Per consentire l'accesso tra VPC, crea un nuovo endpoint per il proxy. Il proxy stesso risiede nello stesso VPC dell'istanza database Amazon RDS. Tuttavia, l'endpoint tra VPC si trova nell'altro VPC, insieme alle altre risorse, ad esempio le istanze EC2. L'endpoint tra VPC è associato a sottoreti e gruppi di sicurezza dello stesso VPC così come EC2 e altre risorse. Queste associazioni consentono di connettersi all'endpoint dalle applicazioni che altrimenti non potrebbero accedere al database a causa delle restrizioni del VPC.

Nella procedura seguente viene illustrato come creare e accedere a un endpoint tra VPC tramite RDS Proxy:

1. Crea due VPC o scegli due VPC già utilizzati per RDS. Ogni VPC dovrebbe avere le proprie risorse di rete associate come un gateway Internet, tabelle di routing, sottoreti e gruppi di sicurezza. Se si dispone di un solo VPC, è possibile consultare la procedura Nozioni di base su Amazon RDS per configurare un altro VPC per l'utilizzo corretto di RDS. Puoi anche esaminare il tuo VPC esistente nella console Amazon EC2 per vedere i tipi di risorse da connettere tra loro.
2. Crea un proxy database associato all'istanza database Amazon RDS a cui desideri connetterti. Segui la procedura riportata in Creazione di un RDS Proxy.
3. Nella pagina Dettagli del proxy nella console RDS, nella sezione Endpoint proxy seleziona Crea endpoint. Segui la procedura riportata in Creazione di un endpoint proxy.
4. Seleziona se impostare l'endpoint tra VPC in lettura/scrittura o in sola lettura.

5. Invece di accettare il valore predefinito dello stesso VPC dell'istanza database Amazon RDS, seleziona un VPC diverso. Questo VPC deve trovarsi nella stessa regione AWS del VPC in cui risiede il proxy.
6. Ora invece di accettare i valori predefiniti per le sottoreti e i gruppi di sicurezza dallo stesso VPC dell'istanza database Amazon RDS, effettua nuove selezioni. Le nuove selezioni dovranno essere relative alle sottoreti e ai gruppi di sicurezza del VPC scelto.
7. Non è necessario modificare alcuna delle impostazioni per i segreti di Secrets Manager. Le stesse credenziali funzionano per tutti gli endpoint del proxy, indipendentemente dal VPC in cui si trova ciascun endpoint.
8. Attendi che il nuovo endpoint raggiunga lo stato Disponibile.
9. Prendi nota del nome completo dell'endpoint. Questo è il valore che termina in *`Region_name`*`.rds.amazonaws.com` fornito come parte della stringa di connessione per l'applicazione di database.
10. Accedi al nuovo endpoint da una risorsa nello stesso VPC dell'endpoint. Un modo semplice per testare questo processo consiste nel creare una nuova istanza EC2 in questo VPC. Quindi, accedi all'istanza EC2 ed `psql` esegui i comandi `mysql` or per connetterti utilizzando il valore dell'endpoint nella stringa di connessione.

Creazione di un endpoint proxy

Console

Per creare un endpoint proxy

1. Accedi alla AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione scegli Proxies (Proxy).
3. Fai clic sul nome del proxy per il quale desideri creare un nuovo endpoint.

   Sarà visualizzata la pagina dei dettagli del proxy.
4. In Endpoint proxy, seleziona Crea endpoint proxy.

   Verrà visualizzata la finestra Crea endpoint proxy.
5. Per Nome endpoint proxy, specifica un nome descrittivo a scelta.
6. Per Ruolo di destinazione, scegli se rendere l'endpoint di lettura/scrittura o di sola lettura.

Le connessioni che utilizzano endpoint di lettura/scrittura possono eseguire qualsiasi tipo di operazione, come istruzioni DDL (Data Definition Language), istruzioni DML (Data Manipulation Language) e query. Questi endpoint si connettono sempre all'istanza primaria del cluster di database RDS. Puoi utilizzare gli endpoint di lettura/scrittura per le operazioni generali del database nel caso in cui utilizzi un singolo endpoint nell'applicazione. È inoltre possibile utilizzare endpoint di lettura/scrittura per operazioni amministrative, applicazioni di elaborazione delle transazioni online (OLTP) e lavori (ETL). extract-transform-load

Le connessioni che utilizzano un endpoint di sola lettura possono soltanto eseguire query. Server proxy per RDS può utilizzare una delle istanze di lettura per ogni connessione all'endpoint. In questo modo, un'applicazione ad uso intensivo di query può sfruttare i vantaggi della capacità di clustering dei cluster di database Multi-AZ. Queste connessioni di sola lettura non impongono alcun sovraccarico sull'istanza primaria del cluster. In questo modo, le query di report e analisi non rallentano le operazioni di scrittura delle applicazioni OLTP.

7. Per Virtual Private Cloud (VPC), scegli l'impostazione predefinita per accedere all'endpoint dalle stesse istanze EC2 o da altre risorse che normalmente vengono utilizzate per accedere al proxy o al database associato. Per impostare l'accesso tra VPC per questo proxy, scegli un VPC diverso da quello predefinito. Per ulteriori informazioni sull'accesso tra VPC, consulta Accesso ai database RDS su VPC.
8. Per Sottoreti, RDS Proxy riempie le stesse sottoreti del proxy associato per impostazione predefinita. Per limitare l'accesso all'endpoint solo a una parte dell'intervallo di indirizzi del VPC a cui è possibile connettersi, rimuovi una o più sottoreti.
9. Per Gruppo di sicurezza VPC puoi selezionare un gruppo di sicurezza esistente o crearne uno nuovo. Per impostazione predefinita, RDS Proxy riempie lo stesso gruppo o gruppi di sicurezza del proxy associato. Se le regole in entrata e in uscita per il proxy sono appropriate per questo endpoint, mantieni la scelta predefinita.

   Se decidi di creare un nuovo gruppo di sicurezza, specifica un nome per il gruppo di sicurezza in questa pagina. Quindi modifica le impostazioni del gruppo di sicurezza dalla console EC2 in un secondo momento.
10. Scegli Crea endpoint proxy.

AWS CLI

Per creare un endpoint proxy, usa il AWS CLI create-db-proxy-endpointcomando.

Includi i parametri obbligatori seguenti:

- `--db-proxy-name` *`value`*
- `--db-proxy-endpoint-name` *`value`*
- `--vpc-subnet-ids` *`list_of_ids`*. Separa gli ID delle sottoreti con gli spazi. Non specificare l'ID del VPC stesso.

Puoi inoltre includere i seguenti parametri facoltativi:

- `--target-role { READ_WRITE | READ_ONLY }`. Questo parametro per impostazione predefinita è `READ_WRITE`. Quando il proxy è associato a un cluster DB Multi-AZ, un cluster che contiene solo un'istanza Writer DB, non è possibile specificare. `READ_ONLY` Per ulteriori informazioni sull'uso previsto degli endpoint di sola lettura con Endpoint di lettura per cluster di database Multi-AZ
- `--vpc-security-group-ids` *`value`*. Separa gli ID dei gruppi di sicurezza con gli spazi. Se ometti questo parametro, RDS Proxy utilizza il gruppo di sicurezza predefinito per il VPC. RDS Proxy determina il VPC in base agli ID sottorete specificati per il parametro `--vpc-subnet-ids`.

Example

Nell'esempio seguente viene creato un endpoint proxy denominato `my-endpoint`.

LinuxmacOSPer, o: Unix

```
aws rds create-db-proxy-endpoint \
  --db-proxy-name my-proxy \
  --db-proxy-endpoint-name my-endpoint \
  --vpc-subnet-ids subnet_id subnet_id subnet_id ... \
  --target-role READ_ONLY \
  --vpc-security-group-ids security_group_id ]
```

Per Windows:

```
aws rds create-db-proxy-endpoint ^
  --db-proxy-name my-proxy ^
  --db-proxy-endpoint-name my-endpoint ^
  --vpc-subnet-ids subnet_id_1 subnet_id_2 subnet_id_3 ... ^
  --target-role READ_ONLY ^
```

```
  --vpc-security-group-ids security_group_id
```

API RDS

Per creare un endpoint proxy, utilizza l'azione ProxyEndpointCreateDB dell'API RDS.

Visualizzazione degli endpoint proxy

Console

Per visualizzare i dettagli di un endpoint proxy

1. Accedi alla AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione scegli Proxies (Proxy).
3. Nell'elenco scegli il proxy di cui si desidera visualizzare l'endpoint. Fai clic sul nome del proxy per visualizzarne la pagina dei dettagli.
4. Nella sezione Endpoint proxy, scegli l'endpoint che desideri visualizzare. Fai clic sul relativo nome per visualizzare la pagina dei dettagli.
5. Esamina i parametri di cui ti interessano i valori. Puoi controllare proprietà come le seguenti:
   - Se l'endpoint è di lettura/scrittura o di sola lettura.
   - L'indirizzo dell'endpoint utilizzato in una stringa di connessione al database.
   - Le sottoreti e i gruppi di sicurezza del VPC associati all'attività.

AWS CLI

Per visualizzare uno o più endpoint proxy, usa il comando. AWS CLI describe-db-proxy-endpoints

Puoi includere i seguenti parametri facoltativi:

- `--db-proxy-endpoint-name`
- `--db-proxy-name`

Nell'esempio seguente viene descritto l'endpoint proxy `my-endpoint`.

Example

Per LinuxmacOS, oUnix:

```
aws rds describe-db-proxy-endpoints \
  --db-proxy-endpoint-name my-endpoint
```

Per Windows:

```
aws rds describe-db-proxy-endpoints ^
  --db-proxy-endpoint-name my-endpoint
```

API RDS

Per descrivere uno o più endpoint proxy, utilizzate l'operazione RDS API ProxyEndpointsDescribeDB.

Modifica di un endpoint proxy

Console

Per modificare uno o più endpoint proxy

1. Accedi alla AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione scegli Proxies (Proxy).
3. Nell'elenco seleziona il proxy di cui desideri modificare l'endpoint. Fai clic sul nome del proxy per visualizzarlo
4. Nella sezione Endpoint proxy, scegli l'endpoint che desideri modificare. Puoi selezionarlo dall'elenco oppure fare clic sul relativo nome e visualizzarne la pagina dei dettagli.
5. Nella pagina dei dettagli del proxy, sotto la sezione Endpoint proxy, seleziona Modifica. Oppure, nella pagina dei dettagli dell'endpoint proxy, per Azioni, scegli Modifica.
6. Modificare i valori dei parametri desiderati.
7. Seleziona Save changes (Salva modifiche).

AWS CLI

Per modificare un endpoint proxy, utilizza il AWS CLI modify-db-proxy-endpointcomando con i seguenti parametri richiesti:

- `--db-proxy-endpoint-name`

Specifica le modifiche alle proprietà dell'endpoint utilizzando uno o più dei seguenti parametri:

- `--new-db-proxy-endpoint-name`
- `--vpc-security-group-ids`. Separa gli ID dei gruppi di sicurezza con gli spazi.

Nel seguente esempio l'endpoint proxy `my-endpoint` viene ridenominato in `new-endpoint-name`.

Example

Per LinuxmacOS, oUnix:

```
aws rds modify-db-proxy-endpoint \
  --db-proxy-endpoint-name my-endpoint \
  --new-db-proxy-endpoint-name new-endpoint-name
```

Per Windows:

```
aws rds modify-db-proxy-endpoint ^
  --db-proxy-endpoint-name my-endpoint ^
  --new-db-proxy-endpoint-name new-endpoint-name
```

API RDS

Per modificare un endpoint proxy, utilizzate l'operazione ProxyEndpointModifyDB dell'API RDS.

Eliminazione di un endpoint proxy

Puoi eliminare un endpoint per il proxy utilizzando la console come descritto di seguito.

> **Note**
>
> Non è possibile eliminare l'endpoint proxy predefinito che RDS Proxy crea automaticamente per ogni proxy.
> Quando elimini un proxy, RDS Proxy elimina automaticamente tutti gli endpoint associati.

Console

Per eliminare un endpoint proxy utilizzando il AWS Management Console

1. Nel riquadro di navigazione scegli Proxies (Proxy).
2. Nell'elenco seleziona il proxy per cui desideri eliminare l'endpoint. Fai clic sul nome del proxy per visualizzarne la pagina dei dettagli.
3. Nella sezione Endpoint proxy, scegli l'endpoint che desideri eliminare. Puoi selezionare uno o più endpoint dall'elenco oppure fare clic sul nome di un singolo endpoint e visualizzarne la pagina dei dettagli.
4. Nella pagina dei dettagli del proxy, sotto la sezione Endpoint proxy, seleziona Elimina. Oppure, nella pagina dei dettagli dell'endpoint proxy, per Azioni, scegli Elimina.

AWS CLI

Per eliminare un endpoint proxy, esegui il delete-db-proxy-endpointcomando con i seguenti parametri richiesti:

- `--db-proxy-endpoint-name`

Il comando seguente elimina l'endpoint proxy denominato `my-endpoint`.

Per LinuxmacOS, oUnix:

```
aws rds delete-db-proxy-endpoint \
  --db-proxy-endpoint-name my-endpoint
```

Per Windows:

```
aws rds delete-db-proxy-endpoint ^
  --db-proxy-endpoint-name my-endpoint
```

API RDS

Per eliminare un endpoint proxy con l'API RDS, esegui l'operazione ProxyEndpointDeleteDB. Specifica il nome dell'endpoint proxy per il parametro `DBProxyEndpointName`.

Limitazioni per gli endpoint proxy

Gli endpoint proxy RDS presentano le seguenti limitazioni:

- Ogni proxy dispone di un endpoint predefinito che è possibile modificare ma non creare o eliminare.
- Il numero massimo di endpoint definiti dall'utente per un proxy è 20. Pertanto, un proxy può avere fino a 21 endpoint: l'endpoint predefinito più 20 creati.
- Quando associ degli endpoint aggiuntivi a un proxy, RDS Proxy determina automaticamente quali istanze database nel cluster utilizzare per ciascun endpoint.

Monitoraggio dei parametri del proxy RDS con Amazon CloudWatch

Puoi monitorare RDS Proxy utilizzando Amazon CloudWatch. CloudWatch raccoglie ed elabora i dati grezzi dai proxy in metriche leggibili. near-real-time Per trovare queste metriche nella CloudWatch console, scegli Metriche, quindi scegli RDS e scegli Metriche per proxy. Per ulteriori informazioni, consulta Using Amazon CloudWatch metrics nella Amazon CloudWatch User Guide.

> **Note**
>
> RDS pubblica questi parametri per ogni istanza Amazon EC2 sottostante associata al proxy. Un singolo proxy potrebbe essere servito da più di un'istanza EC2. Utilizza CloudWatch le statistiche per aggregare i valori di un proxy in tutte le istanze associate.
> Alcuni di questi parametri potrebbero non essere visibili fino al completamento della prima connessione mediante un proxy.

Nei log RDS Proxy, ogni voce è preceduta dal nome dell'endpoint proxy associato. Questo nome può essere il nome specificato per un endpoint definito dall'utente o il nome speciale `default` per l'endpoint predefinito di un proxy che esegue richieste di lettura/scrittura.

Tutte le metriche RDS Proxy sono nel gruppo `proxy`.

Ogni endpoint proxy ha le proprie metriche. CloudWatch Puoi monitorare l'utilizzo di ciascun endpoint proxy in modo indipendente. Per ulteriori informazioni sugli endpoint proxy, consulta Utilizzo degli endpoint Amazon RDS Proxy.

Puoi aggregare i valori per ogni parametro utilizzando uno dei seguenti set di dimensioni. Ad esempio, utilizzando il metodo `ProxyName`, puoi analizzare tutto il traffico per un determinato proxy. Utilizzando gli altri set di dimensioni, puoi suddividere i parametri in modi diversi. Puoi suddividere i parametri in base ai diversi endpoint o ai database di destinazione di ciascun proxy oppure al traffico di lettura/scrittura e di sola lettura verso ciascun database.

- Set di dimensioni 1: `ProxyName`
- Set di dimensioni 2: `ProxyName`, `EndpointName`
- Set di dimensioni 3: `ProxyName`, `TargetGroup`, `Target`
- Set di dimensioni 4: `ProxyName`, `TargetGroup`, `TargetRole`

Parametro	Descrizione	Periodo valido	CloudWatch set di dimensioni
`AvailabilityPercentage`	Percentuale di tempo durante il quale il gruppo target era disponibile nel ruolo indicato dalla dimensione. Questa metrica viene segnalata ogni minuto. La statistica più utile per questo parametro è `Average`.	1 minuto	Dimension set 4
`ClientConnections`	Il numero corrente di connessioni client. Questa metrica viene segnalata ogni minuto. La statistica più utile per questo parametro è `Sum`.	1 minuto	Dimension set 1, Dimension set 2

Parametro	Descrizione	Periodo valido	CloudWatch set di dimensioni
`ClientConnectionsClosed`	Il numero di connessioni client chiuse. La statistica più utile per questo parametro è `Sum`.	1 minuto e oltre	Dimension set 1, Dimension set 2
`ClientConnectionsNoTLS`	Numero corrente di connessioni client senza Transport Layer Security (TLS). Questa metrica viene segnalata ogni minuto. La statistica più utile per questo parametro è `Sum`.	1 minuto e oltre	Dimension set 1, Dimension set 2
`ClientConnectionsReceived`	Numero di richieste di connessione client ricevute. La statistic a più utile per questo parametro è `Sum`.	1 minuto e oltre	Dimension set 1, Dimension set 2
`ClientConnectionsSetupFailedAuth`	Numero di tentativi di connessione client non riusciti a causa di autenticazione errata o configurazione TLS errata. La statistica più utile per questo parametro è `Sum`.	1 minuto e oltre	Dimension set 1, Dimension set 2

Parametro	Descrizione	Periodo valido	CloudWatch set di dimensioni
`ClientConnectionsSetupSucceeded`	Il numero di connessioni client stabilito correttamente con qualsiasi meccanismo di autenticazione con o senza TLS. La statistica più utile per questo parametro è `Sum`.	1 minuto e oltre	Dimension set 1, Dimension set 2
`ClientConnectionsTLS`	Il numero corrente di connessioni client con TLS. Questa metrica viene segnalata ogni minuto. La statistica più utile per questo parametro è `Sum`.	1 minuto e oltre	Dimension set 1, Dimension set 2
`DatabaseConnectionRequests`	Numero di richieste per creare una connessione al database. La statistica più utile per questo parametro è `Sum`.	1 minuto e oltre	Dimension set 1, Dimension set 3, Dimension set 4
`DatabaseConnectionRequestsWithTLS`	Numero di richieste per creare una connessione al database con TLS. La statistica più utile per questo parametro è `Sum`.	1 minuto e oltre	Dimension set 1, Dimension set 3, Dimension set 4

Parametro	Descrizione	Periodo valido	CloudWatch set di dimensioni
`DatabaseConnections`	Il numero corrente di connessioni al database. Questa metrica viene segnalata ogni minuto. La statistica più utile per questo parametro è `Sum`.	1 minuto	Dimension set 1, Dimension set 3, Dimension set 4
`DatabaseConnectionsBorrowLatency`	Il tempo in microsecondi necessario per il proxy monitorato per ottenere una connessione al database. La statistica più utile per questo parametro è `Average`.	1 minuto e oltre	Dimension set 1, Dimension set 2
`DatabaseConnectionsCurrentlyBorrowed`	Il numero corrente di connessioni al database nello stato di prestito. Questa metrica viene segnalata ogni minuto. La statistica più utile per questo parametro è `Sum`.	1 minuto	Dimension set 1, Dimension set 3, Dimension set 4

Parametro	Descrizione	Periodo valido	CloudWatch set di dimensioni
`DatabaseConnectionsCurrentlyInTransaction`	Il numero corrente di connessioni al database in una transazione. Questa metrica viene segnalata ogni minuto. La statistica più utile per questo parametro è `Sum`.	1 minuto	Dimension set 1, Dimension set 3, Dimension set 4
`DatabaseConnectionsCurrentlySessionPinned`	Il numero corrente di connessioni al database attualmente bloccati a causa di operazioni nelle richieste client che modificano lo stato della sessione. Questa metrica viene segnalata ogni minuto. La statistica più utile per questo parametro è `Sum`.	1 minuto	Dimension set 1, Dimension set 3, Dimension set 4
`DatabaseConnectionsSetupFailed`	Il numero di richieste di connessione al database non riuscite. La statistica più utile per questo parametro è `Sum`.	1 minuto e oltre	Dimension set 1, Dimension set 3, Dimension set 4

Parametro	Descrizione	Periodo valido	CloudWatch set di dimensioni
`DatabaseConnectionsSetupSucceeded`	Il numero di connessioni al database stabilito correttamente con o senza TLS. La statistica più utile per questo parametro è `Sum`.	1 minuto e oltre	Dimension set 1, Dimension set 3, Dimension set 4
`DatabaseConnectionsWithTLS`	Il numero corrente di connessioni al database con TLS. Questa metrica viene segnalata ogni minuto. La statistica più utile per questo parametro è `Sum`.	1 minuto	Dimension set 1, Dimension set 3, Dimension set 4
`MaxDatabaseConnectionsAllowed`	Il numero massimo di connessioni al database consentite. Questa metrica viene segnalata ogni minuto. La statistica più utile per questo parametro è `Sum`.	1 minuto	Dimension set 1, Dimension set 3, Dimension set 4
`QueryDatabaseResponseLatency`	Tempo in microsecondi impiegato dal database per rispondere alla query. La statistica più utile per questo parametro è `Average`.	1 minuto e oltre	Dimension set 1, Dimension set 2, Dimension set 3, Dimension set 4

Parametro	Descrizione	Periodo valido	CloudWatch set di dimensioni
`QueryRequests`	Il numero di query ricevute. Una query che include più istruzioni viene conteggiata come una query. La statistica più utile per questo parametro è `Sum`.	1 minuto e oltre	Dimension set 1, Dimension set 2
`QueryRequestsNoTLS`	Numero di query ricevute da connessioni non TLS. Una query che include più istruzioni viene conteggiata come una query. La statistica più utile per questo parametro è `Sum`.	1 minuto e oltre	Dimension set 1, Dimension set 2
`QueryRequestsTLS`	Numero di query ricevute dalle connessioni TLS. Una query che include più istruzioni viene conteggiata come una query. La statistica più utile per questo parametro è `Sum`.	1 minuto e oltre	Dimension set 1, Dimension set 2

Parametro	Descrizione	Periodo valido	CloudWatch set di dimensioni
`QueryResponseLatency`	Il tempo in microsecondi tra l'ottenimento di una richiesta di query e il proxy che risponde ad essa. La statistica più utile per questo parametro è `Average`.	1 minuto e oltre	Dimension set 1, Dimension set 2

È possibile trovare i registri delle attività del proxy RDS CloudWatch in. AWS Management Console Ogni proxy ha una voce nella pagina Log groups (Gruppi di registro) .

> **Important**
>
> Questi registri sono destinati all'utilizzo umano per scopi di risoluzione dei problemi e non per l'accesso programmatico. Il formato e il contenuto dei registri sono soggetti a modifiche.
> In particolare, i log meno recenti non contengono prefissi che indicano l'endpoint per ogni richiesta. Nei log più recenti, ogni voce è preceduta dal nome dell'endpoint proxy associato. Questo nome può essere il nome specificato per un endpoint definito dall'utente o il nome speciale `default` per le richieste che utilizzano l'endpoint predefinito di un proxy.

Utilizzo degli eventi RDS Proxy

Un evento indica una modifica in un ambiente, ad esempio un AWS ambiente o un servizio o un'applicazione di un partner Software as a Service (SaaS). In alternativa, può essere una delle vostre applicazioni o servizi personalizzati. Ad esempio, Amazon RDS genera un evento quando si crea o si modifica un proxy RDS. Amazon RDS Aurora offre eventi ad EventBridge Amazon quasi in tempo reale. Di seguito puoi trovare un elenco di eventi RDS Proxy a cui puoi iscriverti e un esempio di evento RDS Proxy.

Per ulteriori informazioni sull'utilizzo degli eventi, consulta quanto segue:

- Per istruzioni su come visualizzare gli eventi utilizzando l'API, o AWS Management Console RDS AWS CLI, consulta. Visualizzazione di eventi Amazon RDS
- Per informazioni su come configurare Amazon RDS Aurora a cui EventBridge inviare eventi, consulta. Creazione di una regola che si attiva su un evento Amazon RDS

Eventi RDS Proxy

La tabella seguente riporta la categoria di eventi e un elenco di eventi applicabili quando il tipo di origine è un proxy RDS.

Categoria	ID evento RDS	Messaggio	Note
modifica della configura zione	RDS-EVENT-0204	Proxy DB *nome* modificato da RDS.	
modifica della configura zione	RDS-EVENT-0207	RDS ha modificato l'endpoint del proxy DB *nome*.	
modifica della configura zione	RDS-EVENT-0213	RDS ha rilevato l'aggiunta dell'istanza database e l'ha aggiunta automaticamente al gruppo di destinazione del proxy DB *nome*.	
modifica della configura zione	RDS-EVENT-0213	RDS ha rilevato la creazione dell'istanza database *nome* e l'ha rimossa automaticamente dal gruppo di destinazione *nome* del proxy DB *nome*.	
modifica della configura zione	RDS-EVENT-0214	RDS ha rilevato l'elimina zione dell'istanza database *nome* e l'ha rimossa automaticamente dal	

Categoria	ID evento RDS	Messaggio	Note
		gruppo di destinazione *nome* del proxy DB *nome*.	
modifica della configura zione	RDS-EVENT-0215	RDS ha rilevato l'elimina zione del cluster database *nome* e l'ha rimosso automaticamente dal gruppo di destinazione *nome* del proxy DB *nome*.	
creazione	RDS-EVENT-0203	RDS ha creato il proxy DB *nome*.	
creazione	RDS-EVENT-0206	RDS ha creato l'endpoint *nome* del proxy DB *nome*.	
eliminazione	RDS-EVENT-0205	RDS ha eliminato il proxy DB *nome*.	
eliminazione	RDS-EVENT-0208	RDS ha eliminato l'endpoint *nome* per il proxy DB *nome*.	
errore	RDS-EVENT-0243	RDS non è riuscito ad eseguire il provisioning della capacità per il proxy *nome* perché non ci sono sufficienti indirizzi IP disponibili nelle sottoreti : *nome*. Per risolvere il problema, assicurarsi che le sottoreti abbiano il numero minimo di indirizzi IP non utilizzati come consigliato nella documenta zione di Server proxy per Amazon RDS.	Per determinare il numero consigliato per la classe di istanza, consulta Pianifica zione della capacità degli indirizzi IP.

Categoria	ID evento RDS	Messaggio	Note
errore	RDS-EVENT-0275	*RDS ha limitato alcune connessioni al nome del proxy DB.* Il numero di richieste di connessione simultane e dal client al proxy ha superato il limite.	

Di seguito è riportato un esempio di evento di RDS Proxy in formato JSON. L'evento mostra che RDS ha modificato l'endpoint denominato `my-endpoint` del proxy RDS denominato `my-rds-proxy`. L'ID evento è RDS-EVENT-0207.

```
{
  "version": "0",
  "id": "68f6e973-1a0c-d37b-f2f2-94a7f62ffd4e",
  "detail-type": "RDS DB Proxy Event",
  "source": "aws.rds",
  "account": "123456789012",
  "time": "2018-09-27T22:36:43Z",
  "region": "us-east-1",
  "resources": [
     "arn:aws:rds:us-east-1:123456789012:db-proxy:my-rds-proxy"
  ],
  "detail": {
    "EventCategories": [
      "configuration change"
    ],
    "SourceType": "DB_PROXY",
    "SourceArn": "arn:aws:rds:us-east-1:123456789012:db-proxy:my-rds-proxy",
    "Date": "2018-09-27T22:36:43.292Z",
    "Message": "RDS modified endpoint my-endpoint of DB Proxy my-rds-proxy.",
    "SourceIdentifier": "my-endpoint",
    "EventID": "RDS-EVENT-0207"
  }
}
```

Esempi della riga di comando per RDS Proxy

Per vedere come interagiscono le combinazioni di comandi di connessione e istruzioni SQL con RDS Proxy, consulta gli esempi seguenti.

Esempi

- Preserving Connections to a MySQL Database Across a Failover
- Adjusting the max_connections Setting for an Aurora DB Cluster

Example Mantenimento delle connessioni a un database MySQL attraverso un failover

In questo esempio di MySQL viene illustrato come le connessioni aperte continuano a funzionare durante un failover, come quando un database viene riavviato o diventa non disponibile a causa di un problema. In questo esempio viene utilizzato un proxy denominato `the-proxy` e un cluster di database Aurora con istanze DB `instance-8898` e `instance-9814`. Quando il comando `failover-db-cluster` viene eseguito dalla riga di comando di Linux, l'istanza writer a cui il proxy è connesso cambia in un'istanza database diversa. Puoi vedere che l'istanza database associata al proxy cambia mentre la connessione rimane aperta.

```
$ mysql -h the-proxy.proxy-demo.us-east-1.rds.amazonaws.com -u admin_user -p
Enter password:
...

mysql> select @@aurora_server_id;
+--------------------+
| @@aurora_server_id |
+--------------------+
| instance-9814      |
+--------------------+
1 row in set (0.01 sec)

mysql>
[1]+  Stopped                 mysql -h the-proxy.proxy-demo.us-east-1.rds.amazonaws.com
 -u admin_user -p
$ # Initially, instance-9814 is the writer.
$ aws rds failover-db-cluster --db-cluster-identifier cluster-56-2019-11-14-1399
JSON output
$ # After a short time, the console shows that the failover operation is complete.
$ # Now instance-8898 is the writer.
$ fg
```

```
mysql -h the-proxy.proxy-demo.us.us-east-1.rds.amazonaws.com -u admin_user -p

mysql> select @@aurora_server_id;
+--------------------+
| @@aurora_server_id |
+--------------------+
| instance-8898      |
+--------------------+
1 row in set (0.01 sec)

mysql>
[1]+  Stopped                 mysql -h the-proxy.proxy-demo.us-east-1.rds.amazonaws.com
 -u admin_user -p
$ aws rds failover-db-cluster --db-cluster-identifier cluster-56-2019-11-14-1399
JSON output
$ # After a short time, the console shows that the failover operation is complete.
$ # Now instance-9814 is the writer again.
$ fg
mysql -h the-proxy.proxy-demo.us-east-1.rds.amazonaws.com -u admin_user -p

mysql> select @@aurora_server_id;
+--------------------+
| @@aurora_server_id |
+--------------------+
| instance-9814      |
+--------------------+
1 row in set (0.01 sec)
+---------------+---------------+
| Variable_name | Value         |
+---------------+---------------+
| hostname      | ip-10-1-3-178 |
+---------------+---------------+
1 row in set (0.02 sec)
```

Example Regolazione dell'impostazione max_connections per un cluster di database Aurora

In questo esempio viene illustrato come è possibile regolare l'impostazione `max_connections` per un cluster di database Aurora MySQL. A tale scopo, crei il gruppo di parametri del cluster di database in base alle impostazioni dei parametri predefinite per i cluster compatibili con MySQL 5.7. Specifichi un valore per l'impostazione `max_connections`, sovrascrivendo la formula che imposta il valore predefinito. Associ il gruppo di parametri del cluster di database al cluster di database.

```
export REGION=us-east-1
```

```
export CLUSTER_PARAM_GROUP=rds-proxy-mysql-57-max-connections-demo
export CLUSTER_NAME=rds-proxy-mysql-57

aws rds create-db-parameter-group --region $REGION \
  --db-parameter-group-family aurora-mysql5.7 \
  --db-parameter-group-name $CLUSTER_PARAM_GROUP \
  --description "Aurora MySQL 5.7 cluster parameter group for RDS Proxy demo."

aws rds modify-db-cluster --region $REGION \
  --db-cluster-identifier $CLUSTER_NAME \
  --db-cluster-parameter-group-name $CLUSTER_PARAM_GROUP

echo "New cluster param group is assigned to cluster:"
aws rds describe-db-clusters --region $REGION \
  --db-cluster-identifier $CLUSTER_NAME \
  --query '*[*].{DBClusterParameterGroup:DBClusterParameterGroup}'

echo "Current value for max_connections:"
aws rds describe-db-cluster-parameters --region $REGION \
  --db-cluster-parameter-group-name $CLUSTER_PARAM_GROUP \
  --query '*[*].{ParameterName:ParameterName,ParameterValue:ParameterValue}' \
  --output text | grep "^max_connections"

echo -n "Enter number for max_connections setting: "
read answer

aws rds modify-db-cluster-parameter-group --region $REGION --db-cluster-parameter-
group-name $CLUSTER_PARAM_GROUP \
  --parameters "ParameterName=max_connections,ParameterValue=$
$answer,ApplyMethod=immediate"

echo "Updated value for max_connections:"
aws rds describe-db-cluster-parameters --region $REGION \
  --db-cluster-parameter-group-name $CLUSTER_PARAM_GROUP \
  --query '*[*].{ParameterName:ParameterName,ParameterValue:ParameterValue}' \
  --output text | grep "^max_connections"
```

Risoluzione dei problemi per RDS Proxy

Di seguito, sono disponibili idee per la risoluzione di alcuni problemi comuni relativi al proxy RDS e informazioni sui CloudWatch registri di RDS Proxy.

Nei log RDS Proxy, ogni voce è preceduta dal nome dell'endpoint proxy associato. Questo nome può essere quello specificato per un endpoint definito dall'utente. In alternativa, può essere il nome speciale dell'endpoint predefinito `default` di un proxy che esegue richieste di lettura/scrittura. Per ulteriori informazioni sugli endpoint proxy, consulta Utilizzo degli endpoint Amazon RDS Proxy.

Argomenti

Verifica della connettività a un proxy

È possibile utilizzare i seguenti comandi per verificare che tutti i componenti, ad esempio il proxy, il database e le istanze di calcolo presenti nella connessione, possano comunicare tra loro.

Esamina il proxy stesso usando il describe-db-proxiescomando. Esamina anche il gruppo target associato utilizzando il comando describe-db-proxy-target-groups. Verifica che i dettagli delle destinazioni corrispondano all'istanza database RDS che intendi associare al proxy. Utilizzare comandi come i seguenti.

```
aws rds describe-db-proxies --db-proxy-name $DB_PROXY_NAME
aws rds describe-db-proxy-target-groups --db-proxy-name $DB_PROXY_NAME
```

Per confermare che il proxy è in grado di connettersi al database sottostante, esamina le destinazioni specificate nei gruppi di destinazione utilizzando il describe-db-proxy-targetscomando. Utilizzare un comando come il seguente.

```
aws rds describe-db-proxy-targets --db-proxy-name $DB_PROXY_NAME
```

L'output del describe-db-proxy-targetscomando include un `TargetHealth` campo. È possibile esaminare i campi `State`, `Reason` e `Description` all'interno di `TargetHealth` per verificare se il proxy può comunicare con l'istanza database sottostante.

- Un valore `State` di `AVAILABLE` indica che il proxy può connettersi all'istanza database.
- Un valore `State` di `UNAVAILABLE` indica un problema di connessione temporaneo o permanente. In questo caso, esaminare i campi `Reason` e `Description`. Ad esempio, se `Reason` ha un valore pari a `PENDING_PROXY_CAPACITY`, provare a connettersi nuovamente dopo che il proxy ha terminato l'operazione di ridimensionamento. Se `Reason` ha un valore di `UNREACHABLE`,

`CONNECTION_FAILED` o `AUTH_FAILURE`, utilizzare la spiegazione del campo `Description` per facilitare la diagnosi del problema.

- Il valore del campo `State` potrebbe essere `REGISTERING` per un breve periodo prima di passare a `AVAILABLE` o `UNAVAILABLE`.

Se il comando Netcat (`nc`) seguente segnala l'esito positivo, puoi accedere all'endpoint del proxy dall'istanza EC2 o da un altro sistema in cui hai eseguito l'accesso. Questo comando segnala un errore se non ti trovi nello stesso VPC del proxy e del database associato. Potresti essere in grado di accedere direttamente al database senza essere nello stesso VPC. Tuttavia, non puoi accedere al proxy a meno che non ti trovi nello stesso VPC.

```
nc -zx MySQL_proxy_endpoint 3306

nc -zx PostgreSQL_proxy_endpoint 5432
```

Puoi utilizzare i seguenti comandi per assicurarti che l'istanza EC2 abbia le proprietà richieste. In particolare, il VPC per l'istanza EC2 deve essere uguale al VPC per a cui il proxy si connette.

```
aws ec2 describe-instances --instance-ids your_ec2_instance_id
```

Esamina i segreti Secrets Manager utilizzati per il proxy.

```
aws secretsmanager list-secrets
aws secretsmanager get-secret-value --secret-id your_secret_id
```

Assicurati che il `SecretString` campo visualizzato da `get-secret-value` sia codificato come una stringa JSON che include i campi `username` and`password`. Nell'esempio seguente viene illustrato il formato del campo `SecretString`.

```
{
  "ARN": "some_arn",
  "Name": "some_name",
  "VersionId": "some_version_id",
  "SecretString": '{"username":"some_username","password":"some_password"}',
  "VersionStages": [ "some_stage" ],
  "CreatedDate": some_timestamp
}
```

Problemi e soluzioni comuni

Questa sezione descrive alcuni problemi comuni e potenziali soluzioni quando si utilizza RDS Proxy.

Dopo aver eseguito il comando `aws rds describe-db-proxy-targets` CLI, se la `TargetHealth` descrizione indica`Proxy does not have any registered credentials`, verifica quanto segue:

- Per l'accesso la proxy, l'utente dispone di credenziali registrate.
- Il ruolo IAM per accedere al segreto di Secrets Manager utilizzato dal proxy è valido.

È possibile che si verifichino i seguenti eventi RDS durante la creazione di o la connessione a un proxy DB.

Categoria	ID evento RDS	Descrizione
errore	RDS-EVENT-0243	RDS non è stato in grado di allocare la capacità per il proxy perché non ci sono sufficienti indirizzi IP disponibili nelle sottoreti. Per risolvere il problema, assicurati che le sottoreti abbiano il numero minimo di indirizzi IP non utilizzati. Per determinare il numero consigliato per la classe di istanza, consulta Pianificazione della capacità degli indirizzi IP.
errore	RDS-EVENT-0275	*`RDS ha limitato alcune connessioni al nome del proxy DB.`* Il numero di richieste di connessione simultanee dal client al proxy ha superato il limite.

È possibile che si verifichino i seguenti problemi durante la creazione di un nuovo proxy o la connessione a un proxy.

Errore	Cause o soluzioni alternative
`403: The security token included in the request is invalid`	Seleziona un ruolo IAM esistente invece di crearne uno nuovo.

È possibile che si verifichino i seguenti problemi durante la connessione a un proxy MySQL.

Errore	Cause o soluzioni alternative
`ERROR 1040 (HY000): Connections rate limit exceeded (`*`limit_value`*` )`	La velocità di richieste di connessione dal client al proxy ha superato il limite.
`ERROR 1040 (HY000): IAM authentication rate limit exceeded`	Il numero di richieste simultanee con autenticazione IAM dal client al proxy ha superato il limite.
`ERROR 1040 (HY000): Number simultaneous connections exceeded (`*`limit_value`*` )`	Il numero di richieste di connessione simultanee dal client al proxy ha superato il limite.
`ERROR 1045 (28000): Access`	Il segreto Secrets Manager utilizzato dal proxy non corrisponde al nome utente e alla password di un utente di database esistente. Aggiorna le

Errore	Cause o soluzioni alternative
`denied for user 'DB_USER'@'%' (usi password: YES)`	credenziali nel segreto Secrets Manager o assicurati che l'utente del database esista e disponga della stessa password del segreto.
`ERROR 1105 (HY000): Unknown error`	Si è verificato un errore sconosciuto.
`ERROR 1231 (42000): Variable ''charact er_set_cl ient'' can't be set to the value of value`	Il valore impostato per il parametro `character_set_client` non è valido. Ad esempio, il valore `ucs2` non è valido perché può provocare un arresto anomalo del server MySQL.
`ERROR 3159 (HY000): This RDS Proxy requires TLS connections.`	Hai abilitato l'impostazione Richiedi Transport Layer Security nel proxy ma la tua connessione includeva il parametro `ssl-mode=DISABLED` nel client MySQL. Eseguire una delle operazioni seguenti: • Disattivare l'impostazione Richiedi Transport Layer Security per il proxy. • Connettersi al database utilizzando l'impostazione minima di `ssl-mode=REQUIRED` nel client MySQL.
`ERROR 2026 (HY000): SSL connection error: Internal Server Error`	L'handshake TLS con il proxy non è riuscito. Alcuni possibili motivi includono quanto segue: • SSL è richiesto ma il server non lo supporta. • Si è verificato un errore interno del server. • Si è verificato un handshake non valido.

Errore	Cause o soluzioni alternative
`ERROR 9501 (HY000): Timed-out waiting to acquire database connection`	Il proxy è in attesa di acquisire una connessione al database. Alcuni possibili motivi includono quanto segue: • Il proxy non è in grado di stabilire una connessione al database perché sono state raggiunte le connessioni massime. • Il proxy non è in grado di stabilire una connessione al database perché il database non è disponibile.

È possibile che si verifichino i seguenti problemi durante la connessione a un proxy PostgreSQL.

Errore	Causa	Soluzione
`IAM authentication is allowed only with SSL connections.`	L'utente ha tentato di connettersi al database utilizzando l'autenticazione IAM con l'impostazione `sslmode=disable` nel client PostgreSQL.	L'utente deve connettersi al database utilizzando l'impostazione minima di `sslmode=require` nel client PostgreSQL. Per ulteriori informazioni, consulta la documentazione Supporto SSL PostgreSQL.
`This RDS Proxy requires TLS connections.`	L'utente ha abilitato l'impostazione Richiedi Transport Layer Security ma ha tentato di connettersi con `sslmode=disable` nel client PostgreSQL.	Per risolvere questo errore, effettuare una delle seguenti operazioni: • Disattivare l'impostazione Richiedi Transport Layer Security del proxy. • Connettersi al database utilizzando l'impostazione minima di `sslmode=allow` nel client PostgreSQL.

Errore	Causa	Soluzione
`IAM authentication failed for user` *`user_name`* `. Check the IAM token for this user and try again.`	Questo errore potrebbe essere dovuto ai seguenti fattori: • Il client ha fornito il nome utente IAM non corretto. • Il client ha fornito un token di autorizzazione IAM non corretto per l'utente • Il client utilizza una policy IAM che non dispone delle autorizzazioni necessarie. • Il client ha fornito un token di autorizzazione IAM scaduto per l'utente.	Per correggere questo errore, effettuare le seguenti operazioni: 1. Verificare che l'utente IAM fornito esista. 2. Verificare che il token di autorizzazione IAM appartenga all'utente IAM fornito. 3. Verificare che la policy IAM disponga di autorizzazioni adeguate per RDS. 4. Verificare la validità del token di autorizzazione IAM utilizzato.
`This RDS proxy has no credentials for the role` *`role_name`* `. Check the credentials for this role and try again.`	Non c'è un Secrets Manager segreto per questo ruolo.	Aggiungere un Secrets Manager segreto per questo ruolo. Per ulteriori informazioni, consulta Configurazione delle politiche AWS Identity and Access Management (IAM).
`RDS supports only IAM, MD5, or SCRAM authentication.`	Il client di database utilizzato per connettersi al proxy utilizza un meccanismo di autenticazione non attualmente supportato dal proxy.	Se non utilizzi l'autenticazione IAM, usa l'autenticazione della password MD5 o SCRAM.

Errore	Causa	Soluzione
`A user name is missing from the connectio n startup packet. Provide a user name for this connection.`	Il client di database utilizzat o per connettersi al proxy non invia un nome utente quando si tenta di stabilire una connessione.	Assicurarsi di definire un nome utente quando si imposta una connessione al proxy utilizzan do il client PostgreSQL di propria scelta.
`Feature not supported : RDS Proxy supports only version 3.0 of the PostgreSQL messaging protocol.`	Il client PostgreSQL utilizzat o per connettersi al proxy utilizza un protocollo precedente a 3.0.	Utilizzare un client PostgreSQ L più recente che supporti il protocollo di messaggistica 3.0. Se si utilizza la CLI `psql` di PostgreSQL, utilizzare una versione maggiore o uguale a 7.4.
`Feature not supported : RDS Proxy currently doesn't support streaming replication mode.`	Il client PostgreSQL utilizzat o per connettersi al proxy sta tentando di utilizzar e la modalità di replica in streaming, che non è attualmente supportata dal proxy RDS.	Disattivare la modalità di replica in streaming nel client PostgreSQL utilizzato per la connessione.
`Feature not supported : RDS Proxy currently doesn't support the option` *`option_name`* `.`	Tramite il messaggio di avvio, il client PostgreSQL utilizzat o per connettersi al proxy richiede un'opzione che non è attualmente supportata dal proxy RDS.	Disattivare l'opzione visualizz ata come non supportata dal messaggio precedente nel client PostgreSQL utilizzato per connettersi.
`The IAM authentic ation failed because of too many competing requests.`	Il numero di richieste simultanee con autenticazione IAM dal client al proxy ha superato il limite.	Ridurre la velocità con cui vengono stabilite le connessio ni che utilizzano l'autenti cazione IAM da un client PostgreSQL.

Errore	Causa	Soluzione
`The maximum number of client connections to the proxy exceeded` *`number_value`* `.`	Il numero di richieste di connessione simultanee dal client al proxy ha superato il limite.	Ridurre il numero di connessioni attive dai client PostgreSQL a questo proxy RDS.
`Rate of connection to proxy exceeded` *`number_value`* `.`	La velocità di richieste di connessione dal client al proxy ha superato il limite.	Ridurre la velocità con cui vengono stabilite le connessioni da un client PostgreSQL.
`The password that was provided for the role` *`role_name`* `is wrong.`	La password per questo ruolo non corrisponde al segreto Secrets Manager.	Controlla il segreto per questo ruolo in Secrets Manager per vedere se la password è uguale a quella utilizzata nel client PostgreSQL.
`The IAM authentication failed for the role` *`role_name`* `. Check the IAM token for this role and try again.`	Si è verificato un problema con il token IAM utilizzato per l'autenticazione IAM.	Generare un nuovo token di autenticazione e utilizzarlo in una nuova connessione.
`IAM is allowed only with SSL connections.`	Un client ha tentato di connettersi utilizzando l'autenticazione IAM, ma SSL non è stato abilitato.	Abilitare SSL nel client PostgreSQL.
`Unknown error.`	Si è verificato un errore sconosciuto.	Contatta il supporto di AWS per indagare sul problema.

Errore	Causa	Soluzione
`Timed-out waiting to acquire database connection.`	Il proxy è in attesa di acquisire una connessione al database. Alcuni possibili motivi includono quanto segue: • Il proxy non può stabilire una connessione al database perché sono state raggiunte le connessioni massime. • Il proxy non può stabilire una connessione al database perché il database non è disponibile.	Le possibili soluzioni sono le seguenti: • Controlla la destinazione dello stato per verificare se non è disponibile. • Controllare se sono presenti transazioni e/o query di lunga durata in esecuzione. È possibile utilizzare le connessioni al database dal connection pool per un lungo periodo di tempo.
`Request returned an error: database_error .`	La connessione al database stabilita dal proxy ha restituito un errore.	La soluzione dipende dall'errore specifico del database. Un esempio : `Request returned an error: database "your-database-name" does not exist`. Ciò significa che il nome del database specificato non esiste nel server di database oppure che il nome utente utilizzato come nome del database (se non è specificato un nome del database) non esiste nel server.

Utilizzo di RDS Proxy con AWS CloudFormation

Puoi usare RDS Proxy con AWS CloudFormation. Questo ti aiuta a creare gruppi di risorse correlate. Un gruppo di questo tipo può includere un proxy che può connettersi a un'istanza database Amazon RDS appena creata. Il supporto di RDS Proxy in AWS CloudFormation coinvolge due nuovi tipi di registro: `DBProxy` e `DBProxyTargetGroup`.

L'elenco seguente mostra un modello AWS CloudFormation di esempio per RDS Proxy.

```
Resources:
 DBProxy:
   Type: AWS::RDS::DBProxy
   Properties:
     DBProxyName: CanaryProxy
     EngineFamily: MYSQL
     RoleArn:
      Fn::ImportValue: SecretReaderRoleArn
     Auth:
       - {AuthScheme: SECRETS, SecretArn: !ImportValue ProxySecret, IAMAuth: DISABLED}
     VpcSubnetIds:
       Fn::Split: [",", "Fn::ImportValue": SubnetIds]

 ProxyTargetGroup:
   Type: AWS::RDS::DBProxyTargetGroup
   Properties:
     DBProxyName: CanaryProxy
     TargetGroupName: default
     DBInstanceIdentifiers:
       - Fn::ImportValue: DBInstanceName
   DependsOn: DBProxy
```

Per ulteriori informazioni sulle risorse di questo esempio, consulta `DBProxy` e `DBProxyTargetGroup`.

Per ulteriori informazioni sulle risorse che è possibile creare utilizzando AWS CloudFormation, consulta RDS resource type reference.

Utilizzo delle integrazioni Zero-ETL di Amazon RDS con Amazon Redshift (anteprima)

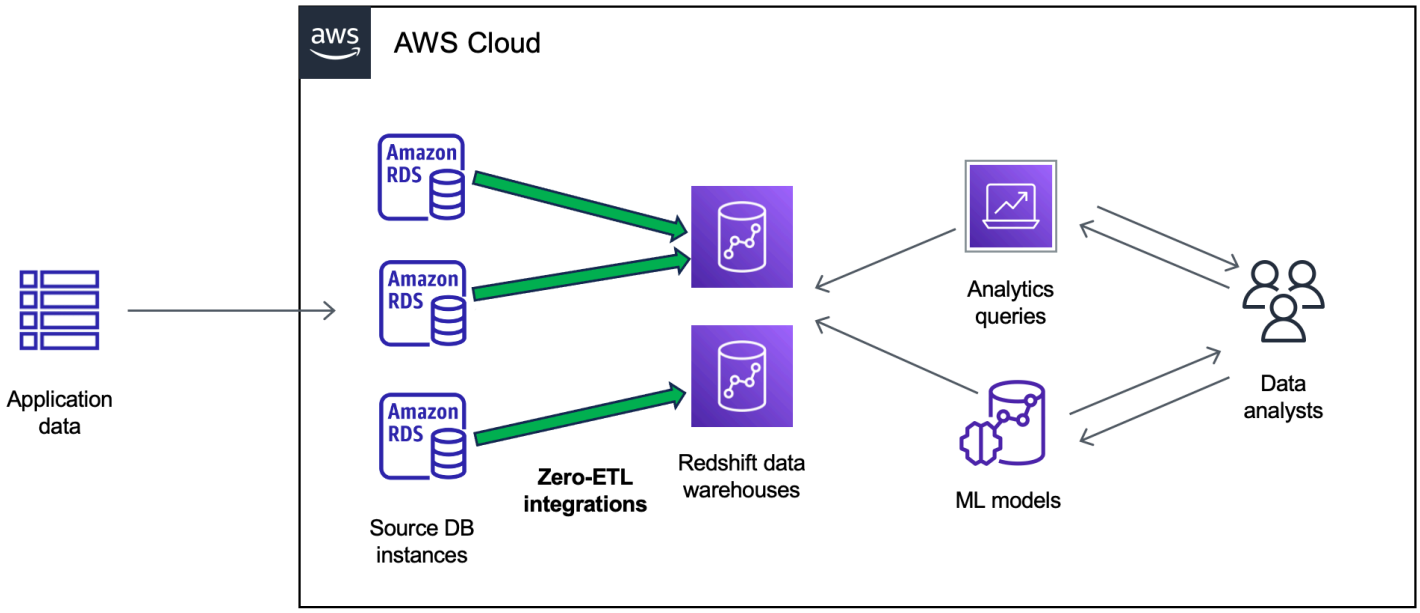
Questa è la documentazione preliminare per le integrazioni Zero-ETL di Amazon RDS con Amazon Redshift, che è in versione di anteprima. La documentazione e la funzionalità sono soggette a modifiche. Consigliamo di utilizzare questa caratteristica solo in ambienti di test e non in ambienti di produzione. Per i termini e condizioni dell'anteprima, consulta la sezione su beta e anteprime nei AWS termini del servizio.

L'integrazione Zero-ETL di Amazon RDS con Amazon Redshift consente di eseguire operazioni di analisi e machine learning (ML) quasi in tempo reale utilizzando Amazon Redshift su petabyte di dati transazionali di RDS. L'estrazione, la trasformazione e il caricamento (ETL) è il processo di combinazione di dati provenienti da più fonti in un ampio data warehouse centrale.

Un'integrazione zero-ETL rende i dati del cluster in tempo reale. Una volta che i dati sono in Amazon Redshift, puoi potenziare i tuoi carichi di lavoro di analisi, ML e intelligenza artificiale utilizzando le funzionalità integrate di Amazon Redshift, come l'apprendimento automatico, le viste materializzate, la condivisione dei dati, l'accesso federato a più data store e data lake e integrazioni con Amazon, Amazon e altri. SageMaker QuickSight Servizi AWS

Per creare un'integrazione zero-ETL, specifichi un database RDS, un cluster come origine e un data warehouse Amazon Redshift come destinazione. L'integrazione replica i dati dal database di origine nel data warehouse di destinazione.

Il diagramma seguente illustra questa funzionalità.



L'integrazione monitora lo stato della pipeline dei dati ed esegue il ripristino in caso di problemi quando possibile. Puoi creare integrazioni da più database RDS (cluster Aurora) in un unico spazio dei nomi Amazon Redshift, che ti consente di ricavare informazioni su più applicazioni.

Argomenti

Vantaggi

Le integrazioni Zero-ETL di RDS con Amazon Redshift offrono i seguenti vantaggi:

- Ti consentono di ottenere approfondimenti di tipo olistico da più origini dati.
- Eliminano la necessità di creare e gestire pipeline dei dati complesse che eseguono operazioni di estrazione, trasformazione e caricamento (ETL). Le integrazioni Zero-ETL forniscono e gestiscono le pipeline per te, eliminando le sfide legate alla loro creazione e gestione.
- Ti consentono di ridurre il carico e i costi operativi e di concentrarti sul miglioramento delle applicazioni.
- Consenti di sfruttare le funzionalità di analisi e ML di Amazon Redshift per ricavare informazioni dettagliate da dati transazionali e di altro tipo, per rispondere efficacemente a eventi critici e urgenti.

Concetti chiave

Per iniziare a utilizzare le integrazioni Zero-ETL, tieni presente i seguenti concetti:

Integrazione

Una pipeline di dati completamente gestita che replica automaticamente i dati e gli schemi transazionali da un un data warehouse Amazon Redshift.

Il cluster del database RDS da cui vengono replicati i dati. È possibile specificare un'istanza DB Single-AZ o Multi-AZ.

Data warehouse di destinazione

Si tratta del data warehouse di Amazon Redshift in cui viene eseguita la replica dei dati. Esistono due tipi di data warehouse: un data warehouse con cluster con provisioning e un data warehouse serverless. Un data warehouse con cluster con provisioning è costituito da un insieme di risorse di calcolo denominate nodi, strutturate in un gruppo denominato cluster. Un data warehouse serverless è composto da un gruppo di lavoro che archivia le risorse di calcolo e da un spazio dei nomi che ospita gli oggetti e gli utenti del database. Entrambi i data warehouse utilizzano un motore Amazon Redshift e contengono uno o più database.

I scrivere sulla stessa destinazione.

Per ulteriori informazioni sui nodi principali e sui nodi di calcolo, consulta Architettura del sistema di data warehouse nella Guida per sviluppatori di database di Amazon Redshift.

Limitazioni dell'anteprima

Le seguenti limitazioni si applicano alle integrazioni Zero-ETL di RDS con Amazon Redshift.

Argomenti

Limitazioni generali

- Il del database di origine deve trovarsi nella stessa regione del data warehouse Amazon Redshift di destinazione.
- Non puoi rinominare un se dispone di integrazioni esistenti.
- Non è possibile eliminare un esistenti. Devi prima eliminare tutte le integrazioni associate.
-
- Non è possibile eliminare un'integrazione se il database di origine è interrotto.
- Amazon RDS supporta solo implementazioni di istanze DB Single-AZ e Multi-AZ come fonti di integrazione. Attualmente non supporta i cluster DB Multi-AZ.
- Le integrazioni zero-ETL attualmente non supportano il filtraggio dei dati.
- Se il di database è all'origine di una distribuzione blu/verde, gli ambienti blu e verde non possono avere integrazioni zero-ETL esistenti durante lo switchover. Occorre eliminare l'integrazione, eseguire lo switchover e poi ricrearla.
- Non è possibile creare un'integrazione per un database di origine in cui viene creata attivamente un'altra integrazione.
- Durante la fase iniziale della creazione di un'integrazione o quando una tabella viene risincronizzata, il seeding dei dati dall'origine alla destinazione può richiedere 20-25 minuti o più, a seconda delle dimensioni del database di origine. Questo ritardo può portare a un aumento del ritardo di replica.
- Alcuni tipi di dati non sono supportati. Per ulteriori informazioni, consulta the section called “Differenze dei tipi di dati”.
- I riferimenti a chiavi esterne con aggiornamenti di tabella predefiniti non sono supportati. In particolare, `ON DELETE` le `ON UPDATE` regole non sono supportate con `CASCADESET NULL`, e

`SET DEFAULT` le azioni. Se si tenta di creare o aggiornare una tabella con tali riferimenti a un'altra tabella, la tabella entrerà in uno stato di errore.

- `ALTER TABLE` La tabella non sarà disponibile per l'interrogazione durante la risincronizzazione. Per ulteriori informazioni, consulta the section called “Una o più tabelle Amazon Redshift richiedono una risincronizzazione”.
- Le transazioni XA non sono supportate.
- Gli identificatori di oggetto, inclusi il nome del database, il nome della tabella, i nomi delle colonne e altri, possono contenere solo caratteri alfanumerici, numeri, $ e _ (carattere di sottolineatura).

Limitazioni di RDS per MySQL

- Il database di origine deve eseguire RDS for MySQL versione 8.0.32 o successiva.
- Le integrazioni Zero-ETL si basano sui log binari MySQL (binlog) per acquisire le modifiche continue dei dati. Non utilizzare il filtraggio dei dati basato su binlog, poiché può causare incongruenze tra i database di origine e di destinazione.
- Le tabelle di sistema, le tabelle temporanee e le viste di RDS for MySQL non vengono replicate su Amazon Redshift.
- Le integrazioni Zero-ETL sono supportate solo per i database configurati per l'utilizzo del motore di storage InnoDB.
- I cluster DB di origine non possono essere configurati con Certificate Authority (CA). `rds-ca-ecc384-g1`

Limitazioni di Amazon Redshift

Per un elenco delle limitazioni di Amazon Redshift relative alle integrazioni zero-ETL, consulta Considerazioni nella Amazon Redshift Management Guide.

Quote

Sul tuo account sono disponibili le seguenti quote relative alle integrazioni Zero-ETL di RDS con Amazon Redshift. Salvo dove diversamente specificato, ogni quota fa riferimento a una Regione specifica.

Nome	Predefinito	Descrizione
Integrazioni	100	Numero totale di integrazioni all'interno di un Account AWS.
Integrazioni per data warehouse di destinazione	50	Numero di integrazioni che inviano dati a un unico data warehouse Amazon Redshift di destinazione.
Integrazioni per istanza di origine	1	

Inoltre, Amazon Redshift pone determinati limiti al numero di tabelle consentite in ogni istanza database o nodo del cluster. Per ulteriori informazioni, consulta Quote e limiti in Amazon Redshift nella Guida alla gestione di Amazon Redshift.

Regioni supportate

Le integrazioni Zero-ETL di RDS con Amazon Redshift sono disponibili in un sottoinsieme di. Regioni AWS Per un elenco delle regioni supportate, consultare the section called “Integrazioni Zero-ETL ”.

Guida introduttiva alle integrazioni Zero-ETL di Amazon RDS con Amazon Redshift

Questa è la documentazione preliminare per le integrazioni Zero-ETL di Amazon RDS con Amazon Redshift, che è in versione di anteprima. La documentazione e la funzionalità sono soggette a modifiche. Consigliamo di utilizzare questa caratteristica solo in ambienti di test e non in ambienti di produzione. Per i termini e condizioni dell'anteprima, consulta la sezione su beta e anteprime nei AWS termini del servizio.

Prima di creare un'integrazione zero-ETL con Amazon Redshift, configura il database RDS, il cluster Aurora le autorizzazioni richiesti. Durante la configurazione, dovrai completare i seguenti passaggi:

1. Creazione di un gruppo di parametri per il personalizzato..

2. un cluster DB di database di origine.
3. Creazione di un data warehouse Amazon Redshift di destinazione.

Dopo aver completato questi passaggi, passa alla sezione the section called “Creazione di integrazioni Zero-ETL”.

Fase 1: creazione di un gruppo di parametri del DB personalizzato

Le integrazioni Zero-ETL di Amazon RDS con Amazon Redshift richiedono valori specifici per i parametri DB che controllano la registrazione binaria (binlog). Per configurare la registrazione binaria, devi prima creare un gruppo di parametri DB personalizzato e quindi associarlo al database di origine.

Crea un gruppo di parametri DB personalizzato con le seguenti impostazioni . Per istruzioni sulla creazione di un gruppo di parametri, consulta the section called “Utilizzo di gruppi di parametri di database”.

- `binlog_format=ROW`
- `binlog_row_image=full`
- `binlog_checksum=NONE`

Inoltre, assicurati che il parametro `binlog_row_value_options` non sia impostato su `PARTIAL_JSON`.

Passaggio 2: selezionare o creare un cluster del database di origine

Dopo aver creato un gruppo di parametri del DB personalizzato, scegli o crea un cluster RDS per l'istanza database MySQL Single-AZ o Multi-AZ Aurora MySQL o Aurora DB. Questo di database sarà l'origine della replica dei dati su Amazon Redshift.

Il di database deve eseguire RDS for MySQL versione 8.0.32 o successiva, Aurora MySQL versione 3.05 (compatibile con MySQL 8.0.32) o successiva o . Per istruzioni su come creare un

In Configurazione aggiuntiva, modifica il gruppo di parametri predefinito del DB con il gruppo di parametri personalizzato creato nel passaggio precedente.

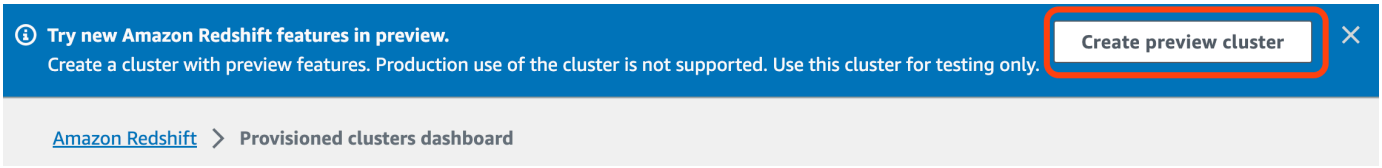
> **Note**
>
> Per istruzioni, consulta the section called “Riavvio di un'istanza database”.

Inoltre, assicurati che i backup automatici siano abilitati sul database. Per ulteriori informazioni, consulta the section called “Abilitazione dei backup automatici”.

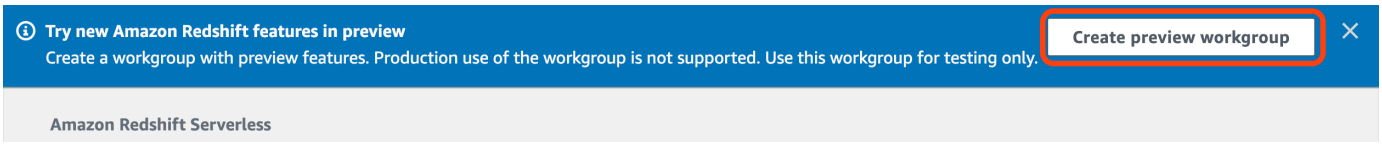
Fase 3: creazione di un data warehouse Amazon Redshift di destinazione

Dopo aver creato il del database di origine, devi creare e configurare un data warehouse di destinazione in Amazon Redshift. Il data warehouse deve soddisfare i seguenti requisiti:

- Creato in anteprima
  - Per creare un'anteprima del cluster con provisioning, scegli Crea cluster di anteprima dal banner sulla dashboard dei cluster con provisioning. Per ulteriori informazioni, consulta Creazione di un cluster di anteprima.



    Quando crei il cluster, imposta Traccia anteprima su `preview_2023`.
  - Per creare un'anteprima del gruppo di lavoro Redshift Serverless, scegli Crea gruppo di lavoro di anteprima dal banner sulla dashboard Serverless. Per ulteriori informazioni, consulta Creazione di un gruppo di lavoro di anteprima.



- Utilizzando un tipo di nodo RA3 (`ra3.xlplusra3.4xlarge`, `ora3.16xlarge`) con almeno due nodi o Redshift Serverless.
- Deve essere crittografato (se si utilizza un cluster con provisioning). Per ulteriori informazioni, consulta Crittografia dei database di Amazon Redshift.

Per istruzioni su come creare un data warehouse, consulta Creazione di un cluster per i cluster con provisioning o Creazione di un gruppo di lavoro con uno spazio dei nomi per Redshift Serverless.

Abilitazione della distinzione tra maiuscole e minuscole nel data warehouse

Affinché l'integrazione venga eseguita correttamente, il parametro di distinzione tra maiuscole e minuscole (`enable_case_sensitive_identifier`) deve essere abilitato per il data warehouse. Per impostazione predefinita, la distinzione tra maiuscole e minuscole è disabilitata su tutti i cluster con provisioning e sui gruppi di lavoro Redshift serverless.

Per abilitare la distinzione tra maiuscole e minuscole, esegui i seguenti passaggi a seconda del tipo di data warehouse:

- Cluster con provisioning: per abilitare la distinzione tra maiuscole e minuscole su un cluster con provisioning, crea un gruppo di parametri personalizzato con il parametro `enable_case_sensitive_identifier` abilitato. Poi, associa il gruppo di parametri al cluster. Per istruzioni, consulta Gestione di gruppi di parametri mediante la console o Configurazione dei valori di parametro mediante AWS CLI.

  > **Note**
  >
  > Ricordati di riavviare il cluster dopo aver associato il gruppo di parametri personalizzati.

- Gruppo di lavoro serverless: per abilitare la distinzione tra maiuscole e minuscole su un gruppo di lavoro SRedshift Serverless, è necessario utilizzare la AWS CLI. La console Amazon Redshift attualmente non supporta la modifica dei valori dei parametri Redshift Serverless. Invia la seguente richiesta di aggiornamento al gruppo di lavoro:

  ```
  aws redshift-serverless update-workgroup \
    --workgroup-name target-workgroup \
    --config-parameters
   parameterKey=enable_case_sensitive_identifier,parameterValue=true
  ```

  Non è necessario riavviare un gruppo di lavoro dopo aver modificato i valori dei parametri.

Configura l'autorizzazione per il data warehouse

Dopo aver creato un data warehouse, è necessario configurare il database RDS di origine cluster come fonte di integrazione autorizzata. Per istruzioni, consulta Configurazione dell'autorizzazione per il data warehouse Amazon Redshift.

Passaggi successivi

Con un database RDS di origine, un cluster e un data warehouse di destinazione Amazon Redshift, ora puoi creare un'integrazione zero-ETL e replicare i dati. Per istruzioni, consultare the section called “Creazione di integrazioni Zero-ETL”.

Creazione di integrazioni Zero-ETL di Amazon RDS con Amazon Redshift

Questa è la documentazione preliminare per le integrazioni Zero-ETL di Amazon RDS con Amazon Redshift, che è in versione di anteprima. La documentazione e la funzionalità sono soggette a modifiche. Consigliamo di utilizzare questa caratteristica solo in ambienti di test e non in ambienti di produzione. Per i termini e condizioni dell'anteprima, consulta la sezione su beta e anteprime nei AWS termini del servizio.

Quando crei un'integrazione Zero-ETL di Amazon RDS , specifichi l'istanza DB RDS Single-AZ o Multi-AZ di origine, il cluster e il data warehouse Amazon Redshift di destinazione. Puoi anche personalizzare le impostazioni di crittografia e aggiungere tag. Amazon RDS crea un'integrazione tra il cluster di origine e la sua destinazione. Una volta che l'integrazione è attiva, tutti i dati inseriti nel del database di origine verranno replicati nel target Amazon Redshift configurato.

Argomenti

Prerequisiti

Prima di creare un'integrazione zero-ETL, devi creare un di database di origine e un data warehouse Amazon Redshift di destinazione. È inoltre necessario consentire la replica nel data warehouse aggiungendo il del database come fonte di integrazione autorizzata.

Per istruzioni su come completare ciascuno di questi passaggi, consulta the section called “Guida introduttiva alle integrazioni Zero-ETL”.

Autorizzazioni richieste

Per creare un'integrazione Zero-ETL, occorrono alcune autorizzazioni IAM. Come requisito minimo, dovrai disporre delle autorizzazioni per eseguire le seguenti operazioni:

-
- Visualizzazione ed eliminazione di tutte le integrazioni Zero-ETL.
- Creazione di integrazioni in entrata nel data warehouse di destinazione. Non hai bisogno di questa autorizzazione se lo stesso account possiede il data warehouse Amazon Redshift e questo account è il principale autorizzato di tale data warehouse. Per informazioni sull'aggiunta di principali autorizzati, consulta Configurazione dell'autorizzazione per il data warehouse Amazon Redshift.

La seguente policy di esempio mostra le autorizzazioni con privilegi minimi richieste per creare e gestire le integrazioni. Potresti non aver bisogno di queste autorizzazioni esatte se il tuo utente o ruolo dispone di autorizzazioni più ampie, come una policy gestita. `AdministratorAccess`

> **Note**
>
> Gli ARN (Amazon Resource Names) di Redshift hanno il seguente formato. Nota l'uso di una barra (`/`) anziché dei due punti (`:`) prima dell'UUID dello spazio dei nomi serverless.
>
> - Cluster con provisioning: `arn:aws:redshift:{region}:{account-id}:namespace:namespace-uuid`
> - Serverless: `arn:aws:redshift-serverless:{region}:{account-id}:namespace/namespace-uuid`

Policy di esempio

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Action": [
            "rds:CreateIntegration"
        ],
        "Resource": [
            "arn:aws:rds:{region}:{account-id}:db:source-db",
            "arn:aws:rds:{region}:{account-id}:integration:*"
```

```
        ]
    },
    {
        "Effect": "Allow",
        "Action": [
            "rds:DescribeIntegrations"
        ],
        "Resource": ["*"]
    },
    {
        "Effect": "Allow",
        "Action": [
            "rds:DeleteIntegration"
        ],
        "Resource": [
            "arn:aws:rds:{region}:{account-id}:integration:*"
        ]
    },
    {
        "Effect": "Allow",
        "Action": [
            "redshift:CreateInboundIntegration"
        ],
        "Resource": [
            "arn:aws:redshift:{region}:{account-id}:namespace:namespace-uuid"
        ]
    }]
}
```

Scelta di un data warehouse di destinazione in un account diverso

Se prevedi di specificare un data warehouse Amazon Redshift di destinazione che si trova in un altro Account AWS, devi creare un ruolo che consenta agli utenti dell'account corrente di accedere alle risorse nell'account di destinazione. Per ulteriori informazioni, consulta Fornire l'accesso a un utente IAM in un altro utente Account AWS di tua proprietà.

Il ruolo deve disporre delle seguenti autorizzazioni, che consentono all'utente di visualizzare i cluster Amazon Redshift con provisioning e gli spazi dei nomi Redshift Serverless disponibili nell'account di destinazione.

Autorizzazioni richieste e policy di attendibilità

```
{
```

```
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "redshift:DescribeClusters",
            "redshift-serverless:ListNamespaces"
         ],
         "Resource":[
            "*"
         ]
      }
   ]
}
```

Il ruolo deve avere la seguente policy di attendibilità, che specifica l'ID dell'account di destinazione.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "AWS": "arn:aws:iam::{external-account-id}:root"
         },
         "Action":"sts:AssumeRole"
      }
   ]
}
```

Per istruzioni sulla creazione del ruolo, consulta Creazione di un ruolo utilizzando policy di attendibilità personalizzate.

Creazione di integrazioni Zero-ETL

È possibile creare un'integrazione zero-ETL un'integrazione utilizzando l'API, the o RDS. AWS Management Console AWS CLI

Per impostazione predefinita, RDS per MySQL elimina immediatamente i file di log binari. Poiché le integrazioni zero-ETL si basano sui log binari per replicare i dati dall'origine alla destinazione, il periodo di conservazione per il database di origine deve essere di almeno un'ora. Non appena crei

un'integrazione, Amazon RDS verifica il periodo di conservazione dei file di log binari per il database di origine selezionato. Se il valore corrente è 0 ore, Amazon RDS lo modifica automaticamente in 1 ora. In caso contrario, il valore rimane invariato.

Console RDS

Creazione di un'integrazione Zero-ETL

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel pannello di navigazione a sinistra, scegli Interfacce di rete.
3. Scegli Crea un'integrazione Zero-ETL.
4. In Identificatore dell'integrazione, inserisci un nome per l'integrazione. Il nome può contenere fino a 63 caratteri alfanumerici e può includere trattini.
5. Seleziona Successivo.
6. Per Origine, seleziona il cluster del database RDS da cui provengono i dati. Il di database deve eseguire RDS for MySQL versione 8.0.32 o successiva, Aurora MySQL versione 3.05 o successiva o).

   > **Note**
   >
   > , RDS ti avvisa se i parametri del cluster non sono configurati correttamente. Se ricevi questo messaggio, puoi scegliere Correggi per me o configurarli manualmente. Per istruzioni su come correggerli manualmente, consulta the section called “Fase 1: creazione di un gruppo di parametri del DB personalizzato”.
   > La modifica dei parametri del DB richiede un riavvio.

7. Una volta configurato correttamente il database di origine, scegli Avanti.
8. Per Destinazione, esegui queste operazioni:

   1. (Facoltativo) Per utilizzare un account diverso Account AWS per il target Amazon Redshift, scegli Specificare un account diverso. Quindi, inserisci l'ARN di un ruolo IAM con le autorizzazioni per visualizzare i data warehouse. Per istruzioni su come creare il ruolo IAM, consulta the section called “Scelta di un data warehouse di destinazione in un account diverso”.
   2. Puoi scegliere un cluster Amazon Redshift con provisioning o uno spazio dei nomi Redshift Serverless come destinazione.

> **Note**
>
> RDS ti avvisa se le policy relative alle risorse o le impostazioni di distinzione tra maiuscole e minuscole per il data warehouse specificato non sono configurate correttamente. Se ricevi questo messaggio, puoi scegliere Correggi per me o configurarli manualmente. Per istruzioni su come correggerli manualmente, consulta Attivazione della distinzione tra maiuscole e minuscole per il data warehouse e Configurazione dell'autorizzazione per il data warehouse nella Guida alla gestione di Amazon Redshift. La modifica della distinzione tra maiuscole e minuscole per un cluster Redshift con provisioning richiede un riavvio. Prima di poter creare l'integrazione, è necessario completare il riavvio e applicare correttamente il nuovo valore del parametro al cluster. Se l'origine e la destinazione selezionate si trovano in Account AWS diversi, Amazon RDS non può correggere queste impostazioni per te. Devi accedere all'altro account e correggerle manualmente in Amazon Redshift.

9. Una volta configurato correttamente il data warehouse di destinazione, scegli Avanti.
10. (Facoltativo) In Tag, aggiungi uno o più tag all'integrazione. Per ulteriori informazioni, consulta the section called “Tagging delle risorse RDS”.
11. In Crittografia, specifica come eseguire la crittografia dell'integrazione. Per impostazione predefinita, RDS crittografa tutte le integrazioni con un. Chiave di proprietà di AWS Per scegliere invece una chiave gestita dal cliente, abilita Personalizza le impostazioni di crittografia e scegli una chiave KMS da utilizzare per la crittografia. Per ulteriori informazioni, consulta the section called “Crittografia delle risorse Amazon RDS”.

> **Note**
>
> Se specifichi una chiave KMS personalizzata, la policy della chiave deve consentire l'operazione `kms:CreateGrant` per il principale del servizio Amazon Redshift (`redshift.amazonaws.com`). Per ulteriori informazioni, consulta Creazione di una policy delle chiavi nella Guida per gli sviluppatori di AWS Key Management Service .

Aggiungi un contesto di crittografia (facoltativo). Per ulteriori informazioni, consultare Contesto della crittografia nella Guida per gli sviluppatori di AWS Key Management Service .

12. Seleziona Successivo.

13. Rivedi le impostazioni dell'integrazione e scegli Crea un'integrazione Zero-ETL.

    Se la creazione ha esito negativo, consulta the section called “Non riesco a creare un'integrazione Zero-ETL” per la procedura di risoluzione dei problemi.

Lo stato dell'integrazione è `Creating` mentre l'integrazione è in fase di creazione, mentre lo stato del data warehouse Amazon Redshift di destinazione è `Modifying`. Durante questo periodo, non puoi eseguire query sul data warehouse o apportare modifiche alla configurazione.

Quando la creazione dell'integrazione viene completata correttamente, lo stato dell'integrazione e del data warehouse Amazon Redshift di destinazione cambia in `Active`.

AWS CLI

Per creare un'integrazione zero-ETL utilizzando il AWS CLI, usa il comando create-integration con le seguenti opzioni:

- `--integration-name`: specifica un nome per l'integrazione.
- `--source-arn`— Specificare l'ARN del del database RDS che sarà l'origine dell'integrazione.
- `--target-arn`: specifica l'ARN del data warehouse di Amazon Redshift che sarà la destinazione dell'integrazione.

Example

PerLinux, o: macOS Unix

```
aws rds create-integration \
    --integration-name my-integration \
    --source-arn arn:aws:rds:{region}:{account-id}:my-cluster \
    --target-arn arn:aws:redshift:{region}:{account-id}:namespace:namespace-uuid
```

Per Windows:

```
aws rds create-integration ^
    --integration-name my-integration ^
    --source-arn arn:aws:rds:{region}:{account-id}:my-cluster ^
    --target-arn arn:aws:redshift:{region}:{account-id}:namespace:namespace-uuid
```

API RDS

Per creare un'integrazione Zero-ETL con l'API Amazon RDS, utilizza l'operazione `CreateIntegration` con i seguenti parametri:

- `IntegrationName`: specifica un nome per l'integrazione.
- `SourceArn`— Specificare l'ARN del cluster per l'integrazione.
- `TargetArn`: specifica l'ARN del data warehouse di Amazon Redshift che sarà la destinazione dell'integrazione.

Passaggi successivi

Dopo aver creato correttamente un'integrazione Zero-ETL, devi creare un database di destinazione all'interno del cluster o del gruppo di lavoro Amazon Redshift di destinazione. Quindi, puoi iniziare ad aggiungere dati al cluster del database RDS di origine e interrogarli in Amazon Redshift. Per istruzioni, consulta Creazione di database di destinazione in Amazon Redshift.

Aggiungere dati a un database RDS di origine (cluster) e interrogarli in Amazon Redshift

Questa è la documentazione preliminare per le integrazioni Zero-ETL di Amazon RDS con Amazon Redshift, che è in versione di anteprima. La documentazione e la funzionalità sono soggette a modifiche. Consigliamo di utilizzare questa caratteristica solo in ambienti di test e non in ambienti di produzione. Per i termini e condizioni dell'anteprima, consulta la sezione su beta e anteprime nei AWS termini del servizio.

Per completare la creazione di un'integrazione Zero-ETL che replichi i dati da Amazon RDS in Amazon Redshift, devi creare un database di destinazione in Amazon Redshift.

Innanzitutto, connettiti al cluster o al gruppo di lavoro Amazon Redshift e crea un database con un riferimento al tuo identificatore di integrazione. Quindi, puoi aggiungere dati al cluster del database RDS di origine e vederli replicati in Amazon Redshift.

Argomenti

- Creazione di un database di destinazione in Amazon Redshift

-
- Interrogazione dei dati di Amazon RDS in Amazon Redshift
- Differenze tra i tipi di dati tra i database RDS e Amazon Redshift

Creazione di un database di destinazione in Amazon Redshift

Prima di poter iniziare a replicare i dati in Amazon Redshift dopo la creazione di un'integrazione, devi creare un database di destinazione nel data warehouse di destinazione. Questo database di destinazione deve includere un riferimento all'identificatore di integrazione. Puoi utilizzare la console Amazon Redshift o Editor di query v2 per creare il database.

Per istruzioni sulla creazione di un database di destinazione, consulta Creazione di un database di destinazione in Amazon Redshift.

Dopo aver configurato l'integrazione, puoi aggiungere alcuni dati al cluster del database RDS che desideri replicare nel tuo data warehouse Amazon Redshift.

> **Note**
>
> Esistono differenze tra i tipi di dati in Amazon RDS e Amazon Redshift. Per una tabella di mappature dei tipi di dati, consulta the section called “Differenze dei tipi di dati”.

Innanzitutto, connettiti al del database di origine utilizzando il client di tua scelta. Per istruzioni, consulta the section called “Connessione a un'istanza database che esegue MySQL”.

Quindi, crea una tabella e inserisci una riga di dati di esempio.

> **Important**
>
> Assicurati che la tabella abbia una chiave primaria. Altrimenti, non può essere replicata nel data warehouse di destinazione.

L'esempio seguente utilizza l'utilità MySQL Workbench.

```
CREATE DATABASE my_db;
```

```
USE my_db;

CREATE TABLE books_table (ID int NOT NULL, Title VARCHAR(50) NOT NULL, Author
 VARCHAR(50) NOT NULL,
Copyright INT NOT NULL, Genre VARCHAR(50) NOT NULL, PRIMARY KEY (ID));

INSERT INTO books_table VALUES (1, 'The Shining', 'Stephen King', 1977, 'Supernatural
 fiction');
```

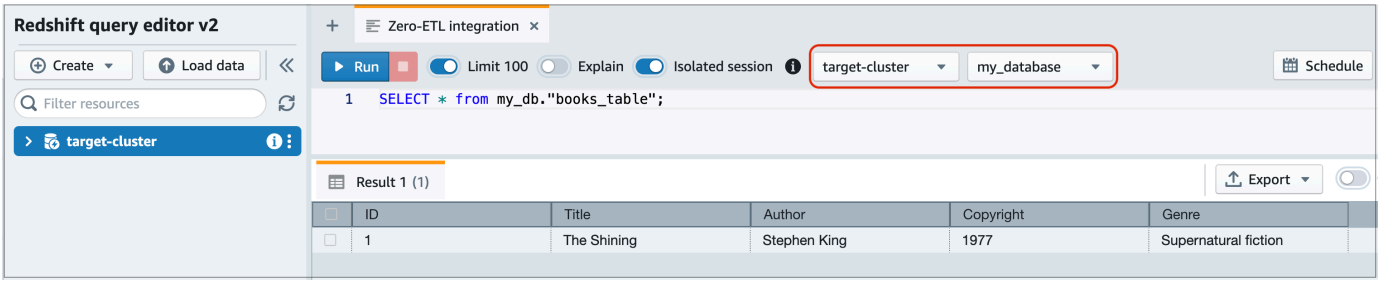
Interrogazione dei dati di Amazon RDS in Amazon Redshift

Dopo aver aggiunto i dati al cluster del database RDS, questi vengono replicati in Amazon Redshift e sono pronti per essere interrogati.

Esecuzione di query sui dati replicati

1. Vai alla console Amazon Redshift e scegli Editor di query v2 nel riquadro di navigazione a sinistra.
2. Connettiti al cluster o al gruppo di lavoro e scegli il database di destinazione (creato dall'integrazione) dal menu a tendina (destination_database in questo esempio). Per istruzioni sulla creazione di un database di destinazione, consulta Creazione di un database di destinazione in Amazon Redshift.
3. Usa un'istruzione SELECT per interrogare i tuoi dati. In questo esempio, è possibile eseguire il comando seguente per selezionare tutti i dati dalla tabella creata nel database RDS di origine del cluster :

   ```
   SELECT * from my_db."books_table";
   ```



   - *my_db* è il nome dello schema del database RDS.
   - *books_table* è il nome della tabella RDS.

È inoltre possibile interrogare i dati utilizzando un client a riga di comando. Per esempio:

```
destination_database=# select * from my_db."books_table";

 ID |       Title |        Author |   Copyright |                 Genre |  txn_seq |
 txn_id
----+-------------+---------------+-------------+-----------------------+----------
+--------+
  1 | The Shining |  Stephen King |        1977 |   Supernatural fiction |        2 |
 12192
```

> **Note**
>
> Per distinguere tra maiuscole e minuscole, usa le virgolette doppie (" ") per i nomi di schemi, tabelle e colonne. Per ulteriori informazioni, consulta enable_case_sensitive_identifier.

Differenze tra i tipi di dati tra i database RDS e Amazon Redshift

La tabella seguente mostra la mappatura di un RDS per MySQL. Le tabelle mostrano le mappature di un tipo di dati Aurora MySQL tipo di dati Amazon Redshift corrispondente. Amazon RDS Aurora attualmente supporta solo questi tipi di dati per integrazioni zero-ETL.

Se una tabella nel del database di origine include un tipo di dati non supportato, la tabella non è sincronizzata e non è utilizzabile dal target Amazon Redshift. Lo streaming dall'origine alla destinazione va avanti, ma la tabella con il tipo di dati non supportato non è disponibile. Per correggere la tabella e renderla disponibile in Amazon Redshift, devi annullare manualmente la modifica iniziale e aggiornare l'integrazione eseguendo `ALTER DATABASE...INTEGRATION REFRESH`.

Tipo di dati RDS per MySQL	Tipo di dati di Amazon Redshift	Descrizione	Limitazioni
INT	INTEGER	Intero a quattro byte firmato	
SMALLINT	SMALLINT	Intero a due byte firmato	

Tipo di dati RDS per MySQL	Tipo di dati di Amazon Redshift	Descrizione	Limitazioni
TINYINT	SMALLINT	Intero a due byte firmato	
MEDIUMINT	INTEGER	Intero a quattro byte firmato	
BIGINT	BIGINT	Intero a otto byte firmato	
INT UNSIGNED	BIGINT	Intero a otto byte firmato	
TINYINT UNSIGNED	SMALLINT	Intero a due byte firmato	
MEDIUMINT UNSIGNED	INTEGER	Intero a quattro byte firmato	
BIGINT UNSIGNED	DECIMAL(20,0)	Numerico esatto di precisione selezionabile	
DECIMALE (p, s) = NUMERICO (p, s)	DECIMAL(p,s)	Numerico esatto di precisione selezionabile	Precisione superiore a 38 e scala superiore a 37 non supportat e
DECIMALE (p, s) SENZA SEGNO = NUMERICO (p, s) SENZA SEGNO	DECIMAL(p,s)	Numerico esatto di precisione selezionabile	Precisione superiore a 38 e scala superiore a 37 non supportat e

Tipo di dati RDS per MySQL	Tipo di dati di Amazon Redshift	Descrizione	Limitazioni
FLOAT4/REAL	REAL	Numero in virgola mobile a precisione singola	
FLOAT4/REAL UNSIGNED	REAL	Numero in virgola mobile a precisione singola	
DOUBLE/REAL/FLOAT8	DOUBLE PRECISION	Numero in virgola mobile a precisione doppia	
DOUBLE/REAL/FLOAT8 UNSIGNED	DOUBLE PRECISION	Numero in virgola mobile a precisione doppia	
BIT (n)	BARBYTE(8)	Valore binario a lunghezza variabile	
BINARY(n)	BARBYTE (n)	Valore binario a lunghezza variabile	
VARBINARY(n)	VARBYTE (n)	Valore binario a lunghezza variabile	
CHAR(n)	VARCHAR(n)	Valore di stringa di lunghezza variabile	

Tipo di dati RDS per MySQL	Tipo di dati di Amazon Redshift	Descrizione	Limitazioni
VARCHAR(n)	VARCHAR(n)	Valore di stringa di lunghezza variabile	
TEXT	VARCHAR(65535)	Valore di stringa di lunghezza variabile fino a 65535 byte	
TINYTEXT	VARCHAR(255)	Valore di stringa di lunghezza variabile fino a 255 byte	
ENUM	VARCHAR(1020)	Valore di stringa di lunghezza variabile fino a 1020 byte	
SET	VARCHAR(1020)	Valore di stringa di lunghezza variabile fino a 1020 byte	
DATE	DATE	Data di calendario (anno, mese, giorno)	
DATETIME	TIMESTAMP	Data e ora (senza fuso orario)	
TIMESTAMP(p)	TIMESTAMP	Data e ora (senza fuso orario)	

Tipo di dati RDS per MySQL	Tipo di dati di Amazon Redshift	Descrizione	Limitazioni
TIME	VARCHAR(18)	Valore di stringa di lunghezza variabile fino a 18 byte	
ANNO	VARCHAR(4)	Valore di stringa di lunghezza variabile fino a 4 byte	
JSON	SUPER	Dati o documenti semistrutturati come valori	

Visualizzazione e monitoraggio delle integrazioni Zero-ETL di Amazon RDS con Amazon Redshift

Questa è la documentazione preliminare per le integrazioni Zero-ETL di Amazon RDS con Amazon Redshift, che è in versione di anteprima. La documentazione e la funzionalità sono soggette a modifiche. Consigliamo di utilizzare questa caratteristica solo in ambienti di test e non in ambienti di produzione. Per i termini e condizioni dell'anteprima, consulta la sezione su beta e anteprime nei AWS termini del servizio.

Puoi visualizzare i dettagli di un'integrazione Zero-ETL di Amazon RDS per visualizzarne le informazioni di configurazione e lo stato attuale. Puoi anche monitorare lo stato della tua integrazione eseguendo query sulle viste di sistema specifiche in Amazon Redshift. Inoltre, Amazon Redshift pubblica alcuni parametri relativi all'integrazione su Amazon, che puoi visualizzare all'interno della CloudWatch console Amazon Redshift.

Argomenti

- Visualizzazione delle integrazioni

- Monitoraggio delle integrazioni tramite tabelle di sistema
- Monitoraggio delle integrazioni con Amazon EventBridge

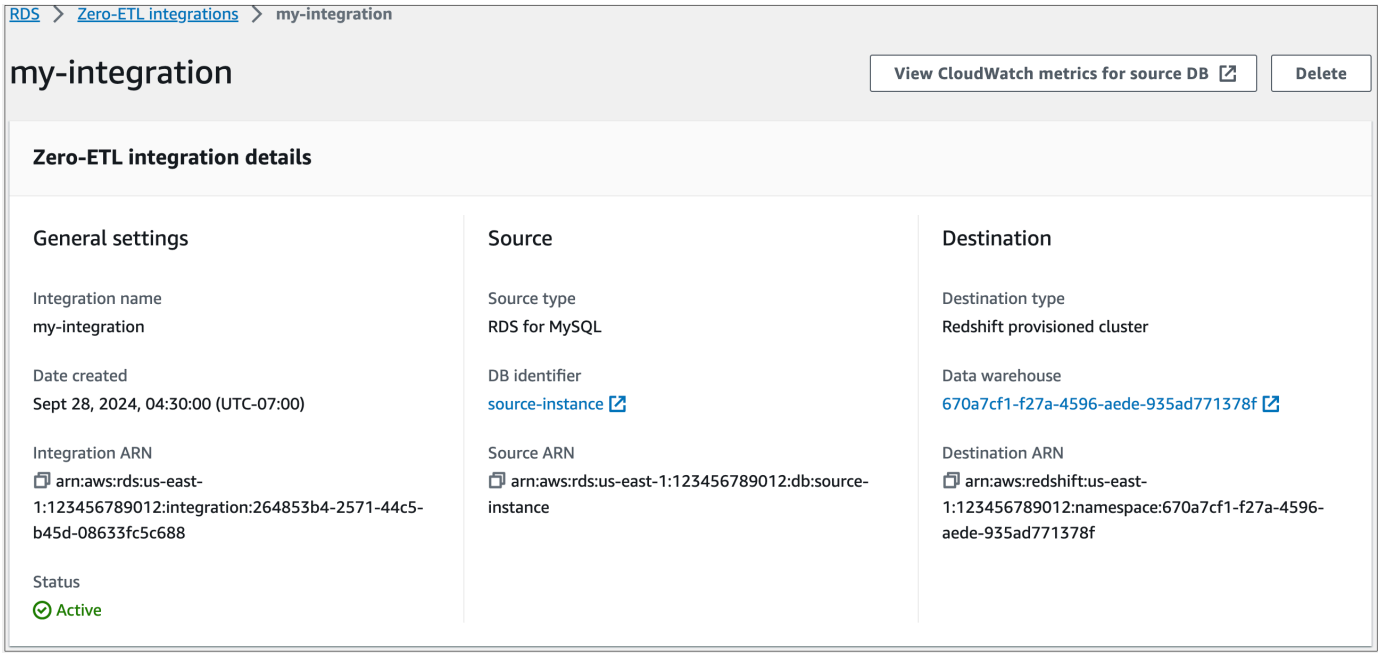
Visualizzazione delle integrazioni

Puoi visualizzare le integrazioni Zero-ETL di Amazon RDS con Amazon Redshift AWS Management Console utilizzando l'API RDS o la. AWS CLI

Console

Visualizzazione dei dettagli di un'integrazione Zero-ETL

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione a sinistra, scegli Integrazioni Zero-ETL.
3. Seleziona un'integrazione per visualizzarne maggiori dettagli, ad esempio il database di origine, il e il data warehouse di destinazione.



Un'integrazione può avere i seguenti stati:

- `Creating`: l'integrazione è in fase di creazione.
- `Active`: l'integrazione sta inviando dati transazionali al data warehouse di destinazione.

- `Syncing`: l'integrazione ha rilevato un errore recuperabile e deve reimpostare i dati. Le tabelle interessate non sono disponibili per l'interrogazione in Amazon Redshift fino al termine della risincronizzazione.
- `Needs attention`: l'integrazione ha rilevato un evento o un errore che richiede un intervento manuale per la risoluzione. Per correggere il problema, segui le istruzioni nel messaggio di errore nella pagina dei dettagli dell'integrazione.
- `Failed`: l'integrazione ha rilevato un evento o un errore irreversibile che non può essere risolto. È necessario eliminare e ricreare l'integrazione.
- `Deleting`: l'integrazione è in fase di eliminazione.

AWS CLI

Per visualizzare tutte le integrazioni zero-ETL nell'account corrente utilizzando il, usa il comando describe-integrations e specifica AWS CLI l'opzione. `--integration-identifier`

Example

LinuxPermacOS, o: Unix

```
aws rds describe-integrations \
    --integration-identifier ee605691-6c47-48e8-8622-83f99b1af374
```

Per Windows:

```
aws rds describe-integrations ^
    --integration-identifier ee605691-6c47-48e8-8622-83f99b1af374
```

API RDS

Per visualizzare l'integrazione Zero-ETL utilizzando l'API Amazon RDS, usa l'operazione `DescribeIntegrations` con il parametro `IntegrationIdentifier`.

Monitoraggio delle integrazioni tramite tabelle di sistema

Amazon Redshift include tabelle e viste di sistema che contengono informazioni sul funzionamento del sistema. Puoi eseguire delle query su queste tabelle e viste esattamente come faresti con qualsiasi altra tabella di database. Per ulteriori informazioni sulle tabelle e viste di sistema in Amazon

Redshift, consulta Riferimento di tabelle e viste di sistema nella Guida per sviluppatori di database di Amazon Redshift.

Puoi interrogare le seguenti visualizzazioni e tabelle di sistema per ottenere informazioni sulle integrazioni zero-ETL con Amazon Redshift:

- SVV_INTEGRATION: fornisce i dettagli relativi alla configurazione delle integrazioni.
- SVV_INTEGRATION_TABLE_STATE: descrive lo stato di ogni tabella all'interno di un'integrazione.
- SYS_INTEGRATION_TABLE_STATE_CHANGE: visualizza i log delle modifiche dello stato della tabella per un'integrazione.
- SYS_INTEGRATION_ACTIVITY: fornisce informazioni sulle esecuzioni completate delle integrazioni.

Tutte le metriche Amazon relative all'integrazione provengono da CloudWatch Amazon Redshift. Per ulteriori informazioni, consulta Monitoraggio delle integrazioni Zero-ETL nella Guida alla gestione di Amazon Redshift. Attualmente, Amazon RDS Aurora non pubblica alcuna metrica di integrazione su. CloudWatch

Monitoraggio delle integrazioni con Amazon EventBridge

Amazon Redshift invia eventi relativi all'integrazione ad Amazon. EventBridge Per un elenco di eventi e i relativi ID evento, consulta Notifiche degli eventi di integrazione Zero-ETL con Amazon EventBridge nella Amazon Redshift Management Guide.

Eliminazione delle integrazioni Zero-ETL di Amazon RDS con Amazon Redshift

Questa è la documentazione preliminare per le integrazioni Zero-ETL di Amazon RDS con Amazon Redshift, che è in versione di anteprima. La documentazione e la funzionalità sono soggette a modifiche. Consigliamo di utilizzare questa caratteristica solo in ambienti di test e non in ambienti di produzione. Per i termini e condizioni dell'anteprima, consulta la sezione su beta e anteprime nei AWS termini del servizio.

I dati transazionali non vengono eliminati da Amazon RDS o Amazon Redshift, ma Amazon RDS non invia nuovi dati ad Amazon Redshift.

Puoi eliminare un'integrazione solo quando ha lo stato di,, o. `Active` `Failed` `Syncing` `Needs attention`

Puoi eliminare le integrazioni zero-ETL utilizzando l'API AWS Management Console AWS CLI, the o RDS.

Console

Eliminazione di un'integrazione Zero-ETL

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione a sinistra, scegli Integrazioni Zero-ETL.
3. Seleziona l'integrazione Zero-ETL che desideri eliminare.
4. Scegli Operazioni, Elimina dominio, quindi conferma l'eliminazione.

AWS CLI

Per eliminare un'integrazione Zero-ETL, usa il comando delete-integration e specifica l'opzione `--integration-identifier`.

Example

PerLinux, omacOS: Unix

```
aws rds delete-integration \
    --integration-identifier ee605691-6c47-48e8-8622-83f99b1af374
```

Per Windows:

```
aws rds delete-integration ^
    --integration-identifier ee605691-6c47-48e8-8622-83f99b1af374
```

API RDS

Per eliminare un'integrazione Zero-ETL utilizzando l'API Amazon RDS, usa l'operazione `DeleteIntegration` con il parametro `IntegrationIdentifier`.

Risoluzione dei problemi delle integrazioni Zero-ETL di Amazon RDS con Amazon Redshift

Questa è la documentazione preliminare per le integrazioni Zero-ETL di Amazon RDS con Amazon Redshift, che è in versione di anteprima. La documentazione e la funzionalità sono soggette a modifiche. Consigliamo di utilizzare questa caratteristica solo in ambienti di test e non in ambienti di produzione. Per i termini e condizioni dell'anteprima, consulta la sezione su beta e anteprime nei AWS termini del servizio.

Puoi verificare lo stato di un'integrazione Zero-ETL eseguendo query sulla tabella di sistema SVV_INTEGRATION in Amazon Redshift. Se la colonna `state` include il valore `ErrorState`, significa che si è verificato un problema. Per ulteriori informazioni, consulta the section called “Monitoraggio tramite tabelle di sistema”.

Usa le seguenti informazioni per risolvere i problemi più comuni con le integrazioni Zero-ETL di Amazon RDS con Amazon Redshift.

Argomenti

- Non riesco a creare un'integrazione Zero-ETL
- La mia integrazione è bloccata in uno stato di Syncing
- Le mie tabelle non si replicano su Amazon Redshift
- Una o più tabelle Amazon Redshift richiedono una risincronizzazione

Non riesco a creare un'integrazione Zero-ETL

Se non riesci a creare un'integrazione Zero-ETL, assicurati che quanto segue sia corretto per l'istanza database di origine:

- Il del database di origine esegue RDS for MySQL versione 8.0.32 o successiva, Aurora MySQL versione 3.05 (compatibile con MySQL 8.0.32) o successiva o . Per convalidare la versione del motore scegli la scheda Configurazione per il cluster DB del database e controlla la versione del motore.
- I parametri del di database sono stati configurati correttamente. Se i parametri richiesti sono impostati in modo errato o non sono associati all'istanza database, la creazione ha esito negativo.

Per informazioni, consulta the section called “Fase 1: creazione di un gruppo di parametri del DB personalizzato”.

Inoltre, assicurati che quanto segue sia corretto per il data warehouse di destinazione:

- È abilitata la distinzione tra maiuscole e minuscole. Consulta Attivazione della distinzione tra maiuscole e minuscole per il data warehouse.
- Hai aggiunto il principale autorizzato e l'origine dell'integrazione corretti. Vedi Configurare l'autorizzazione per il tuo data warehouse Amazon Redshift.
- Il data warehouse è crittografato (se si tratta di un cluster fornito). Vedi Crittografia del database Amazon Redshift.

La mia integrazione è bloccata in uno stato di **`Syncing`**

La tua integrazione potrebbe mostrare costantemente uno stato pari a `Syncing` se modifichi il valore di uno dei parametri del DB richiesti.

Per risolvere questo problema, controlla i valori dei parametri nel gruppo di parametri associato al del database di origine e assicurati che corrispondano ai valori richiesti. Per ulteriori informazioni, consulta the section called “Fase 1: creazione di un gruppo di parametri del DB personalizzato”.

Se modificate qualche parametro, assicuratevi di riavviare il del database per applicare le modifiche.

Le mie tabelle non si replicano su Amazon Redshift

I tuoi dati potrebbero non essere replicati perché una o più tabelle di origine non hanno una chiave primaria. La dashboard di monitoraggio in Amazon Redshift mostra lo stato di queste tabelle e lo `Failed` stato dell'integrazione Zero-ETL complessiva cambia in. `Needs attention`

Per risolvere questo problema, puoi identificare una chiave esistente nella tabella che può diventare una chiave primaria oppure puoi aggiungere una chiave primaria sintetica. Per soluzioni dettagliate, consulta Gestire le tabelle senza chiavi primarie durante la creazione di integrazioni Amazon Aurora MySQL o Amazon RDS for MySQL Zero-ETL con Amazon Redshift.

Una o più tabelle Amazon Redshift richiedono una risincronizzazione

L'esecuzione di determinati comandi sull'istanza database di origine potrebbe richiedere la risincronizzazione delle tabelle. In questi casi, la vista di sistema

SVV_INTEGRATION_TABLE_STATE mostra un valore di `table_state` pari a `ResyncRequired`, il che significa che l'integrazione deve ricaricare completamente i dati di quella tabella specifica da MySQL in Amazon Redshift.

Quando viene avviata la risincronizzazione della tabella, lo stato diventa `Syncing`. Non è necessario eseguire alcuna azione manuale per risincronizzare una tabella. Durante la risincronizzazione dei dati delle tabelle, non puoi accedervi in Amazon Redshift.

Di seguito sono riportati alcuni esempi di operazioni che possono modificare lo stato di una tabella in `ResyncRequired` e le possibili alternative da considerare.

Operazione	Esempio	In alternativa
Aggiunta di una colonna in una posizione specifica	`ALTER TABLE table_name ADD COLUMN column_name INTEGER NOT NULL first;`	Amazon Redshift non supporta l'aggiunta di colonne in posizioni specifiche e utilizzando le parole chiave `first` o `after`. Se l'ordine delle colonne nella tabella di destinazione non è rilevante, aggiungi la colonna alla fine della tabella utilizzando un comando più semplice:

Operazione	Esempio	In alternativa
		`ALTER TABLE table_name ADD COLUMN column_name column_type ;`

Operazione	Esempio	In alternativa
Aggiunta di una colonna timestamp con il valore predefinito CURRENT_TIMESTAMP	`ALTER TABLE table_name` `  ADD COLUMN column_name  TIMESTAMP` `  NOT NULL DEFAULT CURRENT_TIMESTAMP;`	Il CURRENT_TIMESTAMP valore per le righe della tabella esistenti viene calcolato da RDS per MySQL MySQL e non può essere simulato in Amazon Redshift senza la risincronizzazione completa dei dati della tabella. Se possibile, converti il valore predefinito in una costante letterale, ad esempio 2023-01-01 00:00:15, per evitare la latenza a livello di

Operazione	Esempio	In alternativa
		disponibilità della tabella.
Esecuzione di operazioni su più colonne all'interno di un unico comando	`ALTER TABLE table_name ADD COLUMN column_1, RENAME COLUMN column_2 TO column_3;`	Prendi in considerazione la possibilità di suddividere il comando in due operazioni distinte, ADD e RENAME, che non richiedono la risincronizzazione.

Amazon RDS per Db2

Amazon RDS supporta istanze DB che eseguono le seguenti edizioni di: IBM Db2

- Db2 Standard Edition
- Db2 Advanced Edition

Amazon RDS supporta istanze DB che eseguono le seguenti versioni di Db2:

- Db2 11.5

Per ulteriori informazioni sul supporto delle versioni secondarie, consulta Db2 nelle versioni Amazon RDS.

Prima di creare un'istanza DB, completa i passaggi indicati nella Configurazione di Amazon RDS sezione di questa guida per l'utente. Quando crei un'istanza DB utilizzando il tuo utente master, l'utente ottiene `DBADM` l'autorità, con alcune limitazioni. Utilizzate questo utente per attività amministrative come la creazione di account di database aggiuntivi. Non è possibile utilizzare l'`SYSADM`autorità a livello di `SYSMAINT` istanza o `SECADM` l'autorità a livello di database. `SYSCTRL`

Puoi creare:

- Istanze DB
- Snapshot DB
- Ripristini P oint-in-time
- Backup automatici dello storage
- Backup di archiviazione manuali

È possibile utilizzare istanze DB che eseguono Db2 all'interno di un cloud privato virtuale (VPC). Puoi anche aggiungere funzionalità alla tua istanza DB RDS for Db2 abilitando varie opzioni. Amazon RDS supporta implementazioni Multi-AZ per RDS for Db2 come soluzione di failover ad alta disponibilità.

> **Important**
>
> Per offrire un'esperienza di servizio gestito, Amazon RDS non fornisce accesso shell alle istanze database. Limita inoltre l'accesso a determinate procedure e tabelle di sistema che

richiedono privilegi elevati. È possibile accedere al database utilizzando client SQL standard come. IBM Db2 CLP Tuttavia, non è possibile accedere direttamente all'host utilizzando Telnet o Secure Shell (SSH).

Argomenti

Panoramica di Db2 su Amazon RDS

Puoi leggere le seguenti sezioni per avere una panoramica di Db2 su Amazon RDS.

Argomenti

Funzionalità RDS per Db2

Amazon RDS for Db2 supporta la maggior parte delle caratteristiche e funzionalità del IBM Db2 database. Alcune funzionalità potrebbero avere un supporto o privilegi limitati. Per ulteriori informazioni sulle funzionalità del database Db2 per versioni Db2 specifiche, consulta la documentazione. IBM Db2

Please change to "Puoi filtrare le nuove funzionalità Amazon RDS alla pagina Quali sono le novità del database?. Per Prodotti, scegli Amazon RDS. Quindi, è possibile effettuare la ricerca utilizzando parole chiave come. **`Db2 2023`**

> **Note**
>
> I seguenti elenchi non sono esaustivi.

Argomenti

- Funzionalità supportate in RDS for Db2
- Funzionalità non supportate in RDS for Db2

Funzionalità supportate in RDS for Db2

RDS for Db2 supporta funzionalità che includono funzionalità native IBM Db2 e funzionalità fondamentali di Amazon RDS.

Funzionalità native di IBM Db2

RDS per Db2 supporta le seguenti funzionalità del database Db2:

- Creazione di un database standard che utilizza un set di codici, regole di confronto, dimensioni della pagina e territorio definiti dal cliente. Utilizza la procedura rdsadmin.create_database memorizzata di Amazon RDS.
- Aggiunta, eliminazione o modifica di utenti e gruppi locali. Utilizza le stored procedure di Amazon RDS perConcessione e revoca dei privilegi.
- Creazione di ruoli con la rdsadmin.create_role stored procedure di Amazon RDS.
- Support per tabelle organizzate a righe standard.
- Support per il carico di lavoro analitico per tabelle organizzate a colonne.

- Capacità di definire funzionalità di compatibilità con DB2 come e. Oracle MySQL
- Support per procedure archiviate esterne Java basate.
- Support per la crittografia dei dati in transito tramite SSL/TLS.
- Monitoraggio dello stato di un database (`ALIVE`,, `DOWNSTORAGE_FULL`, `UNKNOWN` e). `STANDBY_CONNECTABLE`
- Ripristino di un database offline o online Linux (LE) fornito dal cliente. Utilizza le stored procedure di Amazon RDS perGestione dei database.
- Applicazione di log di archivio Db2 forniti dal cliente per mantenere il database sincronizzato con i database Db2 autogestiti. Utilizza le stored procedure di Amazon RDS perGestione dei database.
- Support per il controllo a livello di istanza Db2 e a livello di database.
- Support per una federazione omogenea.
- Capacità di caricare una tabella da file di dati in Amazon Simple Storage Service (Amazon S3).
- Autorizzazioni concesse a utenti, gruppi o ruoli, ad esempio `CONNECTSYSMON`,,`ACCESSCTRL`,`DATAACCESS`,`SQLADM`, `WLMADMEXPLAIN`, `LOAD` o `IMPLICIT_SCHEMA`

Funzionalità fondamentali di Amazon RDS

RDS for Db2 supporta le seguenti funzionalità di base di Amazon RDS:

- Gruppi di parametri personalizzati da assegnare alle istanze DB.
- Creazione, modifica ed eliminazione di istanze DB.
- Ripristino di un backup del database Db2 offline o online Linux (LE) autogestito.

> **Note**
>
> Per poter ripristinare il backup, non fornire un nome per il database quando crei un'istanza DB. Per ulteriori informazioni, consulta Creazione di un'istanza database Amazon RDS.

- Support dei tipi di storage gp3, io2 e io1.
- Uso di AWS Managed Microsoft AD per l'Kerberosautenticazione e l'autorizzazione di gruppo LDAP per RDS for Db2.
- Modifica di gruppi di sicurezza, porte, tipi di istanze, archiviazione, periodi di conservazione dei backup e altre impostazioni per le istanze Db2 esistenti.

- Protezione dall'eliminazione per le istanze DB.
- point-in-time Ripristino interregionale (PITR).
- Uso di AWS Key Management Service (AWS KMS) per la crittografia dello storage e la crittografia a riposo.
- Istanze DB Multi-AZ con un unico standby per un'elevata disponibilità.
- Riavvio delle istanze DB.
- Aggiornamenti alle password principali.
- Ripristino delle istanze DB a un orario specifico.
- Backup e ripristino di istanze DB utilizzando backup a livello di storage.
- Avvio e arresto delle istanze DB.
- Manutenzione delle istanze DB.

Funzionalità non supportate in RDS for Db2

RDS per Db2 non supporta le seguenti funzionalità del database Db2:

- `SYSADM`e `SYSMAINT` accesso per l'utente principale. `SECADM`
- Procedure memorizzate esterne scritte in C, C++ o Cobol.
- Più istanze DB Db2 su un singolo host.
- Più database Db2 su una singola istanza DB RDS per Db2.
- Plugin GSS-API esterni per l'autenticazione.
- Plugin esterni di terze parti per il backup o il ripristino dei database Db2.
- Elaborazione MPP (Massively Parallel Processing) multinodo, ad esempio. IBM Db2 Warehouse
- IBM Db2 pureScale.
- Disaster Recovery ad alta disponibilità (HADR).
- Crittografia nativa del database.
- Federazione eterogenea per Db2.
- Cross-Region point-in-time-recovery (PITR) per backup crittografati.
- Creazione di routine non recintate. Per ulteriori informazioni, consulta Routine non recintate.
- Creazione di nuovi tablespace di archiviazione non automatici. Per ulteriori informazioni, consulta tablespace di archiviazione non automatici durante la migrazione

Db2 nelle versioni Amazon RDS

Per Db2, i numeri di versione assumono la forma di major.minor.build.revision, ad esempio 11.5.9.0.sb00000000.r1. L'implementazione della nostra versione corrisponde a quella di Db2.

importante

Il numero di versione principale è sia il numero intero che la prima parte frazionaria del numero di versione, ad esempio 11.5. Una modifica di versione è considerata importante se il numero della versione principale cambia, ad esempio se si passa dalla versione 11.5 alla 12.1.

minore

Il numero di versione secondario è costituito sia dalla terza che dalla quarta parte del numero di versione, ad esempio 9.0 in 11.5.9.0. La terza parte indica il modpack Db2, ad esempio 9 in 9.0. La quarta parte indica il fixpack Db2, ad esempio 0 in 9.0. Una modifica di versione è considerata minore se il modpack Db2 o il fixpack Db2 vengono modificati, ad esempio se si passa dalla versione 11.5.9.0 alla 11.5.9.1 o dalla 11.5.9.0 alla 11.5.10.0, con eccezioni relative agli aggiornamenti delle tabelle di catalogo. (Amazon RDS si occupa di queste eccezioni).

costruire

Il numero di build è la quinta parte del numero di versione, ad esempio sb00000000 in 11.5.9.0.sb00000000. Un numero di build in cui la parte numerica è composta da zero indica una build standard. Un numero di build in cui la parte numerica non è composta solo da zeri indica una build speciale. Il numero di build cambia se è presente una correzione di sicurezza o una build speciale di una versione Db2 esistente. Una modifica del numero di build indica inoltre che Amazon RDS ha applicato automaticamente una nuova versione secondaria.

revisione

Il numero di revisione è la sesta parte del numero di versione, ad esempio r1 in 11.5.9.0.sb00000000.r1. Una revisione è una revisione di Amazon RDS di una versione Db2 esistente. Una modifica del numero di revisione indica che Amazon RDS ha applicato automaticamente una nuova versione secondaria.

Argomenti

Versioni secondarie Db2 supportate su Amazon RDS

La tabella seguente mostra le versioni secondarie di Db2 attualmente supportate da Amazon RDS.

> **Note**
>
> Le date con solo un mese e un anno sono approssimative e vengono aggiornate con una data esatta quando nota.

Versione del motore Db2	IBMdata di rilascio	Data di rilascio per RDS	Data di fine del supporto standard RDS
11.5			
11.5.9.0	15 novembre 2023	27 novembre 2023	

È possibile specificare qualsiasi versione Db2 attualmente supportata durante la creazione di una nuova istanza DB. È possibile specificare la versione principale (ad esempio Db2 11.5) e qualsiasi versione secondaria supportata per la versione principale specificata. Se non viene specificata alcuna versione, Amazon RDS utilizza per impostazione predefinita una versione supportata, in genere la più recente. Se viene specificata una versione principale ma non una secondaria, per impostazione predefinita Amazon RDS utilizza una release recente della versione principale specificata. Per visualizzare un elenco delle versioni supportate, nonché i valori predefiniti per le istanze DB appena create, utilizzate il comando (). describe-db-engine-versionsAWS Command Line InterfaceAWS CLI

Ad esempio, per elencare le versioni del motore supportate per RDS for Db2, esegui il comando seguente. AWS CLI Sostituisci la *regione con la* tua. Regione AWS

Per LinuxmacOS, oUnix:

```
aws rds describe-db-engine-versions \
    --filters Name=engine,Values=db2-ae,db2-se \
    --query "DBEngineVersions[].{Engine:Engine, EngineVersion:EngineVersion,
 DBParameterGroupFamily:DBParameterGroupFamily}" \
    --region region
```

Per Windows:

```
aws rds describe-db-engine-versions ^
    --filters Name=engine,Values=db2-ae,db2-se ^
    --query "DBEngineVersions[].{Engine:Engine, EngineVersion:EngineVersion,
 DBParameterGroupFamily:DBParameterGroupFamily}" ^
    --region region
```

Questo comando produce un output simile al seguente esempio:

```
[
    {
    "Engine": "db2-ae",
    "EngineVersion": "11.5.9.0.sb00000000.r1",
    "DBParameterGroupFamily": "db2-ae-11.5"
    },
    {
    "Engine": "db2-se",
    "EngineVersion": "11.5.9.0.sb00000000.r1",
    "DBParameterGroupFamily": "db2-se-11.5"
    }
]
```

La versione predefinita di Db2 potrebbe variare diRegione AWS. Per creare un'istanza DB con una versione secondaria specifica, specifica la versione secondaria durante la creazione dell'istanza DB. È possibile determinare la versione predefinita per un motore `db2-ae` e Regione AWS per i motori di `db2-se` database eseguendo il `describe-db-engine-versions` comando. L'esempio seguente restituisce la versione predefinita per gli `db2-ae` Stati Uniti orientali (Virginia settentrionale).

Per LinuxmacOS, oUnix:

```
aws rds describe-db-engine-versions \
    --default-only --engine db2-ae \
    --query "DBEngineVersions[].{Engine:Engine, EngineVersion:EngineVersion,
 DBParameterGroupFamily:DBParameterGroupFamily}" \
    --region us-east-1
```

Per Windows:

```
aws rds describe-db-engine-versions ^
    --default-only --engine db2-ae ^
    --query "DBEngineVersions[].{Engine:Engine, EngineVersion:EngineVersion,
 DBParameterGroupFamily:DBParameterGroupFamily}" ^
```

```
    --region us-east-1
```

Questo comando produce un output simile al seguente esempio:

```
[
    {
    "Engine": "db2-ae",
    "EngineVersion": "11.5.9.0.sb00000000.r1",
    "DBParameterGroupFamily": "db2-ae-11.5"
    }
]
```

Con Amazon RDS, puoi controllare quando aggiornare la tua istanza Db2 a una nuova versione principale supportata da Amazon RDS. Puoi mantenere la compatibilità con versioni Db2 specifiche, testare nuove versioni con la tua applicazione prima di distribuirla in produzione ed eseguire gli aggiornamenti principali delle versioni nei momenti più adatti alla tua pianificazione.

Quando l'aggiornamento automatico delle versioni secondarie è abilitato, Amazon RDS aggiorna automaticamente le istanze DB alle nuove versioni secondarie Db2, poiché sono supportate da Amazon RDS. L'applicazione di patch avviene durante la finestra di manutenzione pianificata. È possibile modificare un'istanza DB per abilitare o disabilitare gli aggiornamenti automatici delle versioni secondarie.

Ad eccezione delle versioni Db2 11.5.9.1 e 11.5.10.0, gli aggiornamenti automatici alla nuova versione minore di Db2 includono aggiornamenti automatici a nuove build e revisioni. Per 11.5.9.1 e 11.5.10.0, aggiorna manualmente le versioni secondarie.

Se annulli gli aggiornamenti automatici pianificati, puoi eseguire manualmente l'aggiornamento a una versione secondaria supportata seguendo la stessa procedura utilizzata per l'aggiornamento di una versione principale. Per informazioni, consulta Aggiornamento della versione del motore di un'istanza database.

Versioni principali di Db2 supportate su Amazon RDS

Le versioni principali di RDS per Db2 sono disponibili con il supporto standard almeno fino IBM alla fine del supporto (base) per la versione corrispondente. IBM La tabella seguente mostra le date che è possibile utilizzare per pianificare i cicli di test e aggiornamento. Se Amazon estende il supporto per una versione RDS for Db2 per un periodo più lungo di quanto originariamente dichiarato, intendiamo aggiornare questa tabella in modo che rifletta la data successiva.

È possibile utilizzare le date seguenti per pianificare i cicli di test e aggiornamento.

> **Note**
>
> Le date con solo un mese e un anno sono approssimative e vengono aggiornate con una data esatta quando nota.

Versione principale di Db2	IBMdata di rilascio	Data di rilascio per RDS	IBMfine del supporto (base)	IBMfine del supporto (esteso)	Data di fine del supporto standard RDS
Db2 11.5	27 giugno 2019	27 novembre 2023	30 settembre 2025	4 anni dopo la fine del supporto	

Opzioni di licenza Amazon RDS per Db2

Amazon RDS per Db2 offre due opzioni di licenza: Bring Your Own License (BYOL) e Db2 license through. Marketplace AWS

Argomenti

- Porta la tua licenza per Db2
- Licenza Db2 tramite Marketplace AWS
- Passaggio da una licenza Db2 all'altra

Porta la tua licenza per Db2

Nel modello BYOL, utilizzi le licenze di database Db2 esistenti per distribuire database su Amazon RDS. Verifica di disporre della licenza di database Db2 appropriata per la classe di istanze DB e l'edizione del database Db2 che desideri eseguire. È inoltre necessario seguire IBM le politiche per la concessione di licenze del software di IBM database nell'ambiente di cloud computing.

> **Note**
>
> Le istanze DB Multi-AZ sono «cold standby» perché il database Db2 è installato ma non è in esecuzione. Le standby non sono leggibili, non sono in esecuzione o non soddisfano le

richieste. Per ulteriori informazioni, consulta le informazioni sulle IBM Db2licenze sul sito Web IBM.

In questo modello, si continua a utilizzare l'account di IBM supporto attivo e si contatta IBM direttamente per le richieste di servizi di database Db2. Se disponi di un AWS Support account con assistenza clienti, puoi contattarci AWS Support per problemi relativi ad Amazon RDS. Amazon Web Services e IBM disponiamo di un processo di supporto multivendor per i casi che richiedono assistenza da entrambe le organizzazioni.

Amazon RDS supporta il modello BYOL per e. Db2 Standard Edition Db2 Advanced Edition

Argomenti

- IBMID per Bring Your Own License for Db2
- Aggiungere IBM ID a un gruppo di parametri per RDS per istanze DB Db2
- Integrazione con AWS License Manager

IBMID per Bring Your Own License for Db2

Nel modello BYOL, è necessario creare, modificare o ripristinare RDS per le istanze DB Db2. IBM Customer ID IBM Site ID È necessario creare un gruppo di parametri personalizzato con il proprio utente IBM Site ID prima di creare un'IBM Customer IDistanza DB RDS per Db2. Per ulteriori informazioni, consulta Aggiungere IBM ID a un gruppo di parametri per RDS per istanze DB Db2. È possibile eseguire più istanze RDS for Db2 DB con istanze diverse IBM Customer IDs e IBM Site IDs nello stesso sistema operativo. Account AWS Regione AWS

Important

Se sei un IBM Db2 cliente esistente, puoi trovare il tuo IBM Customer ID e il tuo IBM Site ID sul certificato Proof of Entitlement di. IBM Per ulteriori informazioni, consulta le istruzioni su come visualizzare il tuo annuncio IBM Site ID sul IBM Customer ID sito Web IBM.
Se sei un nuovo IBM Db2 cliente, devi prima acquistare una licenza software Db2 da. IBM Dopo aver acquistato una licenza software Db2, riceverai un Proof of Entitlement daIBM, che elenca i tuoi e i tuoi. IBM Customer ID IBM Site ID
Se non riusciamo a verificare la tua licenza da parte tua IBM Customer ID e tuaIBM Site ID, potremmo chiudere tutte le istanze DB in esecuzione con queste licenze non verificate.

Aggiungere IBM ID a un gruppo di parametri per RDS per istanze DB Db2

Poiché non è possibile modificare i gruppi di parametri predefiniti, è necessario creare un gruppo di parametri personalizzato e quindi modificarlo per includere i valori propri e personaliIBM Customer ID. IBM Site ID Per informazioni sui gruppi di parametri, consultare Utilizzo di gruppi di parametri DB in un'istanza DB.

> **Important**
>
> È necessario creare un gruppo di parametri personalizzato con il proprio IBM Customer ID utente IBM Site ID prima di creare un'istanza DB RDS for Db2.

Utilizzate le impostazioni dei parametri nella tabella seguente.

Parametro	Valore
`rds.ibm_customer_id`	`<your IBM Customer ID>`
`rds.ibm_site_id`	`<your IBM Site ID>`
`ApplyMethod`	`immediate` , `pending-reboot`

Questi parametri sono dinamici, il che significa che qualsiasi modifica ad essi ha effetto immediato e che non è necessario riavviare l'istanza DB. Se non desideri che le modifiche abbiano effetto immediato, puoi impostarle `pending-reboot` e `ApplyMethod` programmarle in modo che vengano apportate durante una finestra di manutenzione.

Puoi creare e modificare un gruppo di parametri personalizzato utilizzando l' AWS Management Console AWS CLI, la o l'API Amazon RDS.

Console

Per aggiungere i tuoi IBM Customer ID e i tuoi IBM Site ID a un gruppo di parametri

1. Crea un nuovo gruppo di parametri DB. Per ulteriori informazioni sulla creazione di un gruppo di parametri database, consulta Creazione di un gruppo di parametri del database.
2. Modifica il gruppo di parametri che hai creato. Per ulteriori informazioni sulla modifica di un gruppo di parametri, vedereModifica di parametri in un gruppo di parametri del database.

AWS CLI

Per aggiungere i tuoi IBM Customer ID e i tuoi IBM Site ID a un gruppo di parametri

1. Crea un gruppo di parametri personalizzato eseguendo il create-db-parameter-groupcomando.

   Includi le seguenti opzioni obbligatorie:

   - `--db-parameter-group-name`— Un nome per il gruppo di parametri che state creando.
   - `--db-parameter-group-family`— L'edizione e la versione principale del motore Db2. Valori validi: `db2-se-11.5`, `db2-ae-11.5`.
   - `--description`— Una descrizione per questo gruppo di parametri.

   Per ulteriori informazioni sulla creazione di un gruppo di parametri database, consulta Creazione di un gruppo di parametri del database.
2. Modificate i parametri nel gruppo di parametri personalizzato creato eseguendo il modify-db-parameter-groupcomando.

   Includi le seguenti opzioni obbligatorie:

   - `--db-parameter-group-name`— Il nome del gruppo di parametri creato.
   - `--parameters`— Una matrice di nomi di parametri, valori e metodi di applicazione per l'aggiornamento dei parametri.

   Per ulteriori informazioni sulla modifica di un gruppo di parametri, vedereModifica di parametri in un gruppo di parametri del database.

API RDS

Per aggiungere i tuoi IBM Customer ID e i tuoi IBM Site ID a un gruppo di parametri

1. Crea un gruppo di parametri DB personalizzato utilizzando l'CreateDBParameterGroupoperazione dell'API Amazon RDS.

   Includi i parametri obbligatori seguenti:

   - `DBParameterGroupName`
   - `DBParameterGroupFamily`

- `Description`

Per ulteriori informazioni sulla creazione di un gruppo di parametri database, consulta Creazione di un gruppo di parametri del database.

2. Modifica i parametri nel gruppo di parametri personalizzato che hai creato utilizzando l'operazione dell'API ModifyDBParameterGroupRDS.

   Includi i parametri obbligatori seguenti:

   - `DBParameterGroupName`
   - `Parameters`

   Per ulteriori informazioni sulla modifica di un gruppo di parametri, vedere. Modifica di parametri in un gruppo di parametri del database

Ora sei pronto per creare un'istanza DB e collegare il gruppo di parametri personalizzato all'istanza DB. Per ulteriori informazioni, consulta Creazione di un'istanza database Amazon RDS e Associazione di un gruppo di parametri database a un'istanza database.

Integrazione con AWS License Manager

Per facilitare il monitoraggio dell'utilizzo delle licenze RDS for Db2 nel modello BYOL, si AWS License Managerintegra con RDS for Db2. License Manager supporta il tracciamento delle edizioni del motore RDS for Db2 basate su CPU virtuali (vCPU). È inoltre possibile utilizzare License Manager con AWS Organizations per gestire centralmente tutti gli account aziendali.

La tabella seguente mostra i filtri di informazioni sul prodotto per RDS for Db2.

Filter	Nome	Descrizione
Edizione motore	`db2-se`	Db2 Standard Edition
	`db2-ae`	Edizione avanzata Db2

Per tenere traccia dell'utilizzo della licenza di RDS per le istanze DB Db2, puoi creare una licenza autogestita. In questo caso, le risorse RDS for Db2 che corrispondono al filtro delle informazioni sul

prodotto vengono associate automaticamente alla licenza autogestita. L'individuazione di istanze DB RDS per Db2 può richiedere fino a 24 ore.

Console

Per creare una licenza autogestita per tenere traccia dell'utilizzo della licenza di RDS per le istanze DB Db2

1. Passare a https://console.aws.amazon.com/license-manager/.
2. Crea una licenza autogestita.

   Per istruzioni, consulta Creare una licenza autogestita nella Guida per l'AWS License Manager utente.

   Aggiungere una regola per un RDS Product Information Filter (Filtro di informazioni sui prodotti RDS) nel pannello Product Information (Informazioni sul prodotto) .

   Per ulteriori informazioni, consulta la ProductInformationsezione AWS License Manager API Reference.

AWS CLI

Per creare una licenza autogestita utilizzando AWS CLI, chiamate il create-license-configurationcomando. Utilizza i parametri `--cli-input-json` o `--cli-input-yaml` per passare i parametri al comando.

Example

Il codice seguente crea una licenza autogestita per Db2 Standard Edition.

```
aws license-manager create-license-configuration --cli-input-json file://rds-db2-
se.json
```

Di seguito è riportato il file `rds-db2-se.json` di esempio utilizzato.

```
{
    "Name": "rds-db2-se",
    "Description": "RDS Db2 Standard Edition",
    "LicenseCountingType": "vCPU",
    "LicenseCountHardLimit": false,
    "ProductInformationList": [
```

```
        {
            "ResourceType": "RDS",
            "ProductInformationFilterList": [
                {
                    "ProductInformationFilterName": "Engine Edition",
                    "ProductInformationFilterValue": ["db2-se"],
                    "ProductInformationFilterComparator": "EQUALS"
                }
            ]
        }
    ]
}
```

Per ulteriori informazioni sul prodotto, consulta la pagina relativa all'individuazione automatica dell'inventario delle risorse nella Guida per l'utente di AWS License Manager .

Per ulteriori informazioni sul parametro `--cli-input`, consulta Generazione di parametri di input e skeleton AWS CLI da un file di input JSON o YAML nella Guida per l'utente di AWS CLI .

Licenza Db2 tramite Marketplace AWS

Nella licenza Db2 through Marketplace AWS model, si paga una tariffa oraria per abbonarsi alle licenze Db2. Questo modello consente di iniziare rapidamente a utilizzare RDS for Db2 senza dover acquistare licenze.

Per utilizzare la licenza Db2 tramite Marketplace AWS, è necessario un Marketplace AWS abbonamento attivo per la particolare IBM Db2 edizione che si desidera utilizzare. Se non ne hai già uno, abbonati Marketplace AWS per quell'IBM Db2edizione.

Amazon RDS supporta la licenza Db2 Marketplace AWS per IBM Db2 Standard Edition e IBM Db2 Advanced Edition.

Argomenti

- Terminologia
- Pagamenti e fatturazione
- Iscrizione alle inserzioni di Db2 Marketplace e registrazione con IBM

Terminologia

Questa pagina utilizza la seguente terminologia per discutere dell'integrazione di Amazon RDS con. Marketplace AWS

Abbonamento SaaS

Nel Marketplace AWS, i prodotti software-as-a-service (SaaS) come il modello di pay-as-you-go licenza adottano un modello di abbonamento basato sull'utilizzo. IBM, il venditore di software per Db2, monitora il tuo utilizzo e paghi solo per ciò che usi.

Offerta pubblica

Le offerte pubbliche consentono di acquistare Marketplace AWS prodotti direttamente da AWS Management Console.

Commissioni di Db2 Marketplace

Commissioni addebitate per l'utilizzo della licenza del software Db2 da. IBM Questi costi di servizio vengono contabilizzati Marketplace AWS e appaiono sulla AWS fattura nella Marketplace AWS sezione.

Amazon RDS

Commissioni AWS addebitate per i servizi RDS per Db2, che escludono le licenze utilizzate per le licenze Db2. Marketplace AWS Le tariffe vengono contabilizzate tramite il servizio Amazon RDS utilizzato e appaiono sulla fattura AWS .

Pagamenti e fatturazione

RDS for Db2 si integra con per offrire licenze orarie Marketplace AWS per Db2. pay-as-you-go Le tariffe di Db2 Marketplace coprono i costi di licenza del software Db2 e le tariffe di Amazon RDS coprono i costi del tuo RDS per l'utilizzo delle istanze DB Db2. Per informazioni sui prezzi, consulta i prezzi di Amazon RDS for Db2.

Per eliminare queste commissioni, devi eliminare qualsiasi istanza di database RDS per Db2. Inoltre, puoi rimuovere gli abbonamenti alle licenze For Db2. Marketplace AWS Se rimuovi gli abbonamenti senza eliminare le istanze DB, Amazon RDS continuerà a fatturarti l'uso delle istanze DB.

Puoi visualizzare le fatture e gestire i pagamenti per le tue istanze DB RDS per Db2 che utilizzano la licenza Db2 tramite la console. Marketplace AWSAWS Billing Le fatture includono due addebiti: uno per l'utilizzo della licenza Db2 tramite Marketplace AWS e uno per l'utilizzo di Amazon RDS. Per ulteriori informazioni sulla fatturazione, consulta Visualizzazione della fattura nella Guida per l'utente.AWS Billing and Cost Management

Iscrizione alle inserzioni di Db2 Marketplace e registrazione con IBM

Per utilizzare la licenza Db2 tramite Marketplace AWS, è necessario utilizzare il AWS Management Console per completare le due attività seguenti. Non è possibile completare queste attività tramite l'API RDS AWS CLI o l'API RDS.

> **Note**
>
> Se desideri creare le tue istanze DB utilizzando AWS CLI o l'API RDS, devi prima completare queste due attività.

Argomenti

- Attività 1: Abbonarsi a Db2 in Marketplace AWS
- Attività 2: Registra il tuo abbonamento con IBM

Attività 1: Abbonarsi a Db2 in Marketplace AWS

Per utilizzare la licenza Db2 con Marketplace AWS, è necessario disporre di un Marketplace AWS abbonamento attivo per Db2. Poiché gli abbonamenti sono associati a un'IBM Db2edizione specifica, è necessario abbonarsi a Db2 Marketplace AWS per ogni edizione di Db2 che si desidera utilizzare: IBM Db2Advanced Edition, Standard Edition. IBM Db2 Per informazioni sugli Marketplace AWS abbonamenti, consulta Abbonamenti basati sull'utilizzo Saas nella Guida all'acquisto.Marketplace AWS

Ti consigliamo di abbonarti a Db2 Marketplace AWSprima di iniziare a creare un'istanza DB.

Attività 2: Registra il tuo abbonamento con IBM

Dopo esserti abbonato a Db2 in Marketplace AWS, completa la registrazione del tuo ordine IBM dalla Marketplace AWS pagina relativa al tipo di abbonamento Db2 che hai scelto. Nella Marketplace AWS pagina, scegli Visualizza opzioni di acquisto, quindi scegli Configura il tuo account. Puoi registrarti con il tuo IBM account esistente o creando un IBM account gratuito.

Passaggio da una licenza Db2 all'altra

È possibile passare da una licenza Db2 all'altra in RDS for Db2. Ad esempio, puoi iniziare con Bring Your Own License e poi passare alla licenza Db2 tramite. Marketplace AWS

> **Important**
>
> Se desideri passare alla licenza Db2 tramite Marketplace AWS, assicurati di avere un Marketplace AWS abbonamento attivo per l'IBM Db2edizione che desideri utilizzare. In caso contrario, prima abbonatevi a Db2 Marketplace AWS per quell'edizione Db2, e poi completate la procedura di ripristino.

Console

Per passare da una licenza Db2 all'altra

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, selezionare Automated backups (Backup automatici).

   I backup automatici vengono visualizzati nella scheda Current Region (Regione corrente).
3. Scegli l'istanza database da ripristinare.
4. In Actions (Operazioni), scegli Restore to point in time (Ripristina a un istante temporale).

   Viene visualizzata la finestra Restore to point in time (Ripristina a un istante temporale).
5. Scegliere Latest restorable time (Ultimo orario di ripristino) per eseguire il ripristino in base al momento più recente oppure scegliere Custom (Personalizzato) per scegliere una data e un'ora.

   Se hai scelto Personalizzato, inserisci la data e l'ora in cui desideri ripristinare l'istanza.

   > **Note**
   >
   > Gli orari vengono visualizzati nel fuso orario locale, indicato come un offset dell'ora UTC (Coordinated Universal Time). Ad esempio, UTC-5 è l'orario standard degli Stati Uniti orientali/ora legale degli Stati Uniti centrali.
6. Per il motore DB, scegli la licenza Db2 che desideri utilizzare.
7. Per DB Instance Identifier (Identificatore istanze database), inserire il nome dell'istanza database di destinazione ripristinata. Il nome deve essere univoco.
8. Scegli altre opzioni in base alle esigenze, ad esempio la classe di istanza database, l'archiviazione e se desideri utilizzare la funzione di scalabilità automatica dell'archiviazione.

Per informazioni su ciascuna impostazione, consulta Impostazioni per istanze database.

9. Scegli Restore to point in time (Ripristina per punto nel tempo).

Per ulteriori informazioni, consulta Ripristino a un'ora specifica per un'istanza database.

AWS CLI

Per passare da una licenza Db2 all'altra, usa il AWS CLI comando -. restore-db-instance-to point-in-time L'esempio seguente ripristina la point-in-time versione più recente, imposta il motore DB su IBM Db2 Advanced Edition e imposta il modello di licenza su Db2 license through. Marketplace AWS

È possibile specificare altre impostazioni. Per informazioni su ciascuna impostazione, consulta Impostazioni per istanze database.

Example

PerLinux, omacOS: Unix

```
aws rds restore-db-instance-to-point-in-time \
    --source-db-instance-identifier my_source_db_instance \
    --target-db-instance-identifier my_target_db_instance \
    --use-latest-restorable-time \
    --engine db2-ae \
    --license-model marketplace-license
```

Per Windows:

```
aws rds restore-db-instance-to-point-in-time ^
    --source-db-instance-identifier my_source_db_instance ^
    --target-db-instance-identifier my_target_db_instance ^
    --use-latest-restorable-time ^
    --engine db2-ae ^
    --license-model marketplace-license
```

Per ulteriori informazioni, consulta Ripristino a un'ora specifica per un'istanza database.

API RDS

Per passare da una licenza Db2 all'altra, chiama l'`RestoreDBInstanceToPointInTime`operazione dell'API Amazon RDS con i seguenti parametri:

- `SourceDBInstanceIdentifier`
- `TargetDBInstanceIdentifier`
- `RestoreTime`
- `Engine`
- `LicenseModel`

Per ulteriori informazioni, consulta Ripristino a un'ora specifica per un'istanza database.

RDS per classi di istanze Db2

La capacità di calcolo e memoria di un'istanza database è determinata dalla relativa classe. La classe di istanza database di cui hai bisogno dipende dalla potenza di elaborazione e dai requisiti di memoria specifici.

Classi di istanze RDS per Db2 supportate

Le classi di istanze RDS for Db2 supportate sono un sottoinsieme delle classi di istanze Amazon RDS DB. Per l'elenco completo delle classi di istanze Amazon RDS, consultaClassi di istanze database.

La tabella seguente elenca tutte le classi di istanza supportate per il database Db2 11.5.9.0.

Edizione Db2	Db2 versione 11.5.9.0
Db2 Standard Edition Modello di licenza Bring Your Own License (BYOL) Licenza Db2 tramite Marketplace AWS	Classi di istanze per uso generico con Intel Xeon Scalable processori di terza generazione, storage SSD e ottimizzazione della rete
	db.m6idn.large — db.m6idn.8xlarge
	Classi di istanze per uso generico basate su processori di terza generazione Intel Xeon Scalable
	db.m6in.large—db.m6in.8xlarge
	Classi di istanze per uso generico
	db.m6i.large — db.m6i.8xlarge

Edizione Db2	Db2 versione 11.5.9.0
	Classi di istanze ottimizzate per la memoria con SSD locali basati su NVMe, alimentate da processori di terza generazione Intel Xeon Scalable
	db.x2iedn.xlarge
	Classi di istanze ottimizzate per la memoria basate su processor i di terza generazione Intel Xeon Scalable
	db.r6idn.large—db.r6idn.4xlarge
	Classi di istanze ottimizzate per la memoria basate su processor i di terza generazione Intel Xeon Scalable
	db.r6in.large—db.r6in.4xlarge
	Classi di istanza ottimizzata per la memoria
	db.r6i.large–db.r6i.4xlarge
	Classi di istanza espandibile
	db.t3.small–db.t3.2xlarge
Db2 Advanced Edition Modello di licenza Bring Your Own License (BYOL) Licenza Db2 tramite Marketpla ce AWS	Classi di istanze per uso generico con Intel Xeon Scalable processori di terza generazione, storage SSD e ottimizzazione della rete
	db.m6idn.12xlarge—db.m6idn.32xlarge
	Classi di istanze per uso generico basate su processori di terza generazione Intel Xeon Scalable
	db.m6in.12xlarge—db.m6in.32xlarge
	Classi di istanze per uso generico
	db.m6i.12xlarge—db.m6i.32xlarge

Edizione Db2	Db2 versione 11.5.9.0
	Classi di istanze ottimizzate per la memoria con SSD locali basati su NVMe, alimentate da processori di terza generazione Intel Xeon Scalable
	db.x2iedn.2xlarge—db.x2iedn.32xlarge
	Classi di istanze ottimizzate per la memoria basate su processor i di terza generazione Intel Xeon Scalable
	db.r6idn.8xlarge—db.r6idn.32xlarge
	Classi di istanze ottimizzate per la memoria basate su processor i di terza generazione Intel Xeon Scalable
	db.r6in.8xlarge—db.r6in.32xlarge
	Classi di istanza ottimizzata per la memoria
	db.r6i.8xlarge — db.r6i.32xlarge

RDS per i parametri Db2

RDS per Db2 supporta la modifica dei parametri del gestore di database (a livello di istanza) e dei parametri del registro Db2 tramite gruppi di parametri. I parametri del database sono modificabili solo tramite la stored procedure. rdsadmin.update_db_param

Per impostazione predefinita, un'istanza DB RDS for Db2 utilizza un gruppo di parametri DB specifico per un database Db2 e un'istanza DB. Questo gruppo di parametri contiene i parametri per il IBM Db2 motore di database. Per informazioni sull'utilizzo dei gruppi di parametri e sull'impostazione dei parametri, consulta Utilizzo di gruppi di parametri.

I parametri RDS for Db2 sono impostati sui valori predefiniti del motore di archiviazione selezionato. Per ulteriori informazioni sui parametri Db2, consultate i parametri di configurazione del database Db2 nella documentazione. IBM Db2

È possibile visualizzare i parametri disponibili per una versione specifica di Db2 utilizzando AWS Management Console o il AWS Command Line Interface (). AWS CLI Per informazioni sulla

visualizzazione dei parametri in un gruppo di parametri Db2 nella console, vedere. Visualizzazione dei valori dei parametri per un gruppo di parametri del database

Utilizzando AWS CLI, è possibile visualizzare i parametri per una versione di Db2 eseguendo il describe-engine-default-parameterscomando. Indica uno dei valori seguenti per l'opzione `--db-parameter-group-family`:

- `db2-ae-11.5`
- `db2-se-11.5`

Ad esempio, per visualizzare i parametri per la versione Db2 Standard Edition 11.5, esegui il comando seguente.

```
aws rds describe-engine-default-parameters --db-parameter-group-family db2-se-11.5
```

Questo comando produce un output simile all'esempio seguente.

```
{
    "EngineDefaults": {
        "Parameters": [
            {
                "ParameterName": "agent_stack_sz",
                "ParameterValue": "1024",
                "Description": "You can use this parameter to determine the amount of
 memory that is allocated by Db2 for each agent thread stack.",
                "Source": "engine-default",
                "ApplyType": "static",
                "DataType": "integer",
                "AllowedValues": "256-32768",
                "IsModifiable": false
            },
            {
                "ParameterName": "agentpri",
                "ParameterValue": "-1",
                "Description": "This parameter controls the priority given to all
 agents and to other database manager instance processes and threads by the operating
 system scheduler. This priority determines how CPU time is allocated to the database
 manager processes, agents, and threads relative to other processes and threads running
 on the machine.",
                "Source": "engine-default",
                "ApplyType": "static",
```

```
                "DataType": "integer",
                "AllowedValues": "1-99",
                "IsModifiable": false
            },
            ...
        ]
    }
}
```

Per elencare solo i parametri modificabili per la versione Db2 Standard Edition 11.5, esegui il comando seguente:

PerLinux, o: macOS Unix

```
aws rds describe-engine-default-parameters \
    --db-parameter-group-family db2-se-11.5 \
    --query 'EngineDefaults.Parameters[?IsModifiable==`true`].
{ParameterName:ParameterName, DefaultValue:ParameterValue}'
```

Per Windows:

```
aws rds describe-engine-default-parameters ^
    --db-parameter-group-family db2-se-11.5 ^
    --query 'EngineDefaults.Parameters[?IsModifiable==`true`].
{ParameterName:ParameterName, DefaultValue:ParameterValue}'
```

Argomenti

- Determinare quali parametri sono modificabili
- Modifica dei parametri

Determinare quali parametri sono modificabili

Per determinare quali parametri del gestore di database, del database e del registro è possibile modificare, esegui i comandi seguenti.

1. Connect al database Db2. *Nell'esempio seguente, sostituisci database_name, master_username e master_password con le tue informazioni.*

   ```
   db2 "connect to database_name user master_username using master_password"
   ```

2. Trova la versione Db2 supportata.

```
db2 "select service_level, fixpack_num from table(sysproc.env_get_inst_info()) as
 instanceinfo"
```

3. Visualizza i parametri per una versione specifica di Db2.

   - Visualizza i parametri di configurazione del gestore del database. Controlla il gruppo di parametri associato alla tua istanza DB utilizzando AWS Management Console o eseguendo il comando seguente:

```
db2 "select cast(substr(name,1,24) as varchar(24)) as name, case
    when value_flags = 'NONE' then '' else value_flags end flags,
    cast(substr(value,1,64) as varchar(64)) as current_value
    from sysibmadm.dbmcfg
    order by name asc with UR"
```

   - Visualizza tutti i parametri di configurazione del database.

```
db2 "select cast(substr(name,1,24) as varchar(24)) as name, case
    when value_flags = 'NONE' then '' else value_flags end flags,
    cast(substr(value,1,64) as varchar(64)) as current_value
    from table(db_get_cfg(null)) order by name asc, member asc with UR"
```

   - Visualizza le variabili di registro attualmente impostate.

```
db2 "select cast(substr(reg_var_name,1,50) as varchar(50)) as reg_var_name,
    cast(substr(reg_var_value,1,50) as varchar(50)) as reg_var_value,
    level from table(env_get_reg_variables(null))
    order by reg_var_name,member with UR"
```

   - Visualizza l'elenco di tutte le variabili di registro supportate.

```
db2 "select cast(substr(reg_var_name,1,50) as varchar(50)) as reg_var_name,
    cast(substr(reg_var_value,1,50) as varchar(50)) as reg_var_value,
    level from table(env_get_reg_variables(null,1))
    order by reg_var_name,member with UR"
```

Modifica dei parametri

È possibile modificare il gestore del database e i parametri del registro in gruppi di parametri personalizzati. Create innanzitutto un gruppo di parametri personalizzato, quindi modificate i parametri in quel gruppo di parametri personalizzato. Per ulteriori informazioni, consulta Utilizzo di gruppi di parametri DB in un'istanza DB.

Per modificare i parametri del database, esegui i comandi seguenti.

1. Connect al `rdsadmin` database. Nell'esempio seguente, sostituisci *`master_username`* e *`master_password`* con le tue informazioni.

```
db2 "connect to rdsadmin user master_username using master_password"
```

2. Modificate i parametri del database chiamando la stored procedure. `rdsadmin.update_db_param` Per ulteriori informazioni, consulta rdsadmin.update_db_param.

```
db2 "call rdsadmin.update_db_param(
    'database_name',
    'parameter_to_modify',
    'changed_value')"
```

Collazione EBCDIC per database Db2 su Amazon RDS

RDS per Db2 supporta la collazione EBCDIC per i database Db2. Puoi specificare una sequenza di confronto EBCDIC per un database solo quando crei il database utilizzando la stored procedure di Amazon the section called “rdsadmin.create_database” RDS.

Quando crei un'istanza DB RDS for Db2 utilizzando l'API, o RDS AWS Management ConsoleAWS CLI, puoi specificare un nome di database. Se specifichi un nome di database, Amazon RDS crea un database con le regole di confronto predefinite di. `SYSTEM` Se devi creare un database con regole di confronto EBCDIC, non specificare un nome di database quando crei un'istanza DB.

La collazione per un database in RDS for Db2 viene impostata al momento della creazione ed è immutabile. Se hai specificato un nome di database quando hai creato un'istanza DB e desideri un database con regole di confronto EBCDIC, elimina l'istanza DB e creane una nuova.

Per creare un database Db2 con regole di confronto EBCDIC

1. Crea un'istanza DB RDS for Db2 senza specificare un nome di database utilizzando l'API, o RDS. AWS Management Console AWS CLI Per ulteriori informazioni, consulta Creazione di un'istanza database.
2. Crea un database Db2 e imposta l'opzione di confronto su un valore EBCDIC chiamando la stored procedure. `rdsadmin.create_database` Per ulteriori informazioni, consulta rdsadmin.create_database.

   > **Important**
   >
   > Dopo aver creato un database utilizzando la stored procedure, non è possibile modificare la sequenza di confronto. Se desideri che un database utilizzi una sequenza di confronto diversa, elimina il database chiamando la the section called “rdsadmin.drop_database” stored procedure. Quindi, create un database con la sequenza di confronto richiesta.

Fuso orario locale per istanze database Amazon RDS per Db2

Il fuso orario di un'istanza Amazon RDS DB che esegue Db2 è impostato per impostazione predefinita. L'impostazione predefinita attuale è Universal Coordinated Time (UTC). Per far corrispondere il fuso orario delle tue applicazioni, puoi invece impostare il fuso orario dell'istanza DB su un fuso orario locale.

Puoi impostare il fuso orario quando si crea prima l'istanza database. Puoi creare la tua istanza DB utilizzando AWS Management Console, l'API RDS o il AWS CLI. Per ulteriori informazioni, consulta Creazione di un'istanza database.

Se l'istanza DB fa parte di una distribuzione Multi-AZ, in caso di failover, il suo fuso orario rimane quello locale che hai impostato.

È possibile ripristinare l'istanza DB a un punto temporale specificato dall'utente. L'ora viene visualizzata nel fuso orario locale. Per ulteriori informazioni, consulta Ripristino a un'ora specifica per un'istanza database.

L'impostazione del fuso orario locale sull'istanza DB presenta le seguenti limitazioni:

- Non è possibile modificare il fuso orario di un'istanza DB RDS for Db2 esistente.

- Non è possibile ripristinare uno snapshot da un'istanza database in un fuso orario a un'istanza database in un fuso orario diverso.
- Consigliamo vivamente di non ripristinare un file di backup da un fuso orario a un fuso orario diverso. Se ripristini un file di backup da un fuso orario a un altro, devi controllare le query e le applicazioni per verificare gli effetti della modifica del fuso orario.

Fusi orari disponibili

È possibile utilizzare i seguenti valori per l'impostazione del fuso orario.

Zona	Time zone (Fuso orario)
Africa	Africa/Cairo, Africa/Casablanca, Africa/Harare, Africa/Lagos, Africa/Luanda, Africa/Monrovia, Africa/Nairobi, Africa/Tripoli, Africa/Windhoek
America	America/Araguaina, America/Argentina/Buenos_Aires, America/Asuncion, America/Bogota, America/Caracas, America/Chicago, America/Chihuahua, America/Cuiaba, America/Denver, America/Detroit, America/Fortaleza, America/Godthab, America/Guatemala, America/Halifax, America/Lima, America/Los_Angeles, America/Manaus, America/Matamoros, America/Mexico_City, America/Monterrey, America/Montevideo, America/New_York, America/Phoenix, America/Santiago, America/Sao_Paulo, America/Tijuana, America/Toronto
Asia	Asia/Amman, Asia/Ashgabat, Asia/Baghdad, Asia/Baku, Asia/Bangkok, Asia/Beirut, Asia/Calcutta, Asia/Damasco, Asia/Dhaka, Asia/Hong_Kong, Asia/Irkutsk, Asia/Giacarta, Asia/Gerusalemme, Asia/Kabul, Asia/Karachi, Asia/Kathmandu, Asia/Kolkata, Asia/Krasnoyarsk, Asia/Magadan, Asia/Manila, Asia/Muscat, Asia/Novosibirsk, Asia/Rangoon, Asia/Riyadh, Asia/Seul, Asia/Shanghai, Asia/Singapore, Asia/Taipei, Asia/Tehran, Asia/Tokyo, Asia/Ulaanbaatar, Asia/Vladivostok, Asia/Yakutsk, Asia/Yerevan
Atlantico	Atlantico/Azzorre, Atlantico/Cape_Verde
Australia	Australia/Adelaide, Australia/Brisbane, Australia/Darwin, Australia/Eucla, Australia/Hobart, Australia/Lord_Howe, Australia/Perth, Australia/Sydney
Brasile	Brasile/, Brasile/Est DeNoronha

Zona	Time zone (Fuso orario)
Canada	Canada/Newfoundland, Canada/Saskatchewan
ecc	Ecc./GMT-3
Europa	Europa/Amsterdam, Europa/Atene, Europa/Berlino, Europa/Dublino, Europa/Helsinki, Europa/Kaliningrad, Europa/Londra, Europa/Madrid, Europa/Mosca, Europa/Parigi, Europa/Praga, Europa/Roma, Europa/Sarajevo, Europa/Stoccolma
Pacifico	Pacifico/Apia, Pacifico/Auckland, Pacifico/Chatham, Pacifico/Fiji, Pacifico/Guam, Pacifico/Honolulu, Pacifico/Kiritimati, Pacifico/Marquesas, Pacifico/Samoa, Pacifico/Tongatapu, Pacifico/Wake
USA	Stati Uniti/Alaska, Stati Uniti/Centrali, Stati Uniti/Est-Indiana, Stati Uniti/Orientali, Stati Uniti/Pacifico
UTC	UTC

Prerequisiti per la creazione di un'istanza DB RDS per Db2

I seguenti elementi sono prerequisiti prima di creare un'istanza DB.

Argomenti

Account amministratore

Quando si crea un'istanza DB, è necessario designare un account amministratore per l'istanza. Amazon RDS concede `ACCESSCTRL` l'autorità a questo account amministratore del database locale.

L'account amministratore presenta le seguenti caratteristiche, funzionalità e limitazioni:

- È un utente locale e non un Account AWS.
- Non dispone di autorità a livello di istanza Db2 come`SYSADM`, o. `SYSMAINT` `SYSCTRL`
- Impossibile interrompere o avviare un'istanza Db2.
- Non è possibile eliminare un database Db2 se è stato specificato il nome al momento della creazione dell'istanza DB.
- Ha pieno accesso al database Db2, comprese le tabelle e le viste del catalogo.
- Può creare utenti e gruppi locali utilizzando le stored procedure di Amazon RDS.
- Può concedere e revocare autorità e privilegi.

L'account amministratore può eseguire le seguenti attività:

- Creare, modificare o eliminare istanze DB.
- Crea istantanee DB.
- Avvia i ripristini point-in-time .
- Crea backup automatici delle istantanee del DB.
- Crea backup manuali delle istantanee del DB.
- Usa altre funzionalità di Amazon RDS.

Ulteriori considerazioni

Prima di creare un'istanza DB, considera i seguenti elementi:

- Ogni istanza DB RDS for Db2 può ospitare un singolo database Db2.
- Initial database name (Nome del database iniziale)
  - Se non fornisci un nome di database quando crei un'istanza DB, Amazon RDS non crea un database.
  - Non fornire un nome di database nelle seguenti circostanze:
    - Vuoi utilizzare le stored procedure di Amazon RDS per creare o eliminare un database.
    - Vuoi creare un database che utilizzi una sequenza di confronto EBCDIC. Per ulteriori informazioni, consulta Collazione EBCDIC per database Db2 su Amazon RDS.
    - Vuoi ripristinare i backup da Amazon S3.
    - Stai migrando da o. AIX Windows Per ulteriori informazioni, consulta Migrazione una tantum da AIX o Windows verso gli ambienti Linux.
- Nel modello Bring Your Own License (BYOL), devi prima creare un gruppo di parametri personalizzato che contenga i tuoi e i tuoiIBM Customer ID. IBM Site ID Per ulteriori informazioni, consulta Porta la tua licenza per Db2.
- Nel Marketplace AWS modello di licenza Db2, è necessario un Marketplace AWS abbonamento attivo per la particolare IBM Db2 edizione che si desidera utilizzare. Se non ne hai già uno, abbonati a Db2 Marketplace AWS per l'IBM Db2edizione che desideri utilizzare. Per ulteriori informazioni, consulta Licenza Db2 tramite Marketplace AWS.

Connessione all'istanza DB RDS for Db2

Dopo aver effettuato il provisioning di Amazon RDS per l'istanza DB di RDS per Db2, puoi utilizzare qualsiasi applicazione client SQL standard per connetterti all'istanza DB. Poiché Amazon RDS è un servizio gestito, non puoi accedere come`SYSADM`, `SYSCTRLSECADM`, o`SYSMAINT`.

Puoi connetterti a un'istanza DB che esegue il motore di IBM Db2 database utilizzandoIBM Db2 CLP, IBM CLPPlusDBeaver, oIBM Db2 Data Management Console.

Argomenti

Individuazione dell'endpoint dell'istanza DB RDS for Db2

Ogni istanza database Amazon RDS dispone di un endpoint e ciascun endpoint è associato a un nome DNS e a un numero di porta per l'istanza database. Per connetterti alla tua istanza DB con un'applicazione client SQL, hai bisogno del nome DNS e del numero di porta dell'istanza DB.

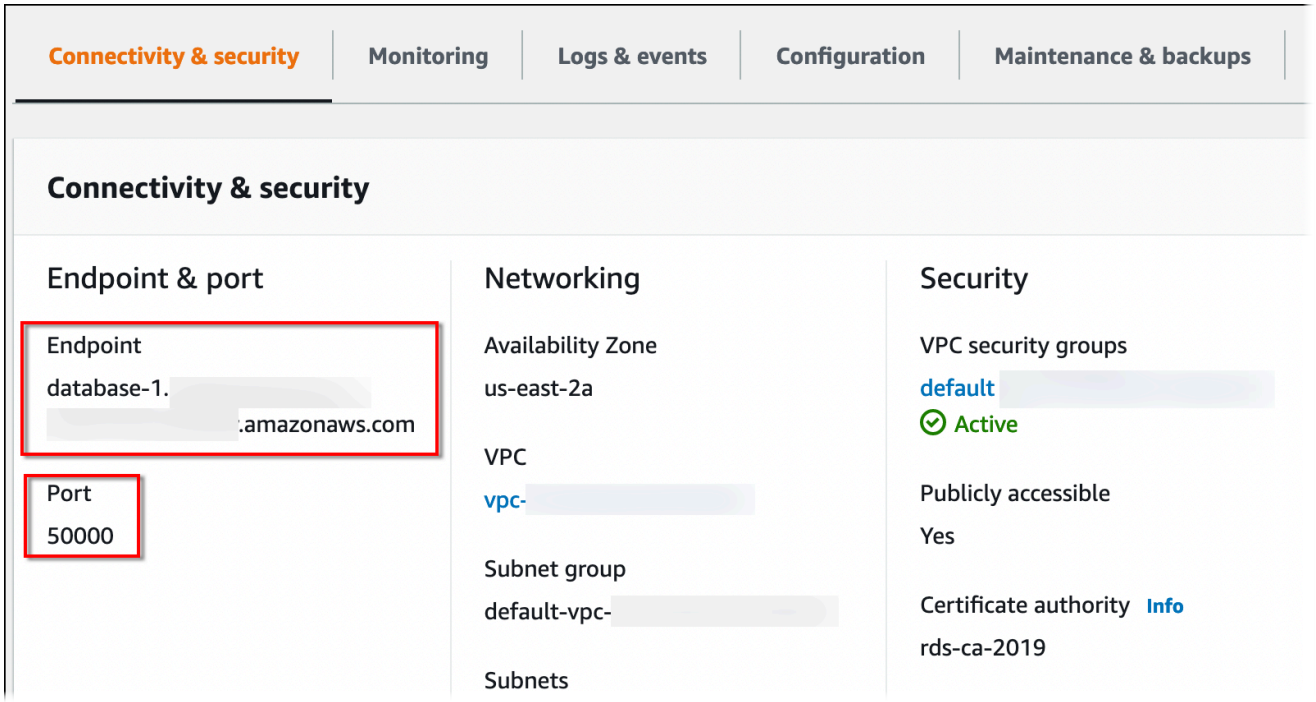
È possibile trovare l'endpoint per un'istanza DB utilizzando AWS Management Console o il. AWS CLI

Console

Per trovare l'endpoint di un'istanza DB RDS for Db2

1. Accedi alla AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nell'angolo in alto a destra della console, scegli la tua istanza DB. Regione AWS
3. Trova il nome DNS e il numero di porta per la tua istanza DB RDS for Db2.
    a. Scegliere Databases (Database) per visualizzare un elenco di istanze database.
    b. Scegli il nome dell'istanza DB RDS for Db2 per visualizzare i dettagli dell'istanza.

c. Nella scheda Connectivity & security (Connettività e sicurezza), copiare l'endpoint. Annotare anche il numero di porta. L'endpoint e il numero di porta sono necessari per la connessione all'istanza database.



AWS CLI

Per trovare l'endpoint di un'istanza DB RDS for Db2, esegui il comando. describe-db-instances Nell'esempio seguente, sostituisci *database-1* con il nome dell'istanza DB.

PerLinux, o: macOS Unix

```
aws rds describe-db-instances \
    --db-instance-identifier database-1 \
    --query 'DBInstances[].
{DBInstanceIdentifier:DBInstanceIdentifier,DBName:DBName,Endpoint:Endpoint}' \
    --output json
```

Per Windows:

```
aws rds describe-db-instances ^
    --db-instance-identifier database-1 ^
    --query 'DBInstances[].
{DBInstanceIdentifier:DBInstanceIdentifier,DBName:DBName,Endpoint:Endpoint}' ^
```

```
    --output json
```

Questo comando produce un output simile all'esempio seguente. La riga `Address` nell'output contiene il nome DNS.

```
[
    {
        "DBInstanceIdentifier": "database-1",
        "DBName": "DB2DB",
        "Endpoint": {
            "Address": "database-1.123456789012.us-east-2.amazonaws.com",
            "Port": 50000,
            "HostedZoneId": "Z2OC4A7DETW6VH"
        }
    }
]
```

Connessione all'istanza DB RDS for Db2 con IBM Db2 CLP

Puoi utilizzare un'utilità da riga di comando, ad esempio IBM Db2 CLP per connetterti ad Amazon RDS per istanze DB Db2. Questa utilità fa parte di. IBM Data Server Runtime Client Per scaricare il client daIBM Fix Central, consulta IBMData Server Client Packages versione 11.5 Mod 8 Fix Pack 0 in IBM Support.

Argomenti

- Terminologia
- Installazione del client
- Connessione a un'istanza database
- Risoluzione dei problemi di connessione all'istanza DB RDS for Db2

Terminologia

I termini seguenti aiutano a spiegare i comandi utilizzati per la connessione all'istanza DB RDS for Db2.

nodo tcpip del catalogo

> Questo comando registra un nodo di database remoto con un client Db2 locale, che rende il nodo accessibile all'applicazione client. Per catalogare un nodo, si forniscono informazioni come

il nome host del server, il numero di porta e il protocollo di comunicazione. Il nodo catalogato rappresenta quindi un server di destinazione in cui risiedono uno o più database remoti. Per ulteriori informazioni, vedete CATALOG TCPIP/TCPIP4/TCPIP6 NODEil comando nella IBM Db2 documentazione.

database di cataloghi

Questo comando registra un database remoto con un client Db2 locale, che rende il database accessibile all'applicazione client. Per catalogare un database, si forniscono informazioni come l'alias del database, il nodo su cui risiede e il tipo di autenticazione necessario per connettersi al database. Per ulteriori informazioni, consulta CATALOG DATABASEil comando nella IBM Db2 documentazione.

Installazione del client

Dopodownloading the package for Linux, installa il client utilizzando i privilegi di root o amministratore.

> **Note**
>
> Per installare il client su AIX oWindows, segui la stessa procedura ma modifica i comandi del tuo sistema operativo.

Per installare il client su Linux

1. Esegui **`./db2_install -f sysreq`**e scegli **`yes`**di accettare la licenza.
2. Scegli la posizione in cui installare il client.
3. Esegui **`clientInstallDir/instance/db2icrt -s client`***`instance_name`*. Sostituisci *`instance_name`* con un utente valido del sistema operativo su. Linux InLinux, il nome dell'istanza DB Db2 è legato al nome utente del sistema operativo.

   Questo comando crea una **`sqllib`**directory nella home directory dell'utente designato suLinux.

Connessione a un'istanza database

Per connetterti alla tua istanza DB RDS for Db2, hai bisogno del nome DNS e del numero di porta. Per informazioni su come trovarli, consulta. Ricerca dell'endpoint È inoltre necessario conoscere

il nome del database, il nome utente principale e la password principale definiti al momento della creazione dell'istanza DB RDS for Db2. Per ulteriori informazioni su come trovarli, consulta. Creazione di un'istanza database

Per connettersi a un'istanza DB RDS for Db2 con IBM Db2 CLP

1. Accedi con il nome utente specificato durante l'installazione del IBM Db2 CLP client.
2. Cataloga la tua istanza DB RDS for Db2. Nell'esempio seguente, sostituite *node_name*, *dns_name e port con un nome* per il nodo nel catalogo locale, il nome DNS dell'istanza DB e il numero di porta.

   ```
   db2 catalog TCPIP node node_name remote dns_name server port
   ```

   Esempio

   ```
   db2 catalog TCPIP node remnode remote database-1.123456789012.us-
   east-1.amazonaws.com server 50000
   ```

3. Catalogate il database e il vostro database. `rdsadmin` Ciò ti consentirà di connetterti al `rdsadmin` database per eseguire alcune attività amministrative utilizzando le stored procedure di Amazon RDS. Per ulteriori informazioni, consulta Amministrazione dell'istanza DB RDS for Db2.

   Nell'esempio seguente, sostituisci *database_alias*, *node_name e database_name con alias per questo database, il nome* del nodo definito nel passaggio precedente e il nome del tuo database. `server_encrypt`crittografa il nome utente e la password sulla rete.

   ```
   db2 catalog database rdsadmin [ as database_alias ] at node node_name
    authentication server_encrypt

   db2 catalog database database_name [ as database_alias ] at node node_name
    authentication server_encrypt
   ```

   Esempio

   ```
   db2 catalog database rdsadmin at node remnode authentication server_encrypt

   db2 catalog database testdb as rdsdb2 at node remnode authentication server_encrypt
   ```

4. Connect al database RDS for Db2. Nell'esempio seguente, sostituisci *rds_database_alias, master_username e master_password con il nome del database, il nome utente principale e la password* principale dell'istanza DB RDS for Db2.

```
db2 connect to rds_database_alias user master_username using master_password
```

   Questo comando produce un output simile all'esempio seguente:

```
Database Connection Information

    Database server        = DB2/LINUXX8664 11.5.9.0
    SQL authorization ID   = ADMIN
    Local database alias   = TESTDB
```

5. Esegui interrogazioni e visualizza i risultati. L'esempio seguente mostra un'istruzione SQL che seleziona il database creato.

```
db2 "select current server from sysibm.dual"
```

   Questo comando produce un output simile all'esempio seguente:

```
1
    ------------------
    TESTDB

    1 record(s) selected.
```

Risoluzione dei problemi di connessione all'istanza DB RDS for Db2

Se ricevi il seguente `NULLID` errore, in genere indica che le versioni del client e del server RDS for Db2 non corrispondono. Per le versioni client Db2 supportate, consulta Combinazioni supportate di client, driver e livelli di server nella documentazione. IBM Db2

```
db2 "select * from syscat.tables"
SQL0805N Package "NULLID.SQLC2O29 0X4141414141454A69" was not found.
SQLSTATE=51002
```

Dopo aver ricevuto questo errore, è necessario associare i pacchetti dal vecchio client Db2 a una versione del server Db2 supportata da RDS per Db2.

Per associare i pacchetti da un vecchio client Db2 a un server Db2 più recente

1. Individua i file di associazione sul computer client. In genere, questi file si trovano nella directory bnd del percorso di installazione del client Db2 e hanno l'estensione.bnd.
2. Connect al server Db2. Nell'esempio seguente, sostituisci *database_name con il nome* del tuo server Db2. *Sostituisci master_username e master_password con le tue informazioni.* Questo utente ha l'autorità. DBADM

   ```
   db2 connect to database_name user master_username using master_password
   ```

3. Esegui il bind comando per associare i pacchetti.

   a. Passa alla directory in cui sono presenti i file di associazione sul computer client.

   b. Esegui il bind comando per ogni file.

      Sono richieste le seguenti opzioni:

      - `blocking all`— Associa tutti i pacchetti nel file bind in un'unica richiesta al database.
      - `grant public`— Concede il permesso di `public` eseguire il pacchetto.
      - `sqlerror continue`— Speciifica che il bind processo continua anche in caso di errori.

      Per ulteriori informazioni sul bind comando, vedete BINDil comando nella IBM Db2 documentazione.

4. Verificate che l'associazione sia avvenuta correttamente interrogando la vista del `syscat.package` catalogo o controllando il messaggio restituito dopo il bind comando.

Per ulteriori informazioni, vedere DB2 v11.5 Bind File and Package Name List in Support. IBM

Connessione all'istanza DB RDS for Db2 con IBM CLPPlus

Puoi utilizzare un'utilità come la connessione IBM CLPPlus a un'istanza database Amazon RDS for Db2. Questa utilità fa parte di. IBM Data Server Runtime Client Per scaricare il client daIBM Fix Central, consulta IBMData Server Client Packages versione 11.5 Mod 8 Fix Pack 0 in IBM Support.

> **Important**
>
> Ti consigliamo di utilizzare un sistema operativo che supporti interfacce utente grafiche come macOSWindows, o Linux con Desktop. IBM CLPPlus Se stai eseguendo headlessLinux, usa switch -nw con i comandi. CLPPlus

Argomenti

Installazione del client

Dopo aver scaricato il pacchetto perLinux, installa il client.

> **Note**
>
> Per installare il client su AIX oWindows, segui la stessa procedura ma modifica i comandi del tuo sistema operativo.

Per installare il client su Linux

1. Esegui **`./db2_install`**.
2. Esegui **`clientInstallDir/instance/db2icrt -s client`***`instance_name`*. Sostituisci *`instance_name`* con un utente valido del sistema operativo su. Linux InLinux, il nome dell'istanza DB Db2 è legato al nome utente del sistema operativo.

   Questo comando crea una **`sqllib`**directory nella home directory dell'utente designato suLinux.

Connessione a un'istanza database

Per connetterti alla tua istanza DB RDS for Db2, hai bisogno del nome DNS e del numero di porta. Per informazioni su come trovarli, consulta. Ricerca dell'endpoint È inoltre necessario conoscere il nome del database, il nome utente principale e la password principale definiti al momento della creazione dell'istanza DB RDS for Db2. Per ulteriori informazioni su come trovarli, consulta. Creazione di un'istanza database

Per connettersi a un'istanza DB RDS for Db2 con IBM CLPPlus

1. Esamina la sintassi del comando. Nell'esempio seguente, sostituisci *clientDir* con la posizione in cui è installato il client.

```
cd clientDir/bin
    ./clpplus -h
```

2. Configura il tuo server Db2. Nell'esempio seguente, sostituisci *dns_name, database_name, endpoint e port con il nome* DNS, il nome del database, l'*endpoint* e la porta per l'istanza DB RDS for Db2. Per ulteriori informazioni, consulta Individuazione dell'endpoint dell'istanza DB RDS for Db2.

```
db2cli writecfg add -dsn dns_name -database database_name -host endpoint -port port
 -parameter "Authentication=SERVER_ENCRYPT"
```

3. Connect alla tua istanza DB RDS for Db2. Nell'esempio seguente, sostituite *master_username e dns_name con il nome utente principale e il nome* DNS.

```
./clpplus -nw master_username@dns_name
```

4. Si apre una finestra. Java Shell Inserisci la password principale per la tua istanza DB RDS for Db2.

> **Note**
>
> Se una Java Shell finestra non si apre, esegui **`./clpplus -nw`** per utilizzare la stessa finestra della riga di comando.

```
Enter password: *********
```

Viene stabilita una connessione e produce un output simile al seguente esempio:

```
Database Connection Information :
---------------------------------
Hostname = database-1.abcdefghij.us-east-1.rds.amazonaws.com
Database server = DB2/LINUXX8664  SQL110590
```

```
SQL authorization ID = admin
Local database alias = DB2DB
Port = 50000
```

5. Esegui interrogazioni e visualizza i risultati. L'esempio seguente mostra un'istruzione SQL che seleziona il database creato.

```
SQL > select current server from sysibm.dual;
```

Questo comando produce un output simile all'esempio seguente:

```
1
    --------------------
    DB2DB
    SQL>
```

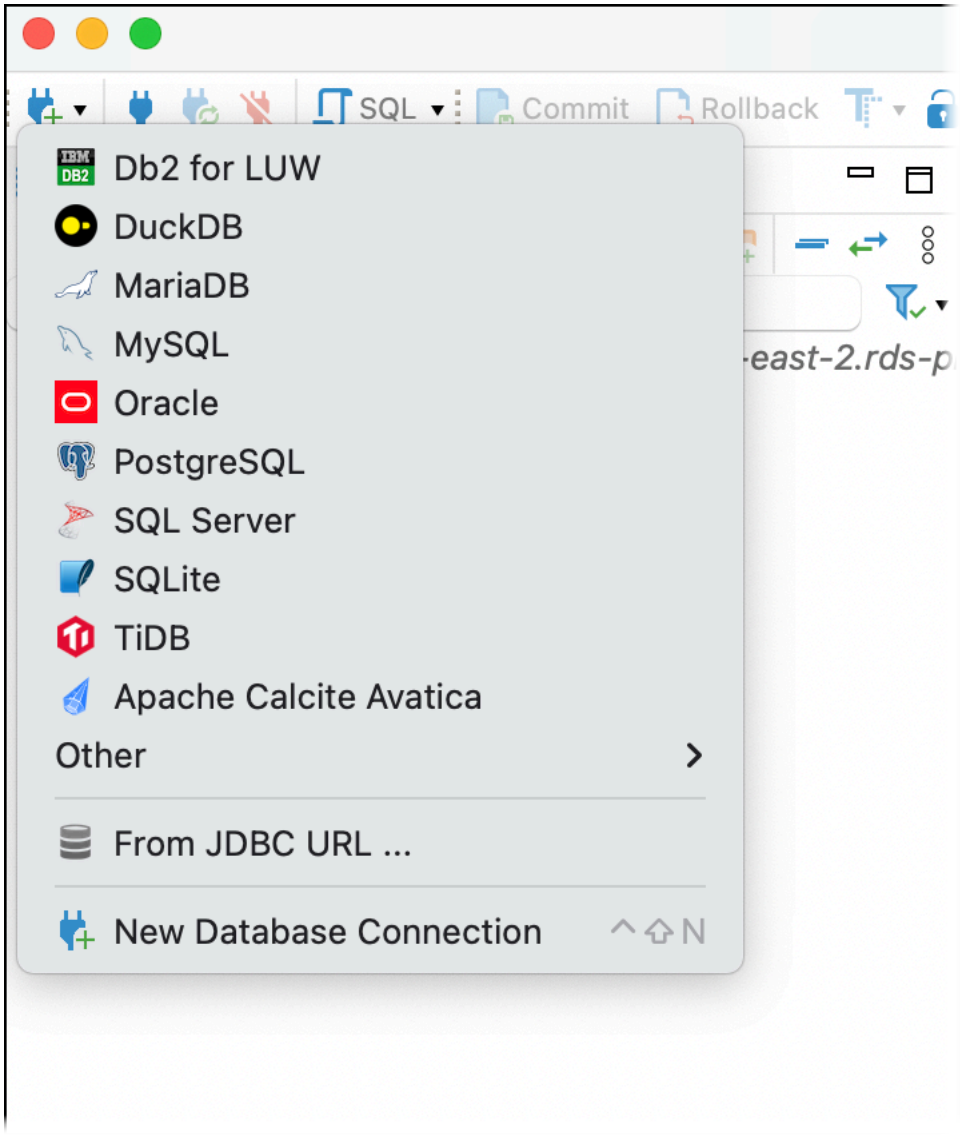
Connessione all'istanza DB RDS for Db2 con DBeaver

Puoi utilizzare strumenti di terze parti, ad esempio DBeaver per connetterti ad Amazon RDS per istanze DB Db2. Per scaricare questa utilità, consulta Community. DBeaver

Per connetterti alla tua istanza DB RDS for Db2, hai bisogno del nome DNS e del numero di porta. Per informazioni su come trovarli, consulta. Ricerca dell'endpoint È inoltre necessario conoscere il nome del database, il nome utente principale e la password principale definiti al momento della creazione dell'istanza DB RDS for Db2. Per ulteriori informazioni su come trovarli, consulta. Creazione di un'istanza database

Per connettersi a un'istanza DB RDS for Db2 con DBeaver

1. Avvia DBeaver.
2. Scegli l'icona Nuova connessione nella barra degli strumenti, quindi scegli Db2 for LUW.



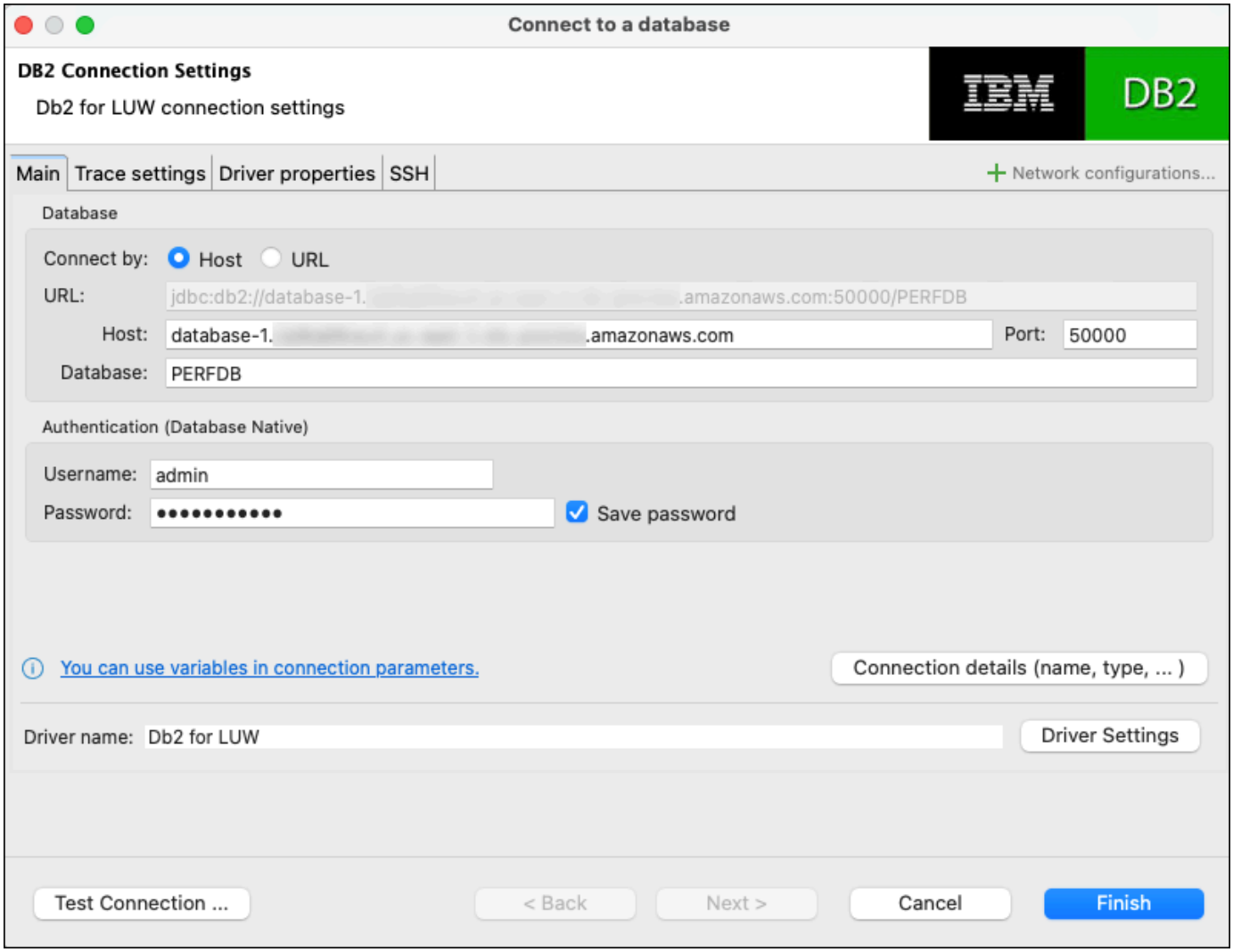
3. Nella finestra Connect to a database, fornisci informazioni per la tua istanza DB RDS for Db2.

   a. Immetti le seguenti informazioni:

      - Per Host, inserisci il nome DNS dell'istanza DB.
      - Per Port, inserisci il numero di porta per l'istanza DB.
      - Per Database, inserisci il nome del database.
      - Per Username (Nome utente) inserire il nome dell'amministratore di database per l'istanza database.
      - Per Password, inserisci la password dell'amministratore del database per l'istanza DB.

   b. Seleziona Salva password.

   c. Scegli Impostazioni driver.



4. Nella finestra Modifica driver, specificare proprietà di sicurezza aggiuntive.

   a. Scegli la scheda Proprietà del driver.

   b. Aggiungi due proprietà utente.

      i. Apri il menu contestuale (fai clic con il pulsante destro del mouse), quindi scegli Aggiungi nuova proprietà.

      ii. Per Property Name, aggiungete EncryptionAlgorithm, quindi scegliete OK.

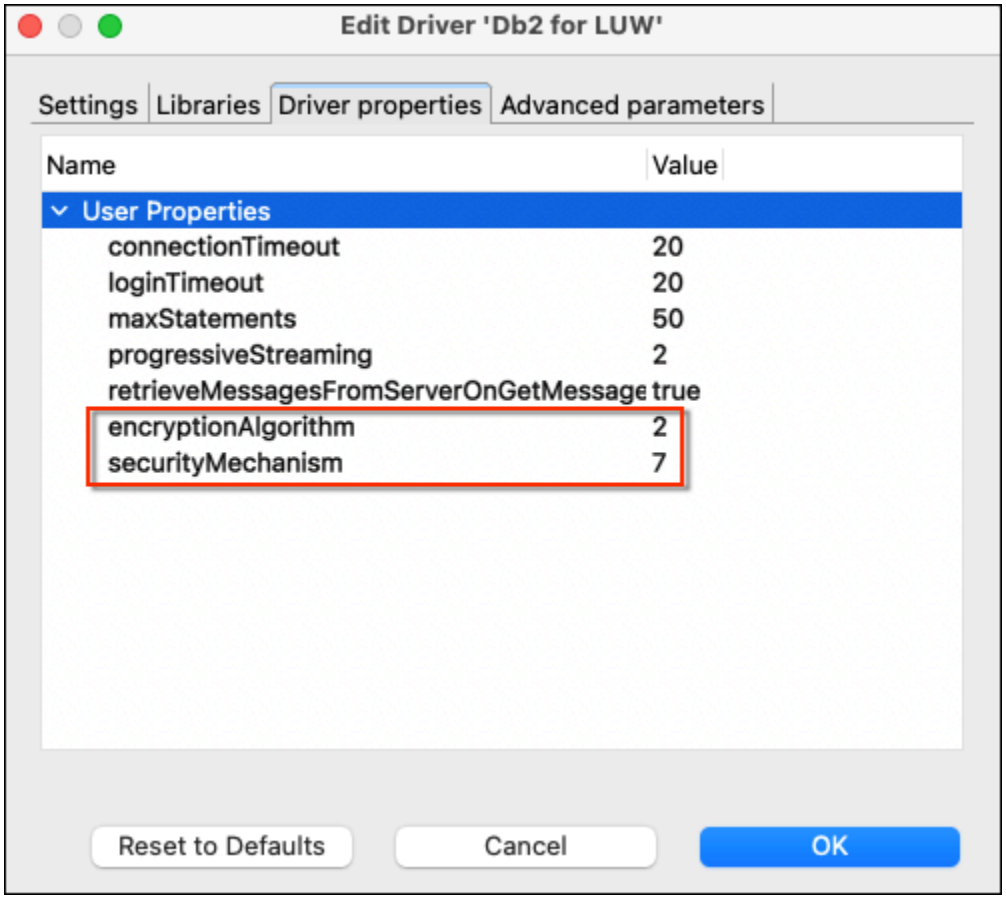
      iii. Con la riga EncryptionAlgorithm selezionata, scegliete la colonna Valore e aggiungete 2.

      iv. Apri il menu contestuale (fai clic con il pulsante destro del mouse), quindi scegli Aggiungi nuova proprietà.

v. Per Property Name, aggiungete SecurityMechanism, quindi scegliete OK.

vi. Con la riga SecurityMechanism selezionata, scegliete la colonna Valore e aggiungete 7.

c. Scegli OK.



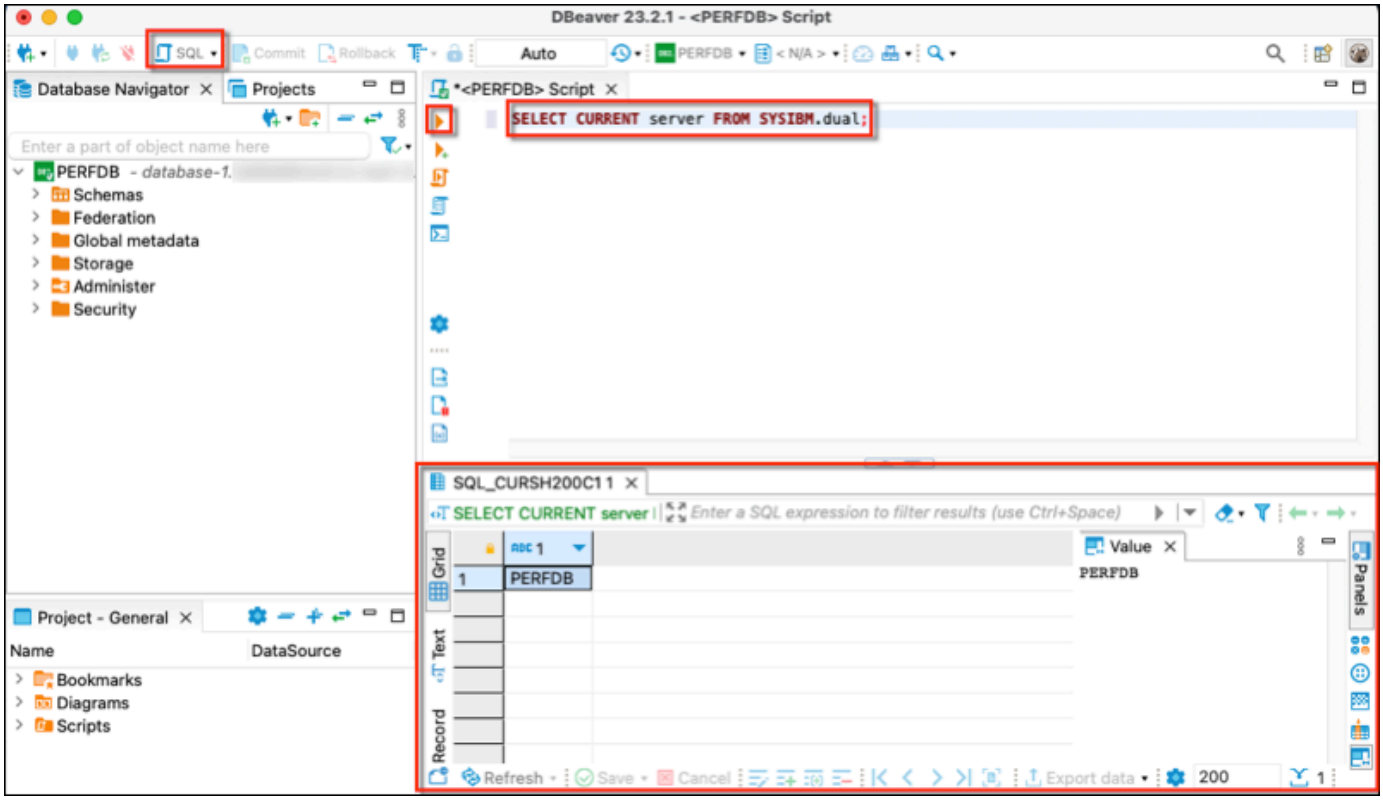
5. Nella finestra Connetti a un database, scegli Test connessione. Se sul computer non è installato un driver DB2 JBDC, il driver viene scaricato automaticamente.
6. Scegli OK.
7. Scegli Fine.
8. Nella scheda Navigazione del database, scegli il nome del database. Ora puoi esplorare gli oggetti.

Ora sei pronto per eseguire i comandi SQL.

Per eseguire comandi SQL e visualizzare i risultati

1. Nel menu in alto, scegli SQL. Si apre un pannello di script SQL.
2. Nel pannello Script, immettete un comando SQL.

3. Per eseguire il comando, scegliete il pulsante Esegui query SQL.
4. Nel pannello dei risultati SQL, visualizza i risultati delle tue query SQL.



Connessione all'istanza DB RDS for Db2 con IBM Db2 Data Management Console

Puoi connetterti alla tua istanza database Amazon RDS for Db2 con. IBM Db2 Data Management Console IBM Db2 Data Management Consolepuò amministrare e monitorare diverse istanze RDS for Db2 DB. Per scaricare questa utilità, consulta le release della IBM Db2 Data Management Console versione 3.1x in IBM Support.

IBM Db2 Data Management Consolerichiede un database Db2 del repository per archiviare i metadati e le metriche delle prestazioni, ma non può creare automaticamente un repository per RDS for Db2.

È innanzitutto necessario creare un database di repository per monitorare uno o più RDS for Db2 DB istanze. Quindi connettiti alla tua istanza DB RDS for Db2 con. IBM Db2 Data Management Console

Argomenti

- Creazione di un database di repository per monitorare le istanze DB
- Connessione a RDS per istanze DB Db2 con IBM Db2 Data Management Console

Creazione di un database di repository per monitorare le istanze DB

È possibile utilizzare un'istanza DB RDS for Db2 esistente di dimensioni adeguate come repository per monitorare altre istanze DB RDS for IBM Db2 Data Management Console Db2. Tuttavia, poiché l'utente amministratore non ha l'`SYSCTRL`autorità per creare pool di buffer e tablespace, l'utilizzo del repository per creare un database di repository non riesce. IBM Db2 Data Management Console È invece necessario creare un database di repository per monitorare l'RDS per le istanze DB Db2. È possibile creare un database di repository in due modi diversi. È possibile creare manualmente un pool di buffer, un tablespace e oggetti per un repository. IBM Db2 Data Management Console Oppure puoi creare un'istanza Amazon EC2 separata per ospitare un IBM Db2 Data Management Console repository.

Argomenti

- Creazione manuale di un buffer pool, di un tablespace e di oggetti
- Creazione di un'istanza Amazon EC2 per ospitare un repository IBM Db2 Data Management Console

Creazione manuale di un buffer pool, di un tablespace e di oggetti

Per creare un buffer pool, un tablespace e oggetti da utilizzare IBM Db2 Data Management Console

1. Consenti i privilegi per il buffer pool e i tablespace.

   a. Apporta modifiche agli script, in particolare per i buffer pool e i tablespace. Per ulteriori informazioni, consulta Configurazione di un database di repository nella documentazione. IBM Db2 Data Management Console

   b. Connect al `rdsadmin` database. Nell'esempio seguente, sostituite *`master_username e master_password`* con le vostre informazioni.

   ```
   db2 connect to rdadmin user master_username using master_password
   ```

   c. Crea un pool di buffer per. IBM Db2 Data Management Console Nell'esempio seguente, sostituisci *`database_name con il nome`* del repository per cui hai creato per monitorare il tuo RDS IBM Db2 Data Management Console per le istanze DB Db2.

   ```
   db2 "call rdsadmin.create_bufferpool('database_name',
        'BP4CONSOLE', 1000, 'Y', 'Y', 16384)"
   ```

d. Crea IBM Db2 Data Management Console un tablespace per. Nell'esempio seguente, sostituisci *database_name con il nome* del repository per cui hai creato per monitorare il tuo RDS IBM Db2 Data Management Console per le istanze DB Db2.

```
db2 "call rdsadmin.create_tablespace('database_name',
     'TS4CONSOLE', 'BP4CONSOLE', 16384)"
```

e. Crea IBM Db2 Data Management Console un tablespace temporaneo per. Nell'esempio seguente, sostituisci *database_name con il nome* del repository per cui hai creato per monitorare il tuo RDS IBM Db2 Data Management Console per le istanze DB Db2.

```
db2 "call rdsadmin.create_tablespace('database_name',
    'TS4CONSOLE_TEMP', 'BP4CONSOLE', 16384, 0, 0, 'T')"
```

2. Crea oggetti manualmente. IBM Db2 Data Management Console Per ulteriori informazioni, consulta Configurazione di un database di repository nella IBM Db2 Data Management Console documentazione.

Creazione di un'istanza Amazon EC2 per ospitare un repository IBM Db2 Data Management Console

Puoi creare un'istanza Amazon Elastic Compute Cloud (Amazon EC2) separata per ospitare un repository. IBM Db2 Data Management Console Per informazioni sulla creazione di un'istanza Amazon EC2, consulta il Tutorial: Get started with Amazon EC2 instances nella Amazon Linux EC2 User Guide for Linux Instances.

Connessione a RDS per istanze DB Db2 con IBM Db2 Data Management Console

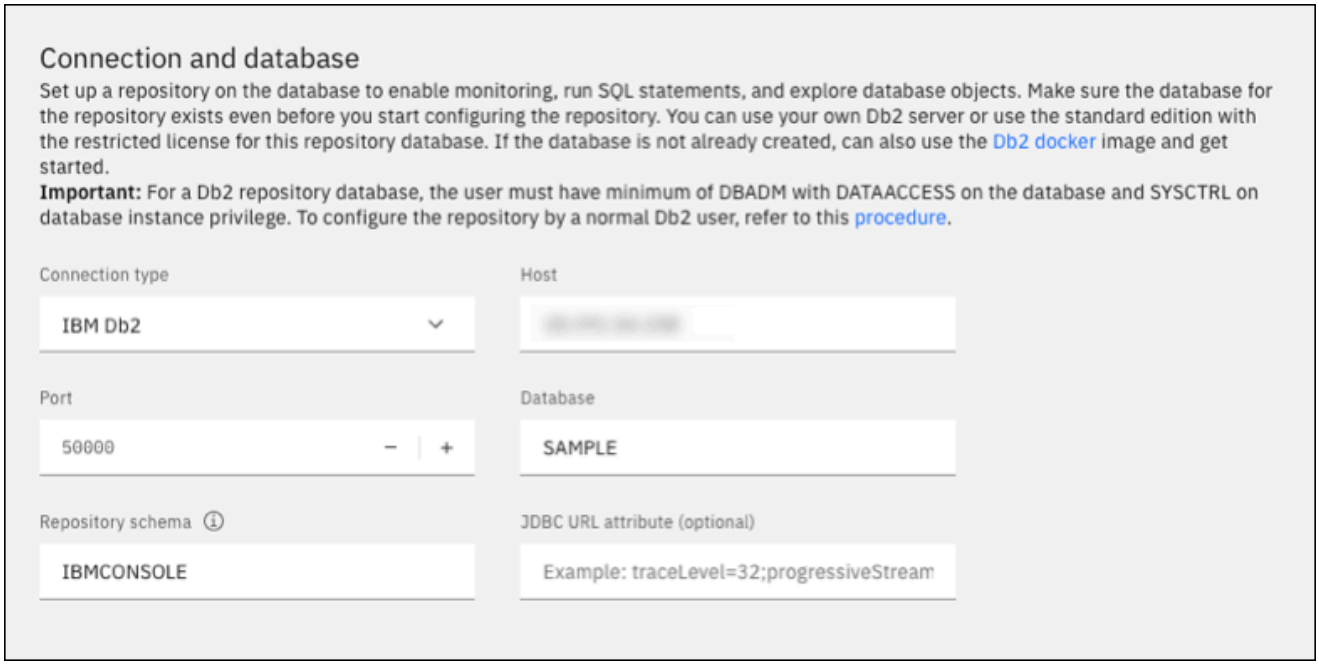
Per connetterti alla tua istanza DB RDS for Db2, hai bisogno del nome DNS e del numero di porta. Per informazioni su come trovarli, consulta. Ricerca dell'endpoint È inoltre necessario conoscere il nome del database, il nome utente principale e la password principale definiti al momento della creazione dell'istanza DB RDS for Db2. Per ulteriori informazioni su come trovarli, consulta. Creazione di un'istanza database Se ti connetti tramite Internet, consenti il traffico verso la porta del database. Per ulteriori informazioni, consulta Creazione di un'istanza database.

Per connetterti a RDS per istanze DB Db2 con IBM Db2 Data Management Console

1. Avvia IBM Db2 Data Management Console.
2. Configura il repository.

a. Nella sezione Connessione e database, inserisci le seguenti informazioni per l'istanza DB RDS for Db2:

   - Per Host, inserisci il nome DNS dell'istanza DB.
   - Per Port, inserisci il numero di porta per l'istanza DB.
   - Per Database, inserisci il nome del database.



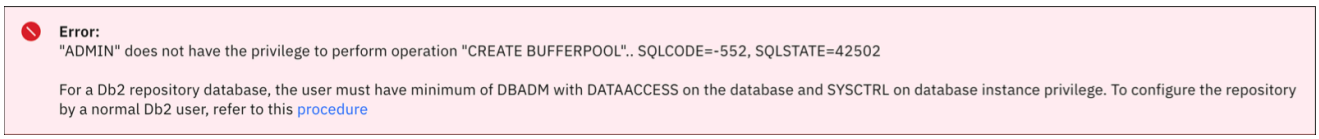
b. Nella sezione Sicurezza e credenziali, inserisci le seguenti informazioni per la tua istanza DB RDS for Db2:

   - Per Tipo di sicurezza, scegli Utente e password crittografati.
   - Per Username (Nome utente) inserire il nome dell'amministratore di database per l'istanza database.
   - Per Password, inserisci la password dell'amministratore del database per l'istanza DB.

c. Scegli Test Connection (Connessione di prova).

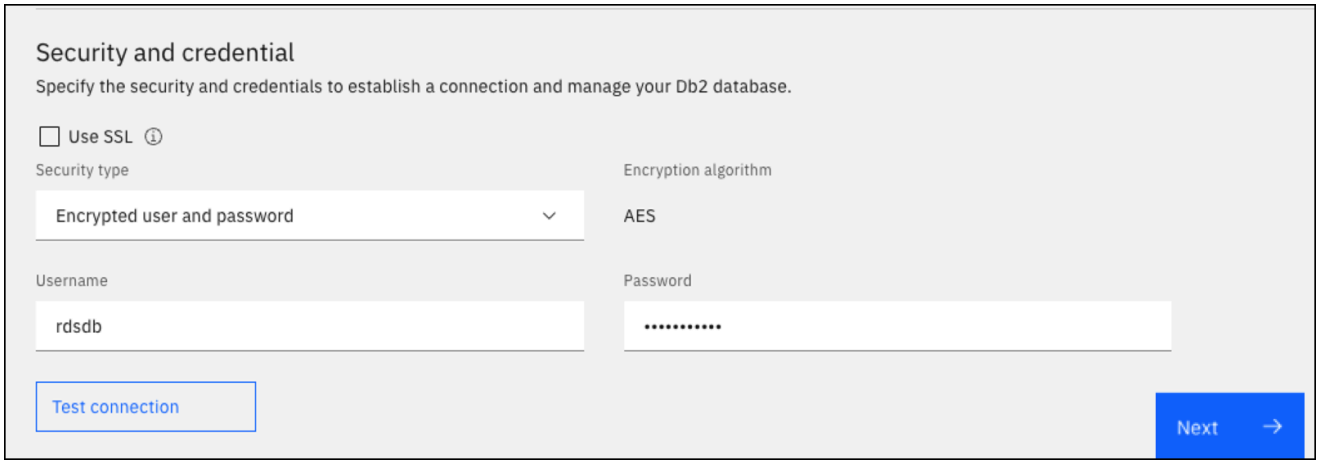
> **Note**
>
> Se la connessione non riesce, conferma che la porta del database sia aperta tramite le regole in entrata del gruppo di sicurezza. Per ulteriori informazioni, consulta Considerazioni per i gruppi di sicurezza.

Il seguente messaggio di errore indica che l'utente amministratore che si connette all'istanza DB RDS for Db2 non dispone dei privilegi per creare pool di buffer o tablespace. Indica inoltre che per i database del repository Db2, l'utente deve disporre di e sul database. `DBADM` `DATACCESS` L'utente deve inoltre disporre del privilegio di `SYSCTRL` istanza del database.



Assicurati di aver creato una tabella di buffer, un tablespace e oggetti per un IBM Db2 Data Management Console repository per monitorare l'istanza DB RDS for Db2. Oppure puoi utilizzare un'istanza DB Amazon EC2 Db2 per ospitare un IBM Db2 Data Management Console repository per monitorare l'istanza DB RDS for Db2. Per ulteriori informazioni, consulta Creazione di un database di repository per monitorare le istanze DB.

d. Dopo aver testato con successo la connessione, scegli Avanti.

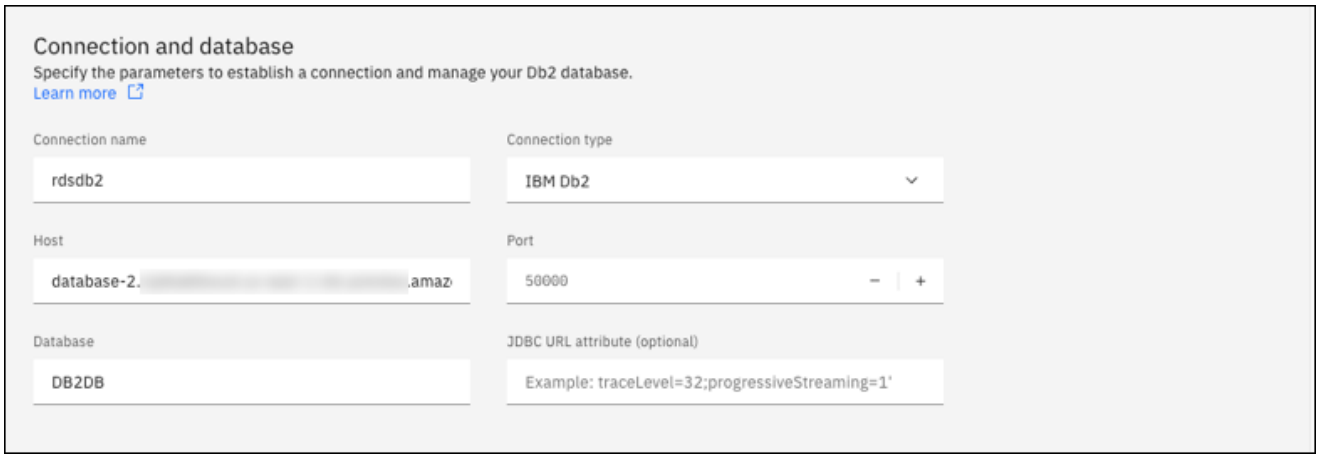


3. Nella finestra di attivazione del monitoraggio degli eventi per l'impostazione delle statistiche, scegli Avanti.

4. (Facoltativo) Aggiungi una nuova connessione. Se desideri utilizzare un'istanza DB RDS for Db2 diversa per l'amministrazione e il monitoraggio, aggiungi una connessione a un'istanza DB RDS for Db2 non repository.

   a. Nella sezione Connessione e database, inserisci le seguenti informazioni per l'istanza DB RDS for Db2 da utilizzare per l'amministrazione e il monitoraggio:

      - Per Nome della connessione, immettere l'identificatore del database Db2.
      - Per Host, inserisci il nome DNS dell'istanza DB.
      - Per Port, inserisci il numero di porta per l'istanza DB.
      - Per Database, inserisci il nome del database.



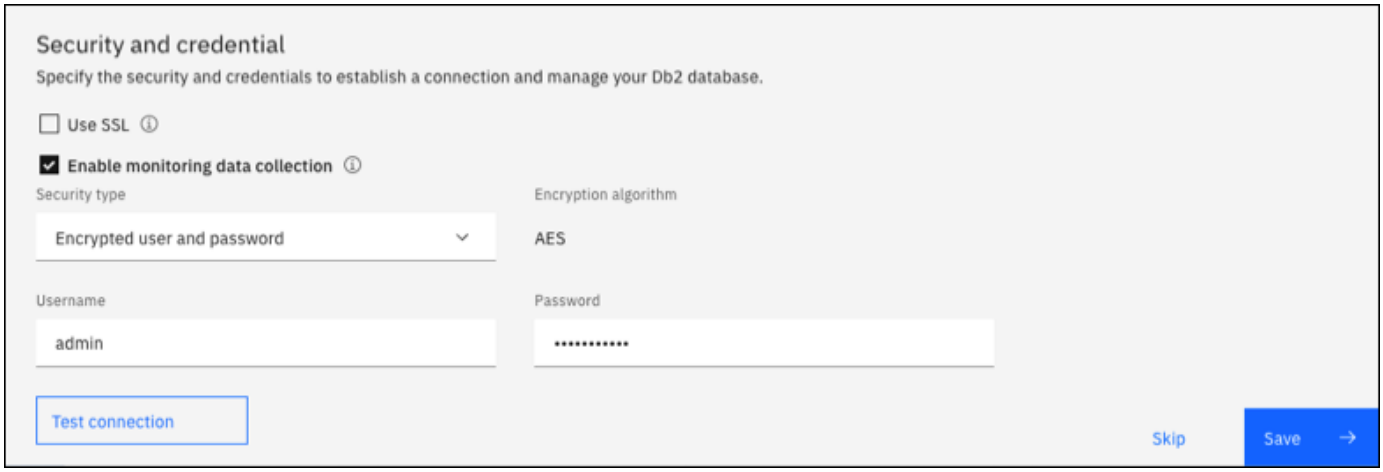
   b. Nella sezione Sicurezza e credenziali, seleziona Abilita la raccolta dei dati di monitoraggio.

   c. Inserisci le seguenti informazioni per la tua istanza DB RDS for Db2:

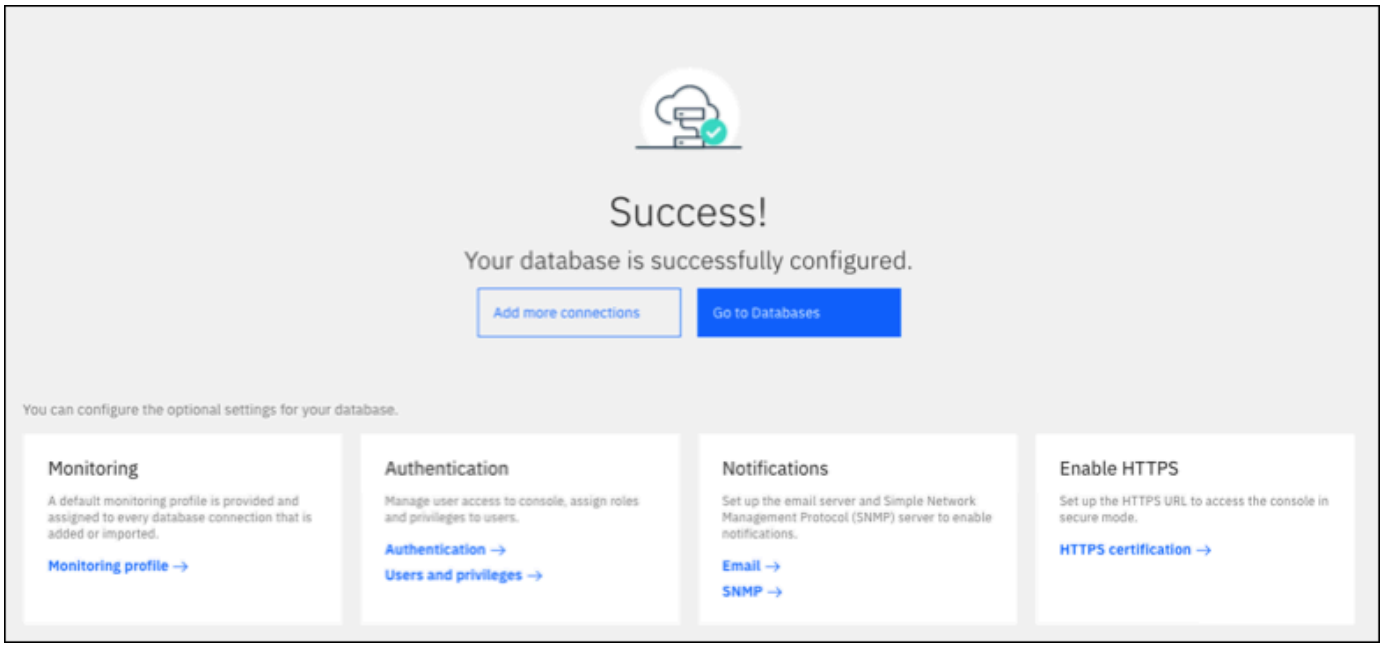
      - Per Username (Nome utente) inserire il nome dell'amministratore di database per l'istanza database.
      - Per Password, inserisci la password dell'amministratore del database per l'istanza DB.

   d. Scegli Test Connection (Connessione di prova).

   e. Dopo aver testato correttamente la connessione, scegli Salva.



Dopo aver aggiunto la connessione, viene visualizzata una finestra simile alla seguente. Questa finestra indica che il database è stato configurato correttamente.



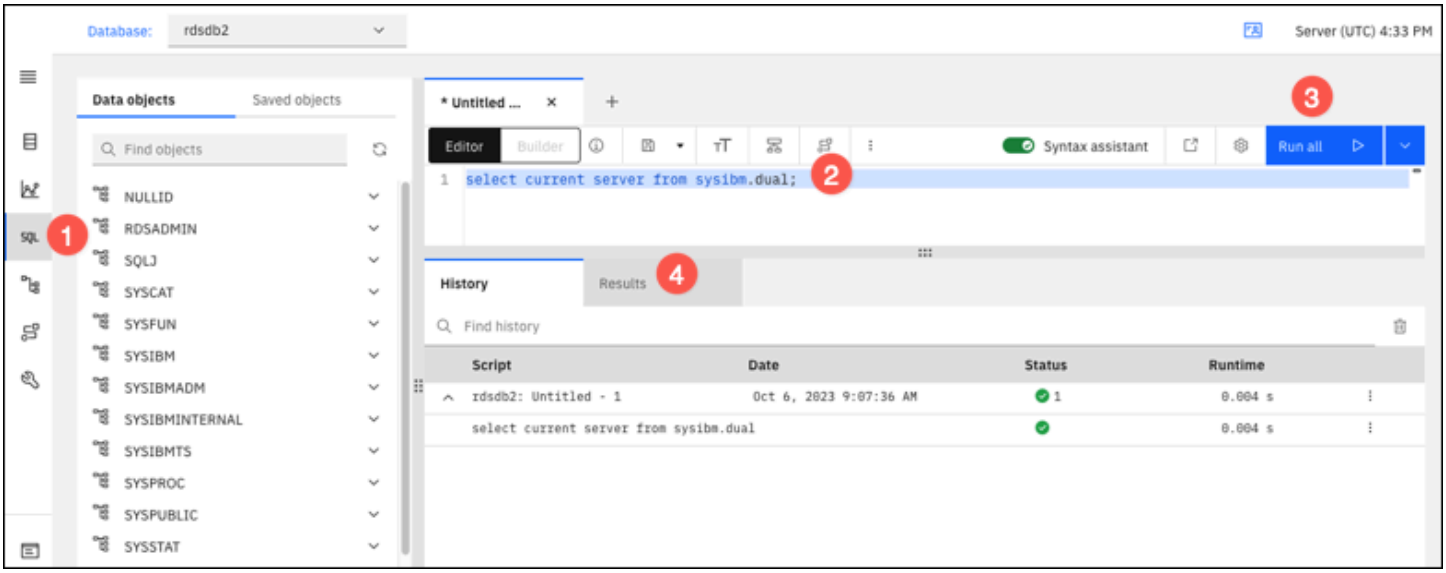
5. Scegli Vai ai database. Viene visualizzata una finestra Database simile alla seguente. Questa finestra è una dashboard che mostra metriche, stati e connessioni.

Ora puoi iniziare a utilizzare IBM Db2 Data Management Console per eseguire i seguenti tipi di attività:

- Gestisci più istanze RDS per istanze DB Db2.
- Eseguire comandi SQL.
- Esplora, crea o modifica dati e oggetti di database.
- Crea `EXPLAIN PLAN` istruzioni in SQL.
- Ottimizza le interrogazioni.

Per eseguire comandi SQL e visualizzare i risultati

1. Nella barra di navigazione a sinistra, scegli SQL.
2. Immettete un comando SQL.
3. Scegli Esegui tutto.
4. Per visualizzare i risultati, scegli la scheda Risultati.



Considerazioni per i gruppi di sicurezza

Per poterti connettere alla tua istanza DB RDS for Db2, questa deve essere associata a un gruppo di sicurezza che contenga gli indirizzi IP e la configurazione di rete necessari. L'istanza DB RDS for Db2 potrebbe utilizzare il gruppo di sicurezza predefinito. Se hai assegnato un gruppo di sicurezza predefinito non configurato quando hai creato l'istanza DB RDS for Db2, il firewall impedisce le connessioni Internet. Per ulteriori informazioni sulla creazione di un nuovo gruppo di sicurezza, consulta Controllo dell'accesso con i gruppi di sicurezza.

Dopo aver creato il nuovo gruppo di sicurezza, modifica l'istanza database per associarla al gruppo di sicurezza. Per ulteriori informazioni, consulta Modifica di un'istanza database Amazon RDS.

Puoi aumentare la sicurezza utilizzando la crittografia SSL per proteggere le connessioni alla tua istanza database. Per ulteriori informazioni, consulta Utilizzo di SSL/TLS con un'istanza DB RDS per Db2.

Protezione di RDS per le connessioni delle istanze DB Db2

Amazon RDS for Db2 supporta modi per migliorare la sicurezza dell'istanza DB RDS for Db2.

Argomenti

Utilizzo di SSL/TLS con un'istanza DB RDS per Db2

SSL è un protocollo standard del settore per proteggere le connessioni di rete tra client e server. Dopo la versione SSL 3.0, il nome è stato cambiato in TLS, ma spesso ci riferiamo ancora al protocollo come SSL. Amazon RDS supporta la crittografia SSL per le istanze database Amazon RDS per Db2. Utilizzando SSL/TLS, puoi crittografare una connessione tra il client dell'applicazione e l'istanza DB RDS for Db2. Il supporto SSL/TLS è disponibile in tutto per RDS for Db2. Regioni AWS

Per abilitare la crittografia SSL/TLS per un'istanza DB RDS for Db2, aggiungi l'opzione SSL Db2 al gruppo di parametri associato all'istanza DB. Amazon RDS utilizza una seconda porta, come richiesto da Db2, per le connessioni SSL/TLS. In questo modo è possibile che tra un'istanza DB e un client Db2 si verifichino contemporaneamente sia comunicazioni in testo non crittografato che comunicazioni crittografate con SSL. Ad esempio, è possibile utilizzare la porta con testo in chiaro per comunicare con altre risorse all'interno di un VPC mentre utilizzi la porta con crittografia SSL per comunicare con risorse all'esterno del VPC.

Argomenti

Creazione di una connessione SSL/TLS

Per creare una connessione SSL/TLS, scegli un'autorità di certificazione (CA), scarica un pacchetto di certificati per tutti Regioni AWS e aggiungi parametri a un gruppo di parametri personalizzato.

Passaggio 1: scegli una CA e scarica un certificato

Scegli un'autorità di certificazione (CA) e scarica un pacchetto di certificati per tuttiRegioni AWS. Per ulteriori informazioni, consulta .

Fase 2: Aggiornare i parametri in un gruppo di parametri personalizzato

> **Important**
>
> Se utilizzi il modello Bring Your Own License (BYOL) per RDS for Db2, modifica il gruppo di parametri personalizzato che hai creato per te e per il tuo. IBM Customer ID IBM Site ID Se utilizzi un modello di licenza diverso per RDS for Db2, segui la procedura per aggiungere parametri a un gruppo di parametri personalizzato. Per ulteriori informazioni, consulta Opzioni di licenza Amazon RDS per Db2.

Non è possibile modificare i gruppi di parametri predefiniti per le istanze DB RDS for Db2. Pertanto, è necessario creare un gruppo di parametri personalizzato, modificarlo e quindi collegarlo alle istanze DB RDS per Db2. Per informazioni sui gruppi di parametri, consultare Utilizzo di gruppi di parametri DB in un'istanza DB.

Utilizzate le impostazioni dei parametri nella tabella seguente.

Parametro	Valore
`DB2COMM`	`TCPIP,SSL`
`SSL_SVCENAME`	`<any port number except the number used for the non-SSL port>`

Per aggiornare i parametri in un gruppo di parametri personalizzato

1. Creare un gruppo di parametri personalizzato eseguendo il create-db-parameter-groupcomando.

   Includi le seguenti opzioni obbligatorie:

   - `--db-parameter-group-name`— Un nome per il gruppo di parametri che state creando.
   - `--db-parameter-group-family`— L'edizione e la versione principale del motore Db2. Valori validi: `db2-se-11-5`, `db2-ae-11.5`.
   - `--description`— Una descrizione per questo gruppo di parametri.

   Per ulteriori informazioni sulla creazione di un gruppo di parametri database, consulta Creazione di un gruppo di parametri del database.

2. Modificate i parametri nel gruppo di parametri personalizzato creato eseguendo il modify-db-parameter-groupcomando.

   Includi le seguenti opzioni obbligatorie:

   - `--db-parameter-group-name`— Il nome del gruppo di parametri creato.
   - `--parameters`— Una matrice di nomi di parametri, valori e metodi di applicazione per l'aggiornamento dei parametri.

   Per ulteriori informazioni sulla modifica di un gruppo di parametri, vedereModifica di parametri in un gruppo di parametri del database.
3. Associate il gruppo di parametri alla vostra istanza DB RDS for Db2. Per ulteriori informazioni, consulta Associazione di un gruppo di parametri database a un'istanza database.

Connect al server del database Db2

Le istruzioni per la connessione al server di database Db2 sono specifiche della lingua.

Java

Per connettersi al server di database Db2 utilizzando Java

1. Scarica il driver JDBC. Per ulteriori informazioni, consulta Versioni e download dei driver JDBC DB2 nella documentazione di SupportIBM.
2. Crea un file di shell script con il seguente contenuto. Questo script aggiunge tutti i certificati del pacchetto a unJava KeyStore.

   > **Important**
   >
   > Verifica che `keytool` esista nel percorso dello script in modo che lo script possa localizzarlo. Se si utilizza un client Db2, è possibile individuare quanto seguekeytool. `~sqlib/java/jdk64/jre/bin`

```
#!/bin/bash
PEM_FILE=$1
PASSWORD=$2
KEYSTORE=$3
```

```
# number of certs in the PEM file
CERTS=$(grep 'END CERTIFICATE' $PEM_FILE| wc -l)
for N in $(seq 0 $(($CERTS - 1))); do
    ALIAS="${PEM_FILE%.*}-$N"
    cat $PEM_FILE |
    awk "n==$N { print }; /END CERTIFICATE/ { n++ }" |
    keytool -noprompt -import -trustcacerts -alias $ALIAS -keystore $KEYSTORE -
storepass $PASSWORD
done
```

3. Per eseguire lo script di shell e importare il PEM file con il pacchetto di certificati in un fileJava KeyStore, esegui il comando seguente. Sostituisci *shell_file_name.sh* con il nome del tuo file di script di shell e *la password* con la password per il tuoJava KeyStore.

```
./shell_file_name.sh global-bundle.pem password truststore.jks
```

4. Per connetterti al tuo server Db2, esegui il seguente comando. Sostituisci i seguenti segnaposto nell'esempio con le informazioni sull'istanza DB RDS for Db2.

   - *ip_address — L'indirizzo* IP per l'endpoint dell'istanza DB.
   - *port* — Il numero di porta per la connessione SSL. Può essere qualsiasi numero di porta tranne il numero utilizzato per la porta non SSL.
   - *database_name — Il nome* del database nell'istanza DB.
   - *master_username* — Il nome utente principale per l'istanza DB.
   - *master_password — La password* principale per l'istanza DB.

```
export trustStorePassword=MyPassword
java -cp ~/dsdriver/jdbc_sqlj_driver/linuxamd64/db2jcc4.jar \
com.ibm.db2.jcc.DB2Jcc -url \
"jdbc:db2://ip_address:port/database_name:\
sslConnection=true;sslTrustStoreLocation=\
~/truststore.jks;\
sslTrustStorePassword=${trustStorePassword};\
sslVersion=TLSv1.2;\
encryptionAlgorithm=2;\
securityMechanism=7;" \
-user master_username -password master_password
```

Node.js

Per connettersi al server del database Db2 tramite Node.js

1. Installa il node-ibm_dbdriver. Per ulteriori informazioni, consulta Installazione del driver node-ibm_db su sistemi Linux e UNIX nella documentazione. IBM Db2
2. Crea un JavaScript file basato sul seguente contenuto. Sostituisci i seguenti segnaposto dell'esempio con le informazioni sull'istanza DB RDS for Db2.

   - *ip_address — L'indirizzo* IP per l'endpoint dell'istanza DB.
   - *master_username* — Il nome utente principale per l'istanza DB.
   - *master_password — La password* principale per l'istanza DB.
   - *database_name — Il nome* del database nell'istanza DB.
   - *port* — Il numero di porta per la connessione SSL. Può essere qualsiasi numero di porta tranne il numero utilizzato per la porta non SSL.

```
var ibmdb = require("ibm_db");
const hostname = "ip_address";
const username = "master_username";
const password = "master_password";
const database = "database_name";
const port = "port";
const certPath = "/root/qa-bundle.pem";
ibmdb.open("DRIVER={DB2};DATABASE=" + database + ";HOSTNAME=" +
 hostname + ";UID=" + username + ";PWD=" + password + ";PORT=" + port +
 ";PROTOCOL=TCPIP;SECURITY=SSL;SSLServerCertificate=" + certPath + ";", function
 (err, conn){
 if (err) return console.log(err);
 conn.close(function () {
 console.log('done');
 });
});
```

3. Per eseguire il JavaScript file, esegui il comando seguente.

```
node ssl-test.js
```

Python

Per connettersi al server del database Db2 utilizzando Python

1. Crea un Python file con il seguente contenuto. Sostituisci i seguenti segnaposto dell'esempio con le informazioni sull'istanza DB RDS for Db2.

   - *port*: il numero di porta per la connessione SSL. Può essere qualsiasi numero di porta tranne il numero utilizzato per la porta non SSL.
   - *master_username* — Il nome utente principale per l'istanza DB.
   - *master_password — La password* principale per l'istanza DB.
   - *database_name — Il nome* del database nell'istanza DB.
   - *ip_address — L'indirizzo* IP per l'endpoint dell'istanza DB.

```
import click
import ibm_db
import sys

port = port;
master_user_id = "master_username" # Master id used to create your DB instance
master_password = "master_password" # Master password used to create your DB
 instance
db_name = "database_name" # If not given "db-name'
vpc_customer_private_ip = "ip_address" # Hosts end points - Customer private IP
 Addressicert_path = "/root/ssl/global-bundle.pem" # cert path

@click.command()
@click.option("--path", help="certificate path")
def db2_connect(path):

    try:
        conn =
 ibm_db.connect(f"DATABASE={db_name};HOSTNAME={vpc_customer_private_ip};PORT={port};

 PROTOCOL=TCPIP;UID={master_user_id};PWD={master_password};SECURITY=ssl;SSLServerCertifi
 "", "")
        try:
            ibm_db.exec_immediate(conn, 'create table tablename (a int);')
            print("Query executed successfully")
        except Exception as e:
```

```
            print(e)
        finally:
            ibm_db.close(conn)
            sys.exit(1)
    except Exception as ex:
        print("Trying to connect...")

if __name__ == "__main__":
    db2_connect()
```

2. Crea il seguente script di shell, che esegue il Python file che hai creato. Sostituiscilo *python_file_name.py*con il nome del tuo file di Python script.

```
#!/bin/bash
PEM_FILE=$1
# number of certs in the PEM file
CERTS=$(grep 'END CERTIFICATE' $PEM_FILE| wc -l)

for N in $(seq 0 $(($CERTS - 1))); do
    ALIAS="${PEM_FILE%.*}-$N"
    cert=`cat $PEM_FILE | awk "n==$N { print }; /END CERTIFICATE/ { n++ }"`
    cat $PEM_FILE | awk "n==$N { print }; /END CERTIFICATE/ { n++ }" >
 $ALIAS.pem
    python3 <python_file_name.py> --path $ALIAS.pem
    output=`echo $?`
    if [ $output == 1 ]; then
        break
    fi
done
```

3. Per importare il PEM file con il pacchetto di certificati ed eseguire lo script di shell, esegui il comando seguente. Sostituisci *shell_file_name.sh* con il nome del tuo file di script di shell.

```
./shell_file_name.sh global-bundle.pem
```

Utilizzo Kerberos dell'autenticazione per RDS for Db2

Puoi utilizzare l'Kerberosautenticazione per autenticare gli utenti quando si connettono alla tua istanza database Amazon RDS for Db2. La tua istanza DB funziona con AWS Directory Service

for Microsoft Active Directory (AWS Managed Microsoft AD) per abilitare l'autenticazione. Kerberos Quando gli utenti si autenticano con un'istanza DB RDS for Db2 aggiunta al dominio trusting, le richieste di autenticazione vengono inoltrate alla directory con cui create. AWS Directory Service Per ulteriori informazioni, consulta Cos'è AWS Directory Service? nella Guida per l'amministrazione di AWS Directory Service.

Innanzitutto, crea una directory per archiviare le credenziali dell'utenteAWS Managed Microsoft AD. Quindi, aggiungi il dominio e altre informazioni della tua AWS Managed Microsoft AD directory all'istanza DB RDS for Db2. Quando gli utenti si autenticano con l'istanza DB RDS for Db2, le richieste di autenticazione vengono inoltrate alla directory. AWS Managed Microsoft AD

Mantenere tutte le credenziali nella stessa directory consente di ridurre il tempo e l'impegno. Con questo approccio, è disponibile una posizione centralizzata per archiviare e gestire le credenziali per più istanze database. L'uso di una directory può inoltre migliorare il profilo di sicurezza complessivo.

Argomenti

- Disponibilità di regioni e versioni
- Panoramica dell'Kerberosautenticazione per RDS per istanze DB Db2
- Configurazione dell'Kerberosautenticazione per RDS per istanze DB Db2
- Gestione di un'istanza database in un dominio
- Connessione a RDS for Db2 con autenticazione Kerberos

Disponibilità di regioni e versioni

Il supporto varia a seconda delle versioni specifiche di ciascun motore di database e a seconda delle Regioni AWS. Per ulteriori informazioni sulla versione e sulla disponibilità regionale di RDS for Db2 con autenticazione, vedere. Kerberos Regioni e motori DB supportati per l'autenticazione Kerberos in Amazon RDS

> **Note**
>
> Kerberosl'autenticazione non è supportata per le classi di istanze DB che sono obsolete per le istanze DB RDS per Db2. Per ulteriori informazioni, consulta RDS per classi di istanze Db2.

Panoramica dell'Kerberosautenticazione per RDS per istanze DB Db2

Per configurare Kerberos l'autenticazione per un'istanza DB RDS for Db2, completa i seguenti passaggi generali, descritti più dettagliatamente in seguito:

1. Utilizza AWS Managed Microsoft AD per creare una directory AWS Managed Microsoft AD. È possibile utilizzare ilAWS Management Console, the AWS Command Line Interface (AWS CLI) o AWS Directory Service per creare la directory. Per ulteriori informazioni, consulta Create your AWS Managed Microsoft AD directory nella AWS Directory ServiceAdministration Guide.
2. Creare un ruolo AWS Identity and Access Management (IAM) che utilizza la policy IAM gestita `AmazonRDSDirectoryServiceAccess`. Il ruolo IAM consente ad Amazon RDS di effettuare chiamate alla tua directory.

   Affinché il ruolo IAM consenta l'accesso, l'endpoint AWS Security Token Service (AWS STS) deve essere attivato nel modo corretto Regione AWS per te. Account AWS AWS STSGli endpoint sono tutti Regioni AWS attivi per impostazione predefinita e puoi utilizzarli senza ulteriori azioni. Per ulteriori informazioni, consulta Attivazione e disattivazione di AWS STS in una Regione AWS nella Guida per l'utente di IAM.
3. Crea o modifica un'istanza DB RDS for Db2 utilizzando l'AWS Management ConsoleAWS CLI, the o l'API RDS con uno dei seguenti metodi:
   - Crea una nuova istanza DB RDS per Db2 utilizzando la console, il create-db-instancecomando o l'operazione API CreateDBInstance. Per istruzioni, consulta Creazione di un'istanza database Amazon RDS.
   - Modifica un'istanza DB RDS for Db2 esistente utilizzando la console, il comando o l'modify-db-instanceoperazione API. ModifyDBInstance Per istruzioni, consulta Modifica di un'istanza database Amazon RDS.
   - Ripristina un'istanza DB RDS for Db2 da un'istantanea DB utilizzando la console, il restore-db-instance-from-db-snapshotcomando o l'operazione API. RestoreDBInstanceFromDBSnapshot Per istruzioni, consulta Ripristino da uno snapshot database.
   - Ripristina un'istanza DB RDS for Db2 point-in-time utilizzando la console, il restore-db-instance-to-point-in-timecomando o l'operazione API. RestoreDBInstanceToPointInTime Per istruzioni, consulta Ripristino a un'ora specifica per un'istanza database.

   Puoi localizzare l'istanza DB nello stesso Amazon Virtual Private Cloud (VPC) della directory o in un altro VPC. Account AWS Quando crei o modifichi l'istanza DB RDS for Db2, esegui le seguenti attività:

- Specifica l'identificativo del dominio (identificativo `d-*`) generato al momento della creazione della directory.
- Specifica anche il nome del ruolo IAM creato.
- Verifica che il gruppo di sicurezza dell'istanza DB possa ricevere traffico in entrata dal gruppo di sicurezza della directory.

4. Configura il tuo client Db2 e verifica che il traffico possa fluire tra l'host del client e AWS Directory Service le seguenti porte:
   - Porta TCP/UDP 53 — DNS
   - TCP 88 Kerberos — autenticazione
   - TCP 389 — LDAP
   - TCP 464 — autenticazione Kerberos

Configurazione dell'Kerberosautenticazione per RDS per istanze DB Db2

Si utilizza AWS Directory Service for Microsoft Active Directory (AWS Managed Microsoft AD) per configurare Kerberos l'autenticazione per un'istanza DB RDS for Db2. Per configurare Kerberos l'autenticazione, procedi nel seguente modo:

Argomenti

Fase 1: creazione di una directory utilizzando AWS Managed Microsoft AD

AWS Directory Servicecrea un file completamente gestito Active Directory inCloud AWS. Quando viene creata una directory AWS Managed Microsoft AD, AWS Directory Service crea automaticamente due controller di dominio e i server DNS. I server di directory vengono creati in sottoreti diverse in un VPC. Questa ridondanza assicura che la directory rimanga accessibile anche se si verifica un errore.

Quando crei una directory AWS Managed Microsoft AD, AWS Directory Service esegue le seguenti operazioni:

- Configura un file Active Directory all'interno del tuo VPC.
- Crea un account amministratore della directory con il nome utente `Admin` e la password specificata. Puoi utilizzare questo account per gestire le directory.

  > **Important**
  >
  > Assicurati di salvare la password. AWS Directory Service non memorizza questa password, che quindi non può essere recuperata o reimpostata.

- Crea un gruppo di sicurezza per i controller della directory. Il gruppo di sicurezza deve consentire la comunicazione con l'istanza DB RDS for Db2.

All'avvioAWS Directory Service for Microsoft Active Directory, AWS crea un'unità organizzativa (OU) che contiene tutti gli oggetti della directory. Questa unità organizzativa, che ha lo stesso nome NetBIOS che hai immesso al momento della creazione della directory, si trova nella radice del dominio. La radice del dominio è di proprietà di ed è gestita da AWS.

L'account `Admin` creato con la directory AWS Managed Microsoft AD dispone delle autorizzazioni per le attività amministrative più comuni per l'unità organizzativa:

- Crea, aggiorna o elimina utenti.
- Aggiungi risorse al tuo dominio, ad esempio server di file o di stampa, quindi assegna le autorizzazioni per tali risorse agli utenti dell'unità organizzativa.
- creazione di UO aggiuntive e container;
- delega dell'autorità;
- Ripristina gli oggetti eliminati dal Active Directory Cestino.
- Moduli Run Active Directory e Domain Name Service (DNS) perWindows PowerShell. AWS Directory Service

L'account `Admin` dispone anche dei diritti per eseguire queste attività in tutto il dominio:

- gestione delle configurazioni DNS (aggiunta, eliminazione o aggiornamento di record, zone e server d'inoltro);

- visualizzazione di log di eventi DNS;
- visualizzazione di log di eventi di sicurezza.

Per creare una directory con AWS Managed Microsoft AD

1. Accedi alla AWS Management Console e apri la console AWS Directory Service all'indirizzo https://console.aws.amazon.com/directoryservicev2/.
2. Scegli Configura directory.
3. Scegli AWS Managed Microsoft AD. AWS Managed Microsoft ADè l'unica opzione attualmente supportata per l'uso con Amazon RDS.
4. Seleziona Avanti.
5. Nella pagina Enter directory information (Inserisci le informazioni sulla directory) inserisci le seguenti informazioni:

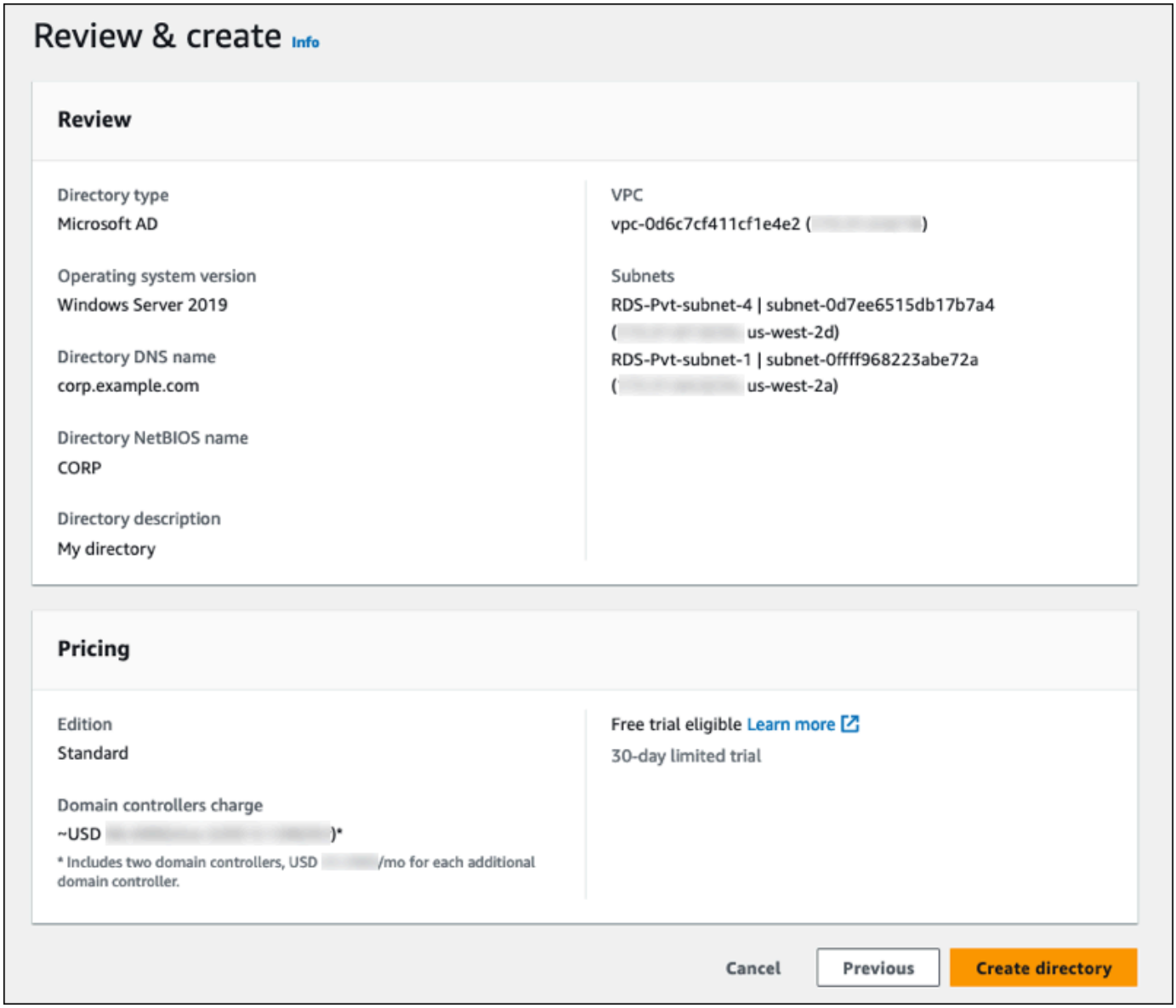
   - Edizione: scegli l'edizione che soddisfa i tuoi requisiti.
   - Nome DNS della directory: il nome completo per la directory, ad esempio`corp.example.com`.
   - Nome NetBIOS della directory: un nome breve opzionale per la directory, ad esempio. `CORP`
   - Descrizione della directory: una descrizione facoltativa per la directory.
   - Password di amministratore: la password per l'amministratore della directory. Il processo di creazione della directory crea un account amministratore con il nome utente `Admin` e questa password.

     La password dell'amministratore della directory non può includere il termine "admin". La password distingue tra maiuscole e minuscole e la lunghezza deve essere compresa tra 8 e 64 caratteri. Deve anche contenere un carattere di almeno tre delle seguenti quattro categorie:

     - Lettere minuscole (a–z)
     - Lettere maiuscole (A–Z)
     - Numeri (0–9)
     - Caratteri non alfanumerici (~!@#$%^&*_-+=`|\(){}[]:;"'<>,.?/)
     - Conferma password: digita nuovamente la password dell'amministratore.

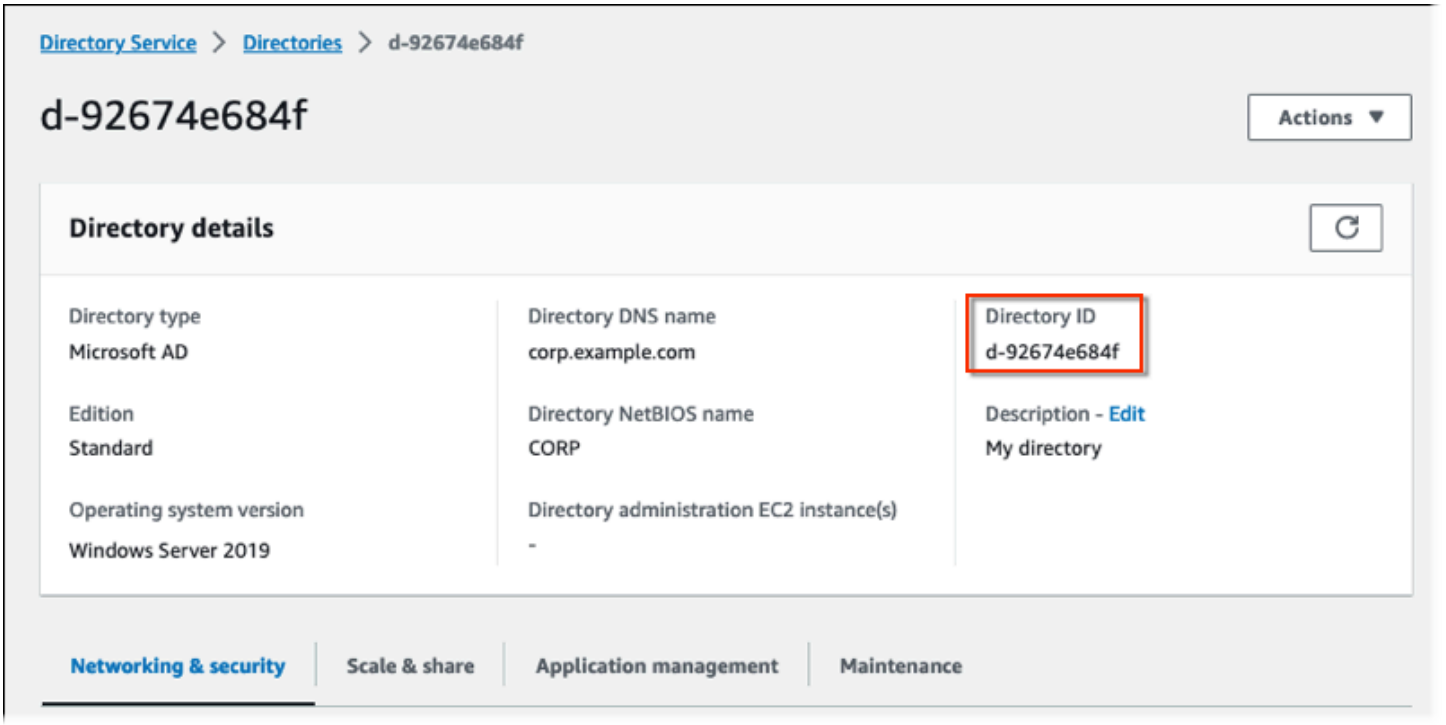
> **Important**
>
> Salvare la password. AWS Directory Service non memorizza questa password, che quindi non può essere recuperata o reimpostata.

6. Seleziona Avanti.
7. Nella pagina Choose VPC and subnets (Scegli VPC e sottoreti) fornire le seguenti informazioni:
   - VPC: scegli il VPC per la directory. È possibile creare l'istanza DB RDS for Db2 nello stesso VPC o in un altro VPC.
   - Sottoreti: scegli le sottoreti per i server di directory. Le due sottoreti devono trovarsi in diverse zone di disponibilità.
8. Seleziona Avanti.
9. Verificare le informazioni della directory Se sono necessarie modifiche, seleziona Previous (Precedente) e apporta le modifiche. Quando le informazioni sono corrette, scegli Create Directory (Crea directory).



Per creare la directory sono necessari alcuni minuti. Una volta creata correttamente la directory, il valore Status (Stato) viene modificato in Active (Attivo).

Per visualizzare le informazioni sulla tua directory, scegli l'ID della directory in Directory ID. Prendere nota del valore Directory ID (ID directory). È necessario questo valore quando si crea o si modifica l'istanza DB RDS for Db2.



Fase 2: creare un ruolo IAM a cui accedere ad Amazon RDS AWS Directory Service

Affinché Amazon RDS possa AWS Directory Service chiamarti, hai Account AWS bisogno di un ruolo IAM che utilizzi la policy `AmazonRDSDirectoryServiceAccess` IAM gestita. Questo ruolo consente ad Amazon RDS di effettuare chiamate versoAWS Directory Service.

Quando crei un'istanza database utilizzando la AWS Management Console e l'account utente della console dispone dell'autorizzazione `iam:CreateRole`, il ruolo IAM viene creato automaticamente. In questo caso, il nome del ruolo è `rds-directoryservice-kerberos-access-role`. In caso contrario, è necessario creare manualmente il ruolo IAM. Quando crei questo ruolo IAM, scegli `Directory Service` e collega ad esso la policy gestita AWS `AmazonRDSDirectoryServiceAccess`.

Per ulteriori informazioni sulla creazione di ruoli IAM per un servizio, consulta Creazione di un ruolo per delegare le autorizzazioni a un servizio AWS nella Guida per l'utente di IAM.

> **Note**
>
> Il ruolo IAM utilizzato per Windows Authentication for RDS non Microsoft SQL Server può essere utilizzato per RDS for Db2.

Facoltativamente, puoi creare policy con le autorizzazioni richieste anziché utilizzare la policy `AmazonRDSDirectoryServiceAccess` gestita. In questo caso, il ruolo IAM deve avere la seguente policy di fiducia IAM:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "directoryservice.rds.amazonaws.com",
          "rds.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Il ruolo deve inoltre avere la seguente politica di ruolo IAM:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "ds:DescribeDirectories",
        "ds:AuthorizeApplication",
        "ds:UnauthorizeApplication",
        "ds:GetAuthorizedApplicationDetails"
      ],
    "Effect": "Allow",
    "Resource": "*"
    }
  ]
}
```

Fase 3: creazione e configurazione di utenti

È possibile creare utenti utilizzando lo Active Directory Users and Computers strumento. Questo è uno dei Active Directory Domain Services seguenti Active Directory Lightweight Directory Services strumenti. Per ulteriori informazioni, consulta Aggiungere utenti e computer al Active Directory dominio nella Microsoft documentazione. In questo caso, gli utenti sono individui o altre entità, come i rispettivi computer, che fanno parte del dominio e le cui identità vengono mantenute nella directory.

Per creare utenti in una AWS Directory Service directory, devi essere connesso a un'istanza Amazon EC2 Windows basata che fa parte della AWS Directory Service directory. Allo stesso tempo, devi accedere come utente con i privilegi per creare utenti. Per ulteriori informazioni, consulta Creazione di un utente nella Guida di amministrazione di AWS Directory Service.

Passaggio 4: Creare un gruppo di amministratori RDS for Db2 in AWS Managed Microsoft AD

RDS per Db2 non supporta Kerberos l'autenticazione per l'utente master o per i due utenti riservati di Amazon RDS e. `rdsdb` `rdsadmin` È invece necessario creare un nuovo gruppo chiamato in. `masterdba` AWS Managed Microsoft AD Per ulteriori informazioni, consulta Creare un account di gruppo Active Directory nella Microsoft documentazione. Tutti gli utenti aggiunti a questo gruppo avranno i privilegi di utente principale.

Dopo aver abilitato Kerberos l'autenticazione, l'utente principale perde il `masterdba` ruolo. Di conseguenza, l'utente master non sarà in grado di accedere all'appartenenza al gruppo di utenti locale dell'istanza a meno che non si disabiliti Kerberos l'autenticazione. Per continuare a utilizzare l'utente master con accesso tramite password, crea un utente AWS Managed Microsoft AD con lo stesso nome dell'utente principale. Quindi, aggiungi quell'utente al gruppo`masterdba`.

Passaggio 5: creare o modificare un'istanza DB RDS for Db2

Crea o modifica un'istanza DB RDS for Db2 da utilizzare con la tua directory. È possibile utilizzare l'API AWS Management ConsoleAWS CLI, the o RDS per associare un'istanza DB a una directory. Questa operazione può essere eseguita in uno dei seguenti modi:

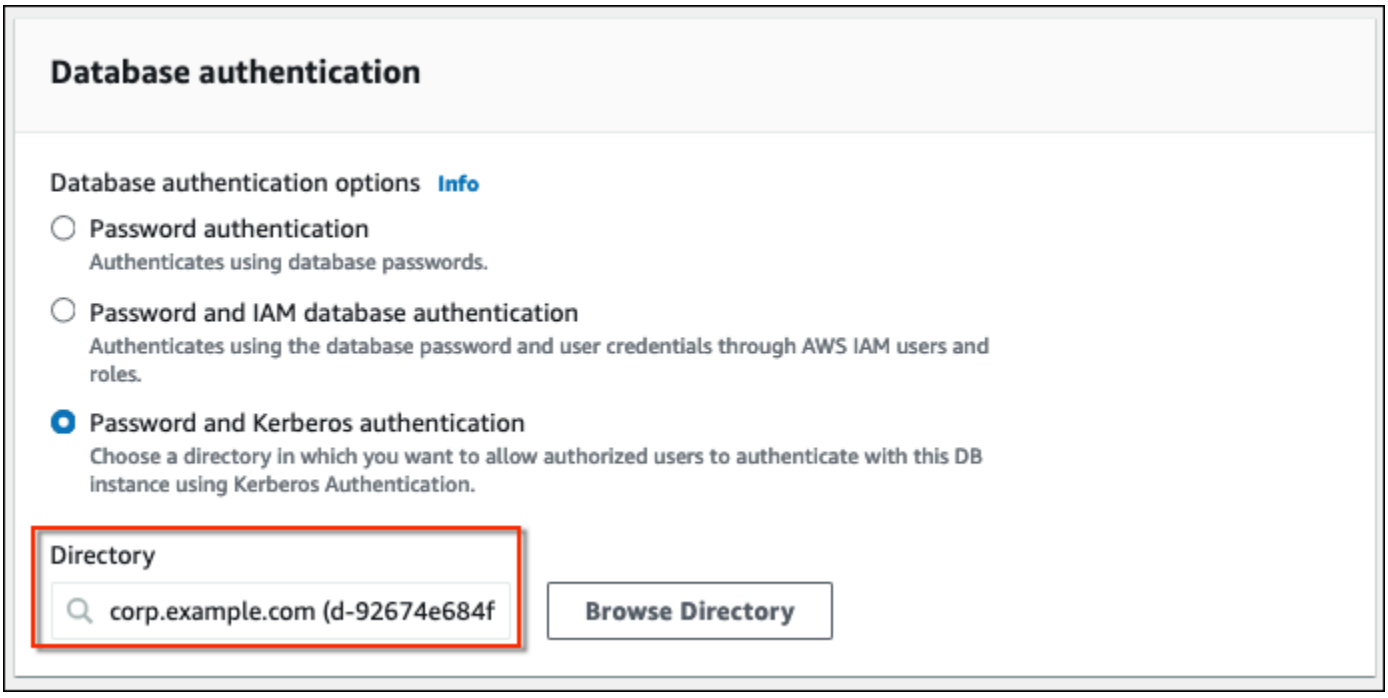
- Crea una nuova istanza DB RDS for Db2 utilizzando la console, il create-db-instancecomando o l'CreateDBInstanceoperazione API. Per istruzioni, consulta Creazione di un'istanza database Amazon RDS.
- Modifica un'istanza DB RDS for Db2 esistente utilizzando la console, il modify-db-instancecomando o l'operazione API ModifyDbInstance. Per istruzioni, consulta Modifica di un'istanza database Amazon RDS.

- Ripristina un'istanza DB RDS for Db2 da un'istantanea DB utilizzando la console, il comando o l'operazione API. restore-db-instance-from-db-snapshotRestoreDBInstanceFromDBSnapshot Per istruzioni, consulta Ripristino da uno snapshot database.
- Ripristina un'istanza DB RDS for Db2 point-in-time utilizzando la console, il restore-db-instance-to-point-in-timecomando o l'operazione API. RestoreDBInstanceToPointInTime Per istruzioni, consulta Ripristino a un'ora specifica per un'istanza database.

Kerberosl'autenticazione è supportata solo per le istanze DB RDS per Db2 in un VPC. L'istanza database Oracle può trovarsi nello stesso VPC della directory o in un VPC diverso. L'istanza DB deve utilizzare un gruppo di sicurezza che consenta l'ingresso e l'uscita all'interno del VPC della directory in modo che l'istanza DB possa comunicare con la directory.

Console

Quando usi la console per creare, modificare o ripristinare un'istanza DB, scegli Password e Kerberos autenticazione nella sezione Autenticazione del database. Quindi scegli Sfoglia directory. Seleziona la directory o scegli Crea directory per utilizzare il Directory Service.



AWS CLI

Quando utilizzi AWS CLI, sono necessari i seguenti parametri per consentire all'istanza database di utilizzare la directory che hai creato:

- Per il `--domain` parametro, utilizzate l'identificatore di dominio (» `d-*` "identificatore) generato al momento della creazione della directory.
- Per il parametro `--domain-iam-role-name`, utilizza il ruolo creato che utilizza la policy IAM gestita `AmazonRDSDirectoryServiceAccess`.

L'esempio seguente modifica un'istanza DB per utilizzare una directory. Sostituite i seguenti segnaposto dell'esempio con i vostri valori:

- *`db_instance_name — Il nome`* dell'istanza DB RDS for Db2.
- *`directory_id`* — L'ID della directory che hai creato. AWS Directory Service for Microsoft Active Directory
- *`role_name — Il nome del`* ruolo IAM che hai creato.

```
aws rds modify-db-instance --db-instance-identifier db_instance_name --domain
 d-directory_id --domain-iam-role-name role_name
```

> **Important**
>
> Se modifichi un'istanza DB per abilitare Kerberos l'autenticazione, riavvia l'istanza DB dopo aver apportato la modifica.

Fase 6: Configurazione di un client Db2

Per configurare un client Db2

1. Crea un file /etc/krb5.conf (o equivalente) per puntare al dominio.

   > **Note**
   >
   > Per i sistemi operativi Windows, crea un file C:\windows\krb5.ini.

2. Verifica che il traffico scorra senza problemi tra l'host client e AWS Directory Service. Utilizzate un'utilità di rete, ad esempio Netcat per le seguenti attività:

   a. Verifica il traffico su DNS per la porta 53.

   b. Verifica il traffico su TCP/UDP per la porta 53 e perKerberos, che include le porte 88 e 464 per. AWS Directory Service

3. Verifica che il traffico scorra senza problemi tra l'host client e l'istanza database sulla porta del database. È possibile utilizzare il comando `db2` per connettersi e accedere al database.

L'esempio seguente è il contenuto del file /etc/krb5.conf per: AWS Managed Microsoft AD

```
[libdefaults]
default_realm = EXAMPLE.COM
[realms]
EXAMPLE.COM = {
kdc = example.com
admin_server = example.com
}
[domain_realm]
.example.com = EXAMPLE.COM
example.com = EXAMPLE.COM
```

Gestione di un'istanza database in un dominio

Puoi utilizzare l'APIAWS Management Console, the o RDS per gestire la AWS CLI tua istanza DB e la sua relazione con la tua. Microsoft Active Directory Ad esempio, puoi associare un file Active Directory per abilitare Kerberos l'autenticazione. È inoltre possibile rimuovere l'associazione per Active Directory disabilitare Kerberos l'autenticazione. È inoltre possibile spostare un'istanza DB in modo che venga autenticata esternamente da una all'altraMicrosoft Active Directory.

Ad esempio, utilizzando il comando modify-db-instanceCLI, è possibile eseguire le seguenti azioni:

- Riprova ad abilitare Kerberos l'autenticazione per un'iscrizione non riuscita specificando l'ID di directory dell'appartenenza corrente per l'opzione. `--domain`
- Disabilita Kerberos l'autenticazione su un'istanza DB specificando `none` l'opzione. `--domain`
- Sposta un'istanza DB da un dominio all'altro specificando l'identificatore di dominio del nuovo dominio per l'opzione. `--domain`

Appartenenza al dominio

Quando l'istanza database viene creata o modificata diventa membro del dominio. È possibile visualizzare lo stato dell'appartenenza al dominio nella console o eseguendo il describe-db-instancescomando. Lo stato dell'istanza di database può essere uno dei seguenti:

- `kerberos-enabled`— L'istanza DB ha Kerberos l'autenticazione abilitata.
- `enabling-kerberos`— AWS sta abilitando Kerberos l'autenticazione su questa istanza DB.
- `pending-enable-kerberos`— Kerberos L'abilitazione dell'autenticazione è in sospeso su questa istanza DB.
- `pending-maintenance-enable-kerberos`— AWS tenterà di abilitare Kerberos l'autenticazione sull'istanza DB durante la successiva finestra di manutenzione programmata.
- `pending-disable-kerberos`— La disabilitazione Kerberos dell'autenticazione è in sospeso su questa istanza DB.
- `pending-maintenance-disable-kerberos`— AWS tenterà di disabilitare Kerberos l'autenticazione sull'istanza DB durante la successiva finestra di manutenzione programmata.
- `enable-kerberos-failed`— Un problema di configurazione ha AWS impedito Kerberos l'attivazione dell'autenticazione sull'istanza DB. Correggere il problema di configurazione prima di emettere nuovamente il comando per modificare l'istanza DB.
- `disabling-kerberos`— AWS sta disabilitando l'Kerberosautenticazione su questa istanza DB.

Una richiesta di abilitazione dell'Kerberosautenticazione può fallire a causa di un problema di connettività di rete o di un ruolo IAM errato. In alcuni casi, il tentativo di abilitare Kerberos l'autenticazione potrebbe fallire quando si crea o si modifica un'istanza DB. In tal caso, verifica di utilizzare il ruolo IAM corretto, quindi modifica l'istanza DB per aggiungerla al dominio.

Connessione a RDS for Db2 con autenticazione Kerberos

Per connettersi a RDS for Db2 con autenticazione Kerberos

1. Al prompt dei comandi, esegui il comando seguente . Nell'esempio seguente, sostituisci il *nome utente con il tuo Microsoft Active Directory nome utente*.

```
kinit username
```

2. Se l'istanza DB RDS for Db2 utilizza un VPC accessibile pubblicamente, aggiungi l'indirizzo IP dell'endpoint dell'istanza DB al tuo file `/etc/hosts` sul client Amazon EC2. L'esempio seguente ottiene l'indirizzo IP e lo aggiunge al file. `/etc/hosts`

```
% dig +short Db2-endpoint.AWS-Region.rds.amazonaws.com
;; Truncated, retrying in TCP mode.
ec2-34-210-197-118.AWS-Region.compute.amazonaws.com.
34.210.197.118

% echo "34.210.197.118  Db2-endpoint.AWS-Region.rds.amazonaws.com" >> /etc/hosts
```

3. Utilizzare il comando seguente per accedere a un'istanza DB RDS for Db2 associata a. Active Directory Sostituisci *`database_name con il nome`* del tuo database RDS for Db2.

```
db2 connect to database_name
```

Amministrazione dell'istanza DB RDS for Db2

Questo argomento descrive le attività di gestione comuni eseguite con un'istanza DB RDS for Db2. Alcune attività sono le stesse per tutte le istanze database di Amazon RDS. Altre attività sono specifiche di RDS for Db2.

Le seguenti attività sono comuni a tutti i database RDS. Esistono anche attività specifiche di RDS for Db2, come la connessione a un database RDS for Db2 con un client SQL standard.

Area attività	Documentazione di riferimento
Classi delle istanze, storage e PIOPS Se si sta creando un'istanza di produzione, occorre conoscere il funzionamento di classi di istanza, tipi di storage e IOPS con provisioning in Amazon RDS.	Classi di istanze database Tipi di storage Amazon RDS
Implementazioni Multi-AZ Un'istanza database in produzione deve utilizzare implement azioni Multi-AZ. Le implementazioni Multi-AZ forniscono alle istanze database maggior disponibilità, longevità dei dati e tolleranza ai guasti.	Configurazione e gestione di un'implementazione multi-AZ
Amazon VPC Se Account AWS disponi di un cloud privato virtuale (VPC) predefinito, l'istanza DB viene creata automaticamente all'inter no del VPC predefinito. Se l'account non dispone di un VPC predefinito e desideri che l'istanza database sia in un VPC, è necessario creare il VPC e i gruppi di sottoreti prima di creare l'istanza database.	Uso di un'istanza database in un VPC
Gruppi di sicurezza Per impostazione predefinita, le istanze database utilizzano un firewall che impedisce l'accesso. Per accedere all'istanza database, assicurati di aver creato un gruppo di sicurezza con gli indirizzi IP e la configurazione di rete corretti.	Controllo dell'accesso con i gruppi di sicurezza

Area attività	Documentazione di riferimento
Gruppi di parametri Poiché la tua istanza DB RDS for Db2 richiede l'aggiunta dei `rds.ibm_site_id` parametri `rds.ibm_customer_id` and, crea un gruppo di parametri prima di creare l'istanza DB. Se l'istanza DB richiede altri parametri di database specifici, aggiungili anche a questo gruppo di parametri prima di creare l'istanza DB.	Aggiungere IBM ID a un gruppo di parametri per RDS per istanze DB Db2 Utilizzo di gruppi di parametri
Connessione all'istanza database Dopo aver creato un gruppo di sicurezza e averlo associato a un'istanza DB, puoi connetterti all'istanza DB con qualsiasi applicazione client SQL standard comeIBM Db2 CLP.	Connessione all'istanza DB RDS for Db2
Backup e ripristino È possibile configurare l'istanza DB per eseguire backup di storage automatici o istantanee di storage manuali e quindi ripristinare le istanze dai backup o dalle istantanee.	Backup, ripristino ed esportazione dei dati
Monitoraggio È possibile monitorare un'istanza DB RDS for Db2 con. IBM Db2 Data Management Console Puoi anche monitorare un'istanza DB RDS for Db2 utilizzando i parametri, gli eventi e il monitoraggio avanzato di CloudWatch Amazon RDS.	Connessione all'istanza DB RDS for Db2 con IBM Db2 Data Management Console Visualizzazione dei parametri nella console Amazon RDS Visualizzazione di eventi Amazon RDS Monitoraggio dei parametri del sistema operativo con il monitoraggio avanzato

Area attività	Documentazione di riferimento
File di log Puoi accedere ai file di registro per la tua istanza DB RDS for Db2.	Monitoraggio dei file di log di Amazon RDS

Argomenti

- Esecuzione di attività di sistema comuni per istanze DB RDS for Db2
- Esecuzione di attività di database comuni per istanze database Amazon RDS for Db2

Esecuzione di attività di sistema comuni per istanze DB RDS for Db2

Puoi eseguire alcune attività comuni di amministratore del database relative al sistema sulle istanze DB di Amazon RDS che eseguono Db2. Per fornire un'esperienza di servizio gestito, Amazon RDS non fornisce accesso shell alle istanze database e limita l'accesso a certe procedure e tabelle di sistema che richiedono privilegi avanzati.

Argomenti

- Creazione di un endpoint di database personalizzato
- Concessione e revoca dei privilegi
- Collegamento all'istanza remota di RDS for Db2 DB

Creazione di un endpoint di database personalizzato

Quando si esegue la migrazione a RDS for Db2, è possibile utilizzare URL di endpoint di database personalizzati per ridurre al minimo le modifiche all'applicazione. Ad esempio, se lo utilizzi `db2.example.com` come record DNS corrente, puoi aggiungerlo ad Amazon Route 53. In Route 53, puoi utilizzare zone ospitate private per mappare l'endpoint del database DNS corrente su un endpoint del database RDS for Db2. Per aggiungere un `CNAME` record `A` o un record personalizzato per un endpoint del database Amazon RDS, consulta Registrazione e gestione di domini utilizzando Amazon Route 53 nella Amazon Route 53 Developer Guide.

> **Note**
>
> Se non riesci a trasferire il dominio su Route 53, puoi utilizzare il tuo provider DNS per creare un `CNAME` record per l'URL dell'endpoint del database RDS for Db2. Consulta la documentazione del tuo provider DNS.

Concessione e revoca dei privilegi

Gli utenti accedono ai database tramite l'appartenenza a gruppi collegati ai database. Se rimuovi tutti i gruppi collegati a un database da un utente, l'utente non potrà connettersi al database.

Utilizza le seguenti procedure per concedere e revocare i privilegi per controllare l'accesso al database.

Queste procedure utilizzano l'IBM Db2 CLPesecuzione su un computer locale per connettersi a un'istanza DB RDS for Db2. Assicurati di catalogare il nodo TCPIP e il database per connetterti all'istanza DB RDS for Db2 in esecuzione sul tuo computer locale. Per ulteriori informazioni, consulta Connessione all'istanza DB RDS for Db2 con IBM Db2 CLP.

Argomenti

- Concedere a un utente l'accesso al database
- Modifica della password di un utente
- Aggiungere gruppi a un utente
- Rimuovere gruppi da un utente
- Rimuovere un utente
- Elencare gli utenti
- Creazione di un ruolo
- Concessione di un ruolo
- Revoca di un ruolo
- Concessione dell'autorizzazione al database
- Revoca dell'autorizzazione del database

Concedere a un utente l'accesso al database

Per concedere a un utente l'accesso al tuo database

1. Connect al `rdsadmin` database utilizzando il nome utente principale e la password principale per l'istanza DB RDS for Db2. Nell'esempio seguente, sostituite *`master_username` e `master_password`* con le vostre informazioni.

   ```
   db2 connect to rdsadmin user master_username using master_password
   ```

   Questo comando produce un output simile all'esempio seguente:

   ```
   Database Connection Information

   Database server        = DB2/LINUXX8664 11.5.8.0
   SQL authorization ID   = ADMIN
   Local database alias   = RDSADMIN
   ```

2. Aggiungi un utente alla tua lista di autorizzazioni chiamando`rdsadmin.add_user`. Per ulteriori informazioni, consulta rdsadmin.add_user.

   ```
   db2 "call rdsadmin.add_user(
       'username',
       'password',
       'group_name,group_name')"
   ```

3. (Facoltativo) Aggiungi altri gruppi all'utente chiamando`rdsadmin.add_groups`. Per ulteriori informazioni, consulta rdsadmin.add_groups.

   ```
   db2 "call rdsadmin.add_groups(
       'username',
       'group_name,group_name')"
   ```

4. Conferma le autorità disponibili per l'utente. *`Nell'esempio seguente, sostituisci rds_database_alias, master_user e master_password con le tue informazioni.`* Inoltre, sostituisci il nome utente con il nome utente dell'utente.

   ```
   db2 terminate
   db2 connect to rds_database_alias user master_user using master_password
   db2 "SELECT SUBSTR(AUTHORITY,1,20) AUTHORITY, D_USER, D_GROUP, D_PUBLIC
   ```

```
      FROM TABLE (SYSPROC.AUTH_LIST_AUTHORITIES_FOR_AUTHID ('username', 'U') ) AS
T
      ORDER BY AUTHORITY"
```

Questo comando produce un output simile all'esempio seguente:

```
AUTHORITY            D_USER D_GROUP D_PUBLIC
-------------------- ------ ------- --------
ACCESSCTRL           N      N       N
BINDADD              N      N       N
CONNECT              N      N       N
CREATETAB            N      N       N
CREATE_EXTERNAL_ROUT N      N       N
CREATE_NOT_FENCED_RO N      N       N
CREATE_SECURE_OBJECT N      N       N
DATAACCESS           N      N       N
DBADM                N      N       N
EXPLAIN              N      N       N
IMPLICIT_SCHEMA      N      N       N
LOAD                 N      N       N
QUIESCE_CONNECT      N      N       N
SECADM               N      N       N
SQLADM               N      N       N
SYSADM               *      N       *
SYSCTRL              *      N       *
SYSMAINT             *      N       *
SYSMON               *      N       *
WLMADM               N      N       N
```

5. Concedi l'RDS per i ruoli `ROLE_NULLID_PACKAGES` Db2 e `ROLE_PROCEDURES` al gruppo a cui hai aggiunto l'utente. `ROLE_TABLESPACES`

> **Note**
>
> Creiamo RDS per istanze DB Db2 in modalità. `RESTRICTIVE` Pertanto, RDS for Db2 svolge i ruoli e `ROLE_PROCEDURES` concede `ROLE_NULLID_PACKAGES` i `ROLE_TABLESPACES` privilegi di esecuzione sui pacchetti per e. `NULLID` IBM Db2 CLP Dynamic SQL Questi ruoli concedono anche i privilegi utente sui tablespace.

a. Connect al database Db2. *Nell'esempio seguente, sostituisci database_name, master_user e master_password con le tue informazioni.*

```
db2 connect to database_name user master_user using master_password
```

b. Assegna il ruolo a un gruppo. ROLE_NULLED_PACKAGES Nell'esempio seguente, sostituisci *group_name* con il nome del gruppo a cui desideri aggiungere il ruolo.

```
db2 "grant role ROLE_NULLID_PACKAGES to group group_name"
```

c. Assegnate il ruolo ROLE_TABLESPACES allo stesso gruppo. Nell'esempio seguente, sostituisci *group_name* con il nome del gruppo a cui desideri aggiungere il ruolo.

```
db2 "grant role ROLE_TABLESPACES to group group_name"
```

d. Assegnate il ruolo ROLE_PROCEDURES allo stesso gruppo. Nell'esempio seguente, sostituisci *group_name* con il nome del gruppo a cui desideri aggiungere il ruolo.

```
db2 "grant role ROLE_PROCEDURES to group group_name"
```

6. Concedi connectbindadd,createtab, e IMPLICIT_SCHEMA autorità al gruppo a cui hai aggiunto l'utente. Nell'esempio seguente, sostituisci *group_name* con il nome del secondo gruppo a cui hai aggiunto l'utente.

```
db2 "grant usage on workload SYSDEFAULTUSERWORKLOAD to public"
db2 "grant connect, bindadd, createtab, implicit_schema on database to
 group group_name"
```

7. Ripeti i passaggi da 4 a 6 per ogni gruppo aggiuntivo a cui hai aggiunto l'utente.

8. Verifica l'accesso dell'utente connettendoti come utente, creando una tabella, inserendo valori nella tabella e restituendo dati dalla tabella. Nell'esempio seguente, sostituisci *rds_database_alias*, *nome utente* e *password con il nome del database e il nome utente e la password* dell'utente.

```
db2 connect to rds_database_alias user username using password
db2 "create table t1(c1 int not null)"
db2 "insert into t1 values (1),(2),(3),(4)"
```

```
db2 "select * from t1"
```

Modifica della password di un utente

Per modificare la password di un utente

1. Connect al `rdsadmin` database utilizzando il nome utente principale e la password principale per l'istanza DB RDS for Db2. Nell'esempio seguente, sostituite *`master_username`* e *`master_password`* con le vostre informazioni.

   ```
   db2 connect to rdsadmin user master_username using master_password
   ```

2. Modificate la password chiamando. `rdsadmin.change_password` Per ulteriori informazioni, consulta rdsadmin.change_password.

   ```
   db2 "call rdsadmin.change_password(
       'username',
       'new_password')"
   ```

Aggiungere gruppi a un utente

Per aggiungere gruppi a un utente

1. Connect al `rdsadmin` database utilizzando il nome utente principale e la password principale per l'istanza DB RDS for Db2. Nell'esempio seguente, sostituite *`master_username`* e *`master_password`* con le vostre informazioni.

   ```
   db2 connect to rdsadmin user master_username using master_password
   ```

2. Aggiungi gruppi a un utente chiamando. `rdsadmin.add_groups` Per ulteriori informazioni, consulta rdsadmin.add_groups.

   ```
   db2 "call rdsadmin.add_groups(
       'username',
       'group_name,group_name')"
   ```

Rimuovere gruppi da un utente

Per rimuovere gruppi da un utente

1. Connect al `rdsadmin` database utilizzando il nome utente principale e la password principale per l'istanza DB RDS for Db2. Nell'esempio seguente, sostituite *`master_username`* e *`master_password`* con le vostre informazioni.

```
db2 connect to rdsadmin user master_username using master_password
```

2. Rimuovi i gruppi chiamando. `rdsadmin.remove_groups` Per ulteriori informazioni, consulta rdsadmin.remove_groups.

> **Warning**
>
> Se rimuovi tutti i gruppi collegati a un database da un utente, l'utente non potrà connettersi al database. Questo perché Amazon RDS concede l'autorità al gruppo, non all'utente.

```
db2 "call rdsadmin.remove_groups(
    'username',
    'group_name,group_name')"
```

Rimuovere un utente

Per rimuovere un utente dall'elenco di autorizzazioni

1. Connect al `rdsadmin` database utilizzando il nome utente principale e la password principale per l'istanza DB RDS for Db2. Nell'esempio seguente, sostituite *`master_username`* e *`master_password`* con le vostre informazioni.

```
db2 connect to rdsadmin user master_username using master_password
```

2. Rimuovi un utente dall'elenco di autorizzazioni chiamando. `rdsadmin.remove_user` Per ulteriori informazioni, consulta rdsadmin.remove_user.

```
db2 "call rdsadmin.remove_user('username')"
```

Elencare gli utenti

Per elencare gli utenti in un elenco di autorizzazioni, richiama la `rdsadmin.list_users` stored procedure. Per ulteriori informazioni, consulta rdsadmin.list_users.

```
db2 "call rdsadmin.list_users()"
```

Creazione di un ruolo

È possibile utilizzare la rdsadmin.create_role stored procedure per creare un ruolo.

Per creare un ruolo

1. Connect al `rdsadmin` database. Nell'esempio seguente, sostituisci *`master_username`* e *`master_password`* con le tue informazioni.

   ```
   db2 connect to rdsadmin user master_username using master_password
   ```

2. Imposta Db2 per l'output del contenuto.

   ```
   db2 set serveroutput on
   ```

3. Creare un ruolo. Per ulteriori informazioni, consulta the section called “rdsadmin.create_role”.

   ```
   db2 "call rdsadmin.create_role(
       'database_name',
       'role_name')"
   ```

4. Imposta Db2 in modo che non emetta contenuti.

   ```
   db2 set serveroutput off
   ```

Concessione di un ruolo

È possibile utilizzare la rdsadmin.grant_role stored procedure per assegnare un ruolo a un ruolo, utente o gruppo.

Per assegnare un ruolo

1. Connect al `rdsadmin` database. Nell'esempio seguente, sostituisci *`master_username`* e *`master_password`* con le tue informazioni.

```
db2 connect to rdsadmin user master_username using master_password
```

2. Imposta Db2 per l'output del contenuto.

```
db2 set serveroutput on
```

3. Assegna un ruolo. Per ulteriori informazioni, consulta the section called “rdsadmin.grant_role”.

```
db2 "call rdsadmin.grant_role(
    'database_name',
    'role_name',
    'grantee',
    'admin_option')"
```

4. Imposta Db2 in modo che non emetta contenuti.

```
db2 set serveroutput off
```

Revoca di un ruolo

È possibile utilizzare la rdsadmin.revoke_role stored procedure per revocare un ruolo da un ruolo, un utente o un gruppo.

Per revocare un ruolo

1. Connect al `rdsadmin` database. Nell'esempio seguente, sostituisci *`master_username`* e *`master_password`* con le tue informazioni.

```
db2 connect to rdsadmin user master_username using master_password
```

2. Revoca un ruolo. Per ulteriori informazioni, consulta the section called “rdsadmin.revoke_role”.

```
db2 "call rdsadmin.revoke_role(
    ?,
    'database_name',
    'role_name',
    'grantee')"
```

Concessione dell'autorizzazione al database

L'utente principale, che dispone dell'DBADMautorizzazione, può concedere DBADM o DATAACCESS autorizzare un ruolo, un utente o un gruppo. ACCESSCTRL

Per concedere l'autorizzazione al database

1. Connect al rdsadmin database utilizzando il nome utente principale e la password principale per l'istanza DB RDS for Db2. Nell'esempio seguente, sostituite *master_username e master_password* con le vostre informazioni.

   ```
   db2 connect to rdsadmin user master_username using master_password
   ```

2. Concedi l'accesso a un utente chiamando. rdsadmin.dbadm_grant Per ulteriori informazioni, consulta rdsadmin.dbadm_grant.

   ```
   db2 "call rdsadmin.dbadm_grant(
       ?,
       'database_name',
       'authorization',
       'grantee')"
   ```

Esempio di caso d'uso

La procedura seguente illustra come creare un ruolo, concedere DBADM l'autorizzazione al ruolo e assegnare il ruolo a un utente.

Per creare un ruolo, concedi **DBADM** l'autorizzazione e assegna il ruolo a un utente

1. Connect al rdsadmin database utilizzando il nome utente principale e la password principale per l'istanza DB RDS for Db2. Nell'esempio seguente, sostituite *master_username e master_password* con le vostre informazioni.

   ```
   db2 connect to rdsadmin user master_username using master_password
   ```

2. Create un ruolo chiamato PROD_ROLE per un database chiamato. TESTDB Per ulteriori informazioni, consulta rdsadmin.create_role.

   ```
   db2 "call rdsadmin.create_role(
       'TESTDB',
   ```

```
    'PROD_ROLE')"
```

3. Assegna il ruolo a un utente chiamatoPROD_USER. PROD_USERViene concessa l'autorizzazione di amministratore per assegnare ruoli. Per ulteriori informazioni, consulta rdsadmin.grant_role.

```
db2 "call rdsadmin.grant_role(
    ?,
    'TESTDB',
    'PROD_ROLE',
    'USER PROD_USER',
    'Y')"
```

4. (Facoltativo) Fornisci autorizzazioni o privilegi aggiuntivi. L'esempio seguente concede DBADM l'autorizzazione a un ruolo denominato in base PROD_ROLE a un database chiamato. FUNDPROD Per ulteriori informazioni, consulta rdsadmin.dbadm_grant.

```
db2 "call rdsadmin.dbadm_grant(
    ?,
    'FUNDPROD',
    'DBADM',
    'ROLE PROD_ROLE')"
```

5. Termina la sessione.

```
db2 terminate
```

6. Connect al testdb database utilizzando il nome utente principale e la password principale per l'istanza DB RDS for Db2. Nell'esempio seguente, sostituite *master_username e master_password* con le vostre informazioni.

```
db2 connect to testdb user master_username using master_password
```

7. Aggiungi altre autorizzazioni al ruolo.

```
db2 "grant connect, implicit_schema on database to role PROD_ROLE"
```

Revoca dell'autorizzazione del database

L'utente principale, che dispone dell'DBADMautorizzazione, può revocare o revocare DBADM DATAACCESS l'autorizzazione a un ruolo, utente o gruppo. ACCESSCTRL

Per revocare l'autorizzazione del database

1. Connect al `rdsadmin` database utilizzando il nome utente principale e la password principale per l'istanza DB RDS for Db2. Nell'esempio seguente, sostituite *`master_username e master_password`* con le vostre informazioni.

   ```
   db2 connect to rdsadmin user master_username using master_password
   ```

2. Revoca l'accesso utente chiamando. `rdsadmin.dbadm_revoke` Per ulteriori informazioni, consulta rdsadmin.dbadm_revoke.

   ```
   db2 "call rdsadmin.dbadm_revoke(
       ?,
       'database_name,
       'authorization',
       'grantee')"
   ```

Collegamento all'istanza remota di RDS for Db2 DB

Da collegare all'istanza DB remota di RDS for Db2

1. Esegui una sessione lato clientIBM Db2 CLP. Per informazioni sulla catalogazione dell'istanza e del database DB RDS for Db2, consulta. Connessione all'istanza DB RDS for Db2 con IBM Db2 CLP Prendi nota del nome utente e della password principale per l'istanza DB RDS for Db2.
2. Collegalo all'istanza DB RDS for Db2. Nell'esempio seguente, sostituite *`node_name, master_username e master_password con il nome del nodo TCPIP che avete catalogato e il nome utente e la password`* principali per l'istanza DB RDS for Db2.

   ```
   db2 attach to node_name user master_username using master_password
   ```

Dopo il collegamento all'istanza RDS for Db2 DB remota, è possibile eseguire i seguenti comandi e altri comandi. `get snapshot` Per ulteriori informazioni, vedete il GET SNAPSHOTcomando nella documentazione. IBM Db2

```
db2 list applications
db2 get snapshot for all databases
db2 get snapshot for database manager
```

```
db2 get snapshot for all applications
```

Esecuzione di attività di database comuni per istanze database Amazon RDS for Db2

Puoi eseguire alcune attività DBA comuni relative ai database sulle tue istanze DB RDS per Db2. Per offrire un'esperienza di servizio gestito, Amazon RDS non fornisce accesso shell alle istanze database. Inoltre, l'utente principale non può eseguire comandi o utilità che richiedono `SYSADM` o autorizzano. `SYSMAINT` `SYSCTRL`

Argomenti

- Gestione dei buffer pool
- Gestione dello storage
- Gestione dei tablespace
- Generazione di report sulle prestazioni
- Raccolta di informazioni sui database
- Forzare le applicazioni a uscire dai database

Gestione dei buffer pool

È possibile creare, modificare o eliminare i pool di buffer per un database RDS for Db2. La creazione, la modifica o l'eliminazione dei pool di buffer richiede un'`SYSADMIN`autorità di livello superiore, che non è disponibile per l'utente principale. Utilizza invece le stored procedure di Amazon RDS.

Puoi anche svuotare i buffer pool.

Argomenti

- Creazione di un buffer pool
- Modifica di un pool di buffer
- Eliminazione di un pool di buffer
- Svuotare i buffer pool

Creazione di un buffer pool

Per creare un pool di buffer per il database RDS for Db2, chiamate la stored procedure. `rdsadmin.create_bufferpool` Per ulteriori informazioni, consulta la CREATE BUFFERPOOLdichiarazione nella documentazione. IBM Db2

Per creare un pool di buffer

1. Connect al `rdsadmin` database utilizzando il nome utente principale e la password principale per l'istanza DB RDS for Db2. Nell'esempio seguente, sostituite *`master_username`* e *`master_password`* con le vostre informazioni.

   ```
   db2 "connect to rdsadmin user master_user using master_password"
   ```

2. Crea un buffer pool chiamando. `rdsadmin.create_bufferpool` Per ulteriori informazioni, consulta rdsadmin.create_bufferpool.

   ```
   db2 "call rdsadmin.create_bufferpool(
       'database_name',
       'buffer_pool_name',
       buffer_pool_size,
       'immediate',
       'automatic',
       page_size,
       number_block_pages,
       block_size)"
   ```

Modifica di un pool di buffer

Per modificare un pool di buffer per il database RDS for Db2, chiamate la stored procedure. `rdsadmin.alter_bufferpool` Per ulteriori informazioni, consulta la ALTER BUFFERPOOLdichiarazione nella documentazione. IBM Db2

Per modificare un pool di buffer

1. Connect al `rdsadmin` database utilizzando il nome utente principale e la password principale per l'istanza DB RDS for Db2. Nell'esempio seguente, sostituite *`master_username`* e *`master_password`* con le vostre informazioni.

   ```
   db2 "connect to rdsadmin user master_username using master_password"
   ```

2. Modifica un pool di buffer chiamando. `rdsadmin.alter_bufferpool` Per ulteriori informazioni, consulta rdsadmin.alter_bufferpool.

```
db2 "call rdsadmin.alter_bufferpool(
    'database_name',
    'buffer_pool_name',
    buffer_pool_size,
    'immediate',
    'automatic',
    change_number_blocks,
    number_block_pages,
    block_size)"
```

Eliminazione di un pool di buffer

Per eliminare un pool di buffer per il database RDS for Db2, chiamate la stored procedure. `rdsadmin.drop_bufferpool` Per ulteriori informazioni, consulta Dropping buffer pool nella documentazione. IBM Db2

> **Important**
>
> Assicurati che nessun tablespace sia assegnato al buffer pool che desideri eliminare.

Eliminare un pool di buffer

1. Connect al `rdsadmin` database utilizzando il nome utente principale e la password principale per l'istanza DB RDS for Db2. Nell'esempio seguente, sostituite *`master_username e master_password`* con le vostre informazioni.

```
db2 "connect to rdsadmin user master_user using master_password"
```

2. Elimina un pool di buffer chiamando. `rdsadmin.drop_bufferpool` Per ulteriori informazioni, consulta rdsadmin.drop_bufferpool.

```
db2 "call rdsadmin.drop_bufferpool(
    'database_name',
    'buffer_pool_name')"
```

Svuotare i buffer pool

È possibile svuotare i buffer pool per forzare un checkpoint in modo che RDS for Db2 scriva le pagine dalla memoria allo storage.

> **Note**
>
> Non è necessario svuotare i buffer pool. Db2 scrive i log in modo sincrono prima di eseguire le transazioni. Le pagine sporche potrebbero essere ancora in un pool di buffer, ma Db2 le scrive nello storage in modo asincrono. Anche se il sistema si spegne in modo imprevisto, al riavvio del database, Db2 esegue automaticamente il ripristino in caso di arresto anomalo. Durante il ripristino in caso di arresto anomalo, Db2 scrive le modifiche salvate nel database o ripristina le modifiche per le transazioni non eseguite.

Per svuotare i buffer pool

1. Connect al database Db2 utilizzando il nome utente principale e la password principale per l'istanza DB RDS for Db2. *Nell'esempio seguente, sostituisci rds_database_alias, master_username e master_password con le tue informazioni.*

   ```
   db2 connect to rds_database_alias user master_username using master_password
   ```

2. Svuota i buffer pool.

   ```
   db2 flush bufferpools all
   ```

Gestione dello storage

Db2 utilizza lo storage automatico per gestire lo storage fisico per oggetti di database come tabelle, indici e file temporanei. Invece di allocare manualmente lo spazio di archiviazione e tenere traccia dei percorsi di archiviazione utilizzati, lo storage automatico consente al sistema Db2 di creare e gestire i percorsi di archiviazione in base alle esigenze. Ciò può semplificare l'amministrazione dei database Db2 e ridurre la probabilità di errori dovuti a errori umani. Per ulteriori informazioni, consulta Archiviazione automatica nella IBM Db2 documentazione.

Con RDS for Db2, è possibile aumentare dinamicamente le dimensioni di archiviazione con l'espansione automatica dei volumi logici e del file system. Per ulteriori informazioni, consulta Uso dello storage per istanze database di Amazon RDS.

Gestione dei tablespace

È possibile creare, modificare, rinominare o eliminare tablespace per un database RDS for Db2. La creazione, la modifica, la ridenominazione o l'eliminazione di tablespace richiede un'autorità di livello superiore, che non è disponibile per l'utente principale. `SYSADM` Utilizza invece le stored procedure di Amazon RDS.

Argomenti

- Creazione di un tablespace
- Modificare un tablespace
- Rinominare un tablespace
- Eliminazione di un tablespace
- Verifica dello stato di un tablespace
- Restituzione di informazioni dettagliate sui tablespace
- Elencare lo stato e il gruppo di archiviazione per un tablespace
- Elencare i tablespace di una tabella
- Elenco dei contenitori di tablespace

Creazione di un tablespace

Per creare un tablespace per il database RDS for Db2, chiamate la stored procedure. `rdsadmin.create_tablespace` Per ulteriori informazioni, vedere la CREATE TABLESPACEdichiarazione nella documentazione. IBM Db2

> **Important**
>
> Per creare un tablespace, è necessario disporre di un pool di buffer della stessa dimensione di pagina da associare al tablespace. Per ulteriori informazioni, consulta Gestione dei buffer pool.

Per creare un tablespace

1. Connect al `rdsadmin` database utilizzando il nome utente principale e la password principale per l'istanza DB RDS for Db2. Nell'esempio seguente, sostituite *`master_username`* e *`master_password`* con le vostre informazioni.

```
db2 "connect to rdsadmin user master_username using master_password"
```

2. Crea un tablespace chiamando. `rdsadmin.create_tablespace` Per ulteriori informazioni, consulta rdsadmin.create_tablespace.

```
db2 "call rdsadmin.create_tablespace(
    'database_name',
    'tablespace_name',
    'buffer_pool_name',
    tablespace_initial_size,
    tablespace_increase_size,
    'tablespace_type')"
```

Modificare un tablespace

Per modificare un tablespace per il database RDS for Db2, chiamate la stored procedure. `rdsadmin.alter_tablespace` È possibile utilizzare questa procedura memorizzata per modificare il pool di buffer di una tablespace, abbassare il limite massimo o portare una tablespace online. Per ulteriori informazioni, vedere ALTER TABLESPACE la dichiarazione nella documentazione. IBM Db2

Per modificare un tablespace

1. Connect al `rdsadmin` database utilizzando il nome utente principale e la password principale per l'istanza DB RDS for Db2. Nell'esempio seguente, sostituite *`master_username`* e *`master_password`* con le vostre informazioni.

```
db2 "connect to rdsadmin user master_username using master_password"
```

2. Modifica un tablespace chiamando. `rdsadmin.alter_tablespace` Per ulteriori informazioni, consulta rdsadmin.alter_tablespace.

```
db2 "call rdsadmin.alter_tablespace(
    'database_name',
    'tablespace_name',
```

```
    'buffer_pool_name',
    buffer_pool_size,
    tablespace_increase_size,
    'max_size', 'reduce_max',
    'reduce_stop',
    'reduce_value',
    'lower_high_water',
    'lower_high_water_stop',
    'switch_online')"
```

Rinominare un tablespace

Per modificare il nome di un tablespace per il database RDS for Db2, chiama la stored procedure. `rdsadmin.rename_tablespace`

Per rinominare un tablespace

1. Connect al `rdsadmin` database utilizzando il nome utente principale e la password principale per l'istanza DB RDS for Db2. Nell'esempio seguente, sostituite *`master_username e master_password`* con le vostre informazioni.

   ```
   db2 "connect to rdsadmin user master_username using master_password"
   ```

2. Rinomina un tablespace chiamando. `rdsadmin.rename_tablespace` Per ulteriori informazioni, incluse le restrizioni su come denominare un tablespace, consulta. rdsadmin.rename_tablespace

   ```
   db2 "call rdsadmin.rename_tablespace(
       'database_name',
       'source_tablespace_name',
       'target_tablespace_name')"
   ```

Eliminazione di un tablespace

Per eliminare un tablespace per il database RDS for Db2, chiama la stored procedure. `rdsadmin.drop_tablespace` Prima di eliminare una tablespace, rilasciate tutti gli oggetti nella tablespace, ad esempio tabelle, indici o oggetti di grandi dimensioni (LOB). Per ulteriori informazioni, vedete Eliminare gli spazi delle tabelle nella documentazione. IBM Db2

Per eliminare un tablespace

1. Connect al `rdsadmin` database utilizzando il nome utente principale e la password principale per l'istanza DB RDS for Db2. Nell'esempio seguente, sostituite *`master_username e master_password`* con le vostre informazioni.

   ```
   db2 "connect to rdsadmin user master_username using master_password"
   ```

2. Elimina un tablespace chiamando. `rdsadmin.drop_tablespace` Per ulteriori informazioni, consulta rdsadmin.drop_tablespace.

   ```
   db2 "call rdsadmin.drop_tablespace(
       'database_name',
       'tablespace_name')"
   ```

Verifica dello stato di un tablespace

È possibile controllare lo stato di un tablespace utilizzando il comando. `cast`

Per controllare lo stato di un tablespace

1. Connect al database Db2 utilizzando il nome utente principale e la password principale per l'istanza DB RDS for Db2. *`Nell'esempio seguente, sostituisci rds_database_alias, master_username e master_password con le tue informazioni.`*

   ```
   db2 connect to rds_database_alias user master_username using master_password
   ```

2. Restituisce un output di riepilogo.

   Per un output riassuntivo:

   ```
   db2 "select cast(tbsp_id as smallint) as tbsp_id,
   cast(tbsp_name as varchar(35)) as tbsp_name,
   cast(tbsp_type as varchar(3)) as tbsp_type,
   cast(tbsp_state as varchar(10)) as state,
   cast(tbsp_content_type as varchar(8)) as contents from
    table(mon_get_tablespace(null,-1)) order by tbsp_id"
   ```

Restituzione di informazioni dettagliate sui tablespace

Per restituire informazioni dettagliate sui tablespace

1. Connect al database Db2 utilizzando il nome utente principale e la password principale per l'istanza DB RDS for Db2. *Nell'esempio seguente, sostituisci rds_database_alias, master_username e master_password con le tue informazioni.*

```
db2 connect to rds_database_alias user master_username using master_password
```

2. Restituisce i dettagli su tutti i tablespace del database per un membro o per tutti i membri.

   Per un membro:

```
db2 "select cast(member as smallint) as member,
cast(tbsp_id as smallint) as tbsp_id,
cast(tbsp_name as varchar(35)) as tbsp_name,
cast(tbsp_type as varchar(3)) as tbsp_type,
cast(tbsp_state as varchar(10)) as state,
cast(tbsp_content_type as varchar(8)) as contents,
cast(tbsp_total_pages as integer) as total_pages,
cast(tbsp_used_pages as integer) as used_pages,
cast(tbsp_free_pages as integer) as free_pages,
cast(tbsp_page_top as integer) as page_hwm,
cast(tbsp_page_size as integer) as page_sz,
cast(tbsp_extent_size as smallint) as extent_sz,
cast(tbsp_prefetch_size as smallint) as prefetch_sz,
cast(tbsp_initial_size as integer) as initial_size,
cast(tbsp_increase_size_percent as smallint) as increase_pct,
cast(storage_group_name as varchar(12)) as stogroup from
 table(mon_get_tablespace(null,-1)) order by member, tbsp_id "
```

   Per tutti i membri:

```
db2 "select cast(member as smallint) as member
cast(tbsp_id as smallint) as tbsp_id,
cast(tbsp_name as varchar(35)) as tbsp_name,
cast(tbsp_type as varchar(3)) as tbsp_type,
cast(tbsp_state as varchar(10)) as state,
cast(tbsp_content_type as varchar(8)) as contents,
cast(tbsp_total_pages as integer) as total_pages,
```

```
cast(tbsp_used_pages as integer) as used_pages,
cast(tbsp_free_pages as integer) as free_pages,
cast(tbsp_page_top as integer) as page_hwm,
cast(tbsp_page_size as integer) as page_sz,
cast(tbsp_extent_size as smallint) as extent_sz,
cast(tbsp_prefetch_size as smallint) as prefetch_sz,
cast(tbsp_initial_size as integer) as initial_size,
cast(tbsp_increase_size_percent as smallint) as increase_pct,
cast(storage_group_name as varchar(12)) as stogroup from
 table(mon_get_tablespace(null,-2)) order by member, tbsp_id "
```

Elencare lo stato e il gruppo di archiviazione per un tablespace

Per elencare lo stato e il gruppo di archiviazione di un tablespace, esegui la seguente istruzione SQL:

```
db2 "SELECT varchar(tbsp_name, 30) as tbsp_name,
                  varchar(TBSP_STATE, 30) state,
                  tbsp_type,
                  varchar(storage_group_name,30) storage_group
FROM TABLE(MON_GET_TABLESPACE('',-2)) AS t"
```

Elencare i tablespace di una tabella

Per elencare i tablespace di una tabella, esegui la seguente istruzione SQL. Nell'esempio seguente, sostituite *SCHEMA_NAME e TABLE_NAME con i nomi* dello schema e della tabella:

```
db2 "SELECT
   VARCHAR(SD.TBSPACE,30) AS DATA_SPACE,
   VARCHAR(SL.TBSPACE,30) AS LONG_SPACE,
   VARCHAR(SI.TBSPACE,30) AS INDEX_SPACE
 FROM
   SYSCAT.DATAPARTITIONS P
   JOIN SYSCAT.TABLESPACES SD ON SD.TBSPACEID = P.TBSPACEID
   LEFT JOIN SYSCAT.TABLESPACES SL ON SL.TBSPACEID = P.LONG_TBSPACEID
   LEFT JOIN SYSCAT.TABLESPACES SI ON SI.TBSPACEID = P.INDEX_TBSPACEID
 WHERE
    TABSCHEMA = 'SCHEMA_NAME'
 AND TABNAME   = 'TABLE_NAME'"
```

Elenco dei contenitori di tablespace

Per elencare i contenitori di tablespace per un tablespace

1. Connect al database Db2 utilizzando il nome utente principale e la password principale per l'istanza DB RDS for Db2. *Nell'esempio seguente, sostituisci rds_database_alias, master_username e master_password con le tue informazioni:*

```
db2 connect to rds_database_alias user master_username using master_password
```

2. Restituisce un elenco di tutti i contenitori di tablespace nel database o di contenitori di tablespace specifici.

   Per tutti i contenitori di tablespace:

```
db2 "select cast(member as smallint) as member,
cast(tbsp_name as varchar(35)) as tbsp_name,
cast(container_id as smallint) as id,
cast(container_name as varchar(60)) as container_path, container_type as type from
 table(mon_get_container(null,-2)) order by member,tbsp_id,container_id"
```

   Per contenitori tablespace specifici:

```
db2 "select cast(member as smallint) as member,
cast(tbsp_name as varchar(35)) as tbsp_name,
cast(container_id as smallint) as id,
cast(container_name as varchar(60)) as container_path, container_type as type from
 table(mon_get_container('TBSP_1',-2)) order by member, tbsp_id,container_id"
```

Generazione di report sulle prestazioni

È possibile generare report sulle prestazioni con una procedura o uno script. Per informazioni sull'utilizzo di una procedura, vedere la DBSUMMARYprocedura - Generazione di un rapporto di riepilogo delle metriche delle prestazioni del sistema e delle applicazioni nella IBM Db2 documentazione.

Db2 include un `db2mon.sh` file nella sua `~sqllib/sample/perf` directory. L'esecuzione dello script produce un report completo e a basso costo sulle metriche SQL. Per scaricare il `db2mon.sh` file e i file di script correlati, consulta la perfdirectory nel repository IBM db2-samples. GitHub

Per generare report sulle prestazioni con lo script

1. Connect al database Db2 utilizzando il nome utente principale e la password principale per l'istanza DB RDS for Db2. Nell'esempio seguente, sostituite *`master_username e master_password`* con le vostre informazioni.

   ```
   db2 connect to rdsadmin user master_username using master_password
   ```

2. Crea un pool di buffer denominato db2monbp con una dimensione di pagina di 4096 chiamando. `rdsadmin.create_bufferpool` Per ulteriori informazioni, consulta rdsadmin.create_bufferpool.

   ```
   db2 "call rdsadmin.create_bufferpool('database_name','db2monbp',4096)"
   ```

3. Crea un tablespace temporaneo denominato db2montmptbsp che utilizza il pool di db2monbp buffer chiamando. `rdsadmin.create_tablespace` Per ulteriori informazioni, consulta rdsadmin.create_tablespace.

   ```
   db2 "call rdsadmin.create_tablespace('database_name',\
       'db2montmptbsp','db2monbp',4096,1000,100,'T')"
   ```

4. Apri lo `db2mon.sh` script e modifica la riga relativa alla connessione a un database.

   a. Rimuovi la seguente riga.

      ```
      db2 -v connect to $dbName
      ```

   b. Sostituisci la riga del passaggio precedente con la riga seguente. Nell'esempio seguente, sostituite *`master_username e master_password con il nome utente principale e la password`* principale per l'istanza DB RDS for Db2.

      ```
      db2 -v connect to $dbName user master_username using master_password
      ```

5. Passate alla directory in cui si trova lo script. Nell'esempio seguente, sostituite la *`directory`* con il nome della directory in cui si trova lo script.

```
cd directory
```

6. Esegui lo db2mon.sh script per generare un report a intervalli specificati. Nell'esempio seguente, sostituisci *rds_database_alias* e *seconds con il nome del database e il numero di secondi* (da 0 a 3600) tra la generazione del report.

```
./db2mon.sh rds_database_alias seconds | tee -a db2mon.out
```

Raccolta di informazioni sui database

Puoi utilizzare una procedura memorizzata di Amazon RDS per raccogliere informazioni sui tuoi database. Queste informazioni possono aiutarti a monitorare i database o a risolvere i problemi.

Per raccogliere informazioni su un database

1. Connect al rdsadmin database utilizzando il nome utente principale e la password principale per l'istanza DB RDS for Db2. Nell'esempio seguente, sostituite *master_username e master_password* con le vostre informazioni.

```
db2 "connect to rdsadmin user master_username using master_password"
```

2. Raccogli informazioni chiamando. rdsadmin.db2pd Per ulteriori informazioni, consulta rdsadmin.db2pd_command.

```
db2 "call rdsadmin.db2pd_command('db2pd_cmd')"
```

Forzare le applicazioni a uscire dai database

Puoi utilizzare una procedura memorizzata di Amazon RDS per forzare l'uscita delle applicazioni dai tuoi database RDS for Db2 per consentire la manutenzione dei database.

Per forzare la rimozione delle applicazioni da un database

1. Connect al rdsadmin database utilizzando il nome utente principale e la password principale per l'istanza DB RDS for Db2. Nell'esempio seguente, sostituite *master_username e master_password* con le vostre informazioni.

```
db2 "connect to rdsadmin user master_username using master_password"
```

2. Forza le applicazioni a uscire da un database chiamando. `rdsadmin.force_application` Per ulteriori informazioni, consulta rdsadmin.force_application.

```
db2 "call rdsadmin.force_application(
    ?,
    'applications')"
```

Integrazione di un'istanza DB RDS per Db2 con Amazon S3

Puoi trasferire file tra la tua istanza DB RDS for Db2 e un bucket Amazon Simple Storage Service (Amazon S3) con le stored procedure di Amazon RDS. Per ulteriori informazioni, consulta Riferimento alla procedura memorizzata RDS per Db2.

> **Note**
>
> L'istanza database e il bucket Amazon S3 devono trovarsi nella stessa Regione AWS.

Affinché RDS for Db2 si integri con Amazon S3, l'istanza DB deve avere accesso a un bucket Amazon S3 in cui risiede il tuo RDS for Db2. Se al momento non disponi di un bucket S3, crea un bucket.

Argomenti

- Fase 1: Creazione di una policy IAM
- Fase 2: Crea un ruolo IAM e allega la tua policy IAM
- Passaggio 3: aggiungi il tuo ruolo IAM all'istanza DB RDS for Db2

Fase 1: Creazione di una policy IAM

In questo passaggio, crei una policy AWS Identity and Access Management (IAM) con le autorizzazioni necessarie per trasferire file dal bucket Amazon S3 all'istanza DB RDS. Questo passaggio presuppone che tu abbia già creato un bucket S3. Per ulteriori informazioni, consulta Creare un bucket nella Guida per l'utente di Amazon S3.

Prima di creare la policy, prendi nota delle seguenti informazioni:

- L'Amazon Resource Name (ARN) del bucket
- L'ARN per la tua chiave AWS Key Management Service (AWS KMS), se il tuo bucket utilizza SSE-KMS la crittografia. SSE-S3

Crea una policy IAM che includa le seguenti autorizzazioni:

```
"kms:GenerateDataKey",
"kms:Decrypt",
```

```
"s3:PutObject",
"s3:GetObject",
"s3:AbortMultipartUpload",
"s3:ListBucket",
"s3:DeleteObject",
"s3:GetObjectVersion",
"s3:ListMultipartUploadParts"
```

Puoi creare una policy IAM utilizzando AWS Management Console o il AWS Command Line Interface (AWS CLI).

Console

Per creare una policy IAM per consentire ad Amazon RDS l'accesso a un bucket Amazon S3

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, seleziona Policy.
3. Scegli Crea policy, quindi scegli JSON.
4. Aggiungi azioni per servizio. Per trasferire file da un bucket Amazon S3 ad Amazon RDS, devi selezionare le autorizzazioni del bucket e le autorizzazioni degli oggetti.
5. Espandi Resources (Risorse). È necessario specificare le risorse del bucket e dell'oggetto.
6. Seleziona Avanti.
7. Per Nome della politica, inserisci un nome per questa politica.
8. (Facoltativo) In Descrizione, inserire una descrizione per questa politica.
9. Scegli Crea policy.

AWS CLI

Per creare una policy IAM per consentire ad Amazon RDS l'accesso a un bucket Amazon S3

1. Esegui il comando create-policy. Nell'esempio seguente, sostituisci *iam_policy_name e s3_bucket_name con un nome per la tua policy IAM e il nome del bucket* Amazon S3 in cui risiede il tuo database RDS for Db2.

   Per, omacOS: Linux Unix

   ```
   aws iam create-policy \
   ```

```
    --policy-name iam_policy_name \
    --policy-document '{
      "Version": "2012-10-17",
      "Statement": [
        {
          "Effect": "Allow",
          "Action": [
            "kms:GenerateDataKey",
            "kms:Decrypt",
            "s3:PutObject",
            "s3:GetObject",
            "s3:AbortMultipartUpload",
            "s3:ListBucket",
            "s3:DeleteObject",
            "s3:GetObjectVersion",
            "s3:ListMultipartUploadParts"
          ],
          "Resource": [
            "arn:aws:s3:::s3_bucket_name/*",
            "arn:aws:s3:::s3_bucket_name"
          ]
        }
      ]
    }'
```

Per Windows:

```
aws iam create-policy ^
    --policy-name iam_policy_name ^
    --policy-document '{
      "Version": "2012-10-17",
      "Statement": [
        {
          "Effect": "Allow",
            "Action": [
              "s3:PutObject",
              "s3:GetObject",
              "s3:AbortMultipartUpload",
              "s3:ListBucket",
              "s3:DeleteObject",
              "s3:GetObjectVersion",
              "s3:ListMultipartUploadParts"
            ],
```

```
            "Resource": [
              "arn:aws:s3:::s3_bucket_name/*",
              "arn:aws:s3:::s3_bucket_name"
            ]
        }
      ]
    }'
```

2. Dopo aver creato la policy, annota l'ARN della policy. Ti serve l'ARN per. Fase 2: Crea un ruolo IAM e allega la tua policy IAM

Per informazioni sulla creazione di una policy IAM, consulta Creating IAM policies nella IAM User Guide.

Fase 2: Crea un ruolo IAM e allega la tua policy IAM

Questo passaggio presuppone che tu abbia creato la policy IAM in Fase 1: Creazione di una policy IAM. In questo passaggio, crei un ruolo IAM per la tua istanza DB RDS for Db2 e quindi alleghi la tua policy IAM al ruolo.

Puoi creare un ruolo IAM per la tua istanza DB utilizzando AWS Management Console o il. AWS CLI

Console

Per creare un ruolo IAM e allegare ad esso la tua policy IAM

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione, seleziona Roles (Ruoli).
3. Selezionare Create role (Crea ruolo).
4. Per il tipo di entità affidabile, seleziona Servizio AWS.
5. Per Servizio o caso d'uso, seleziona RDS, quindi seleziona RDS — Aggiungi ruolo al database.
6. Seleziona Avanti.
7. Per le politiche di autorizzazione, cerca e seleziona il nome della policy IAM che hai creato.
8. Seleziona Avanti.
9. In Role name, (Nome ruolo), inserisci un nome.
10. (Facoltativo) In Description (Descrizione), inserisci una descrizione per il nuovo ruolo.
11. Scegli Crea ruolo.

AWS CLI

Per creare un ruolo IAM e allegare ad esso la tua policy IAM

1. Esegui il comando create-role. Nell'esempio seguente, sostituisci *iam_role_name con un nome* per il tuo ruolo IAM.

   Per, o: Linux macOS Unix

```
aws iam create-role \
    --role-name iam_role_name \
    --assume-role-policy-document '{
      "Version": "2012-10-17",
      "Statement": [
        {
          "Effect": "Allow",
          "Principal": {
            "Service": "rds.amazonaws.com"
          },
          "Action": "sts:AssumeRole"
        }
      ]
    }'
```

   Per Windows:

```
aws iam create-role ^
    --role-name iam_role_name ^
    --assume-role-policy-document '{
      "Version": "2012-10-17",
      "Statement": [
        {
          "Effect": "Allow",
          "Principal": {
            "Service": "rds.amazonaws.com"
          },
          "Action": "sts:AssumeRole"
        }
      ]
    }'
```

2. Una volta creato il ruolo, annota l'ARN del ruolo. Ti serve l'ARN per. Passaggio 3: aggiungi il tuo ruolo IAM all'istanza DB RDS for Db2
3. Esegui il comando attach-role-policy. Nell'esempio seguente, sostituisci *iam_policy_arn con l'ARN* della policy IAM in cui hai creato. Fase 1: Creazione di una policy IAM Sostituisci *iam_role_name con il nome* del ruolo IAM che hai appena creato.

   Per, o: Linux macOS Unix

   ```
   aws iam attach-role-policy \
      --policy-arn iam_policy_arn \
      --role-name iam_role_name
   ```

   Per Windows:

   ```
   aws iam attach-role-policy ^
      --policy-arn iam_policy_arn ^
      --role-name iam_role_name
   ```

Per ulteriori informazioni, consulta la pagina relativa alla creazione di un ruolo per delegare le autorizzazioni a un utente IAM nella Guida per l'utente IAM.

Passaggio 3: aggiungi il tuo ruolo IAM all'istanza DB RDS for Db2

In questo passaggio, aggiungi il tuo ruolo IAM all'istanza DB RDS for Db2. Si notino i requisiti seguenti:

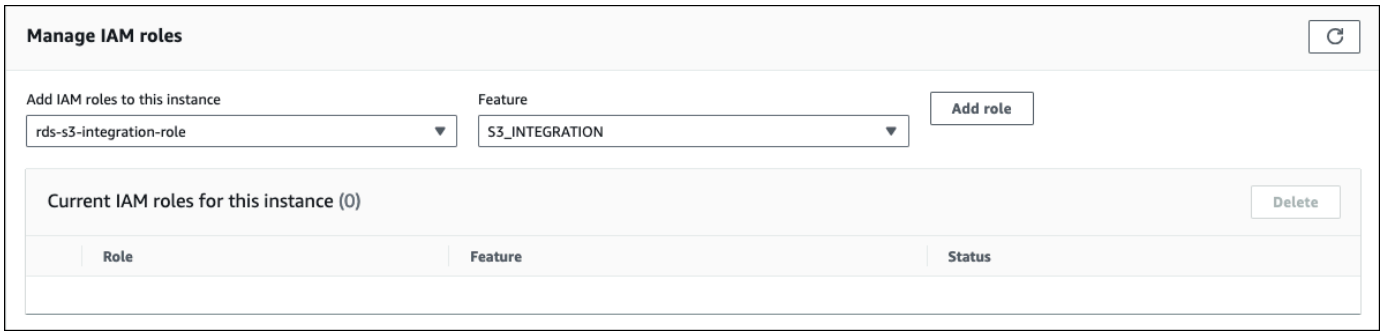
- Devi disporre dell'accesso a un ruolo a cui sono collegate le policy di autorizzazione di Amazon S3.
- Puoi associare un solo ruolo IAM alla tua istanza DB RDS for Db2 alla volta.
- L'istanza DB RDS for Db2 deve essere nello stato Available.

Puoi aggiungere un ruolo IAM alla tua istanza DB utilizzando AWS Management Console o il. AWS CLI

Console

Per aggiungere un ruolo IAM alla tua istanza DB RDS for Db2

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, scegli Databases (Database).
3. Scegli il nome dell'istanza DB RDS for Db2.
4. Sulla scheda Connettività e sicurezza, scorri verso il basso fino alla sezione Gestisci i ruoli IAM in fondo alla pagina.
5. Per Aggiungi i ruoli IAM a questa istanza, scegli il ruolo creato in Fase 2: Crea un ruolo IAM e allega la tua policy IAM.
6. Per Feature (Caratteristica), selezionare S3_INTEGRATION.
7. Scegliere Add role (Aggiungi ruolo).



AWS CLI

Per aggiungere un ruolo IAM all'istanza DB RDS for Db2, esegui il comando. add-role-to-db-instance Nell'esempio seguente, sostituisci *db_instance_name e iam_role_arn con il nome della tua istanza DB e l'ARN* del ruolo IAM in cui hai creato. Fase 2: Crea un ruolo IAM e allega la tua policy IAM

PerUnix, omacOS: Linux

```
aws rds add-role-to-db-instance \
    --db-instance-identifier db_instance_name \
    --feature-name S3_INTEGRATION \
    --role-arn iam_role_arn \
```

Per Windows:

```
aws rds add-role-to-db-instance ^
    --db-instance-identifier db_instance_name ^
    --feature-name S3_INTEGRATION ^
    --role-arn iam_role_arn ^
```

Per confermare che il ruolo è stato aggiunto correttamente all'istanza DB di RDS for Db2, esegui il describe-db-instancescomando. Nell'esempio seguente, sostituite *db_instance_name con il nome* dell'istanza DB.

Per, o: Linux macOS Unix

```
aws rds describe-db-instances \
    --filters "Name=db-instance-id,Values=db_instance_name" \
    --query 'DBInstances[].AssociatedRoles'
```

Per Windows:

```
aws rds describe-db-instances ^
    --filters "Name=db-instance-id,Values=db_instance_name" ^
    --query 'DBInstances[].AssociatedRoles'
```

Questo comando produce un output simile al seguente esempio:

```
[
    [
        {
            "RoleArn": "arn:aws:iam::0123456789012:role/rds-db2-s3-role",
            "FeatureName": "S3_INTEGRATION",
            "Status": "ACTIVE"
        }
    ]
]
```

Migrazione dei dati a Db2 su Amazon RDS

È possibile migrare i database Db2 autogestiti a RDS for Db2 utilizzando uno degli strumenti Db2 o quelli nativi. AWS

Argomenti

Approcci di migrazione che utilizzano AWS

Puoi eseguire una migrazione una tantum del tuo database Db2 da Linux o Windows verso Amazon RDS for Db2. AIX Per ridurre al minimo i tempi di inattività, puoi eseguire una migrazione con tempi di inattività quasi pari a zero. È inoltre possibile eseguire una migrazione sincrona tramite replica o utilizzo. AWS Database Migration Service

Per le migrazioni una tantum per database Db2 Linux basati, Amazon RDS supporta solo backup offline e online. Amazon RDS non supporta backup incrementali e Delta backup. Per migrazioni prossime allo zero per database Linux Db2 basati, Amazon RDS richiede backup online. Ti consigliamo di utilizzare i backup online per migrazioni con tempi di inattività vicini allo zero e i backup offline per le migrazioni in grado di gestire i tempi di inattività.

Argomenti

Migrazione una tantum da un ambiente Linux all'altro Linux

Con questo approccio di migrazione, esegui il backup del tuo database Db2 autogestito in un bucket Amazon S3. Quindi, utilizzi le procedure memorizzate di Amazon RDS per ripristinare il database Db2 su un'istanza database Amazon RDS for Db2. Per ulteriori informazioni sull'uso di Amazon S3, consulta. Integrazione di un'istanza DB RDS per Db2 con Amazon S3

Argomenti

Limitazioni e consigli per l'utilizzo del ripristino nativo

Le seguenti limitazioni e raccomandazioni si applicano all'utilizzo del ripristino nativo:

- Amazon RDS supporta solo backup offline e online per il ripristino nativo. Amazon RDS non supporta backup o Delta incrementali.
- Non puoi eseguire il ripristino da un bucket Amazon S3 in un Regione AWS ambiente diverso dalla regione in cui si trova l'istanza DB RDS for Db2.
- Non è possibile ripristinare un database se l'istanza DB RDS for Db2 contiene già un database.
- Amazon S3 limita la dimensione dei file caricati in un bucket Amazon S3 a 5 TB. Se il file di backup del database supera i 5 TB, suddividi il file di backup in file più piccoli.
- Amazon RDS non supporta routine esterne non limitate, ripristini incrementali o ripristini. Delta
- Non puoi eseguire il ripristino da un database di origine crittografato, ma puoi eseguire il ripristino in un'istanza database di Amazon RDS crittografata.

Quando ripristini il database, il backup viene copiato e quindi estratto sull'istanza DB RDS for Db2. Si consiglia di fornire uno spazio di archiviazione per l'istanza DB RDS for Db2 uguale o superiore alla somma delle dimensioni del backup più le dimensioni del database originale su disco.

La dimensione massima del database ripristinato è la dimensione massima del database supportata meno la dimensione del backup. Ad esempio, se la dimensione massima del database supportata è 64 TiB e la dimensione del backup è 30 TiB, la dimensione massima del database ripristinato è 34 TiB.

```
64 TiB - 30 TiB = 34 TiB
```

Configurazione di backup e ripristino nativi

Per il backup e il ripristino nativi, sono necessari i seguenti componenti: AWS

- Un bucket Amazon S3 per archiviare i tuoi file di backup: carica tutti i file di backup che desideri migrare su Amazon RDS. Ti consigliamo di utilizzare i backup offline per le migrazioni in grado di

gestire i tempi di inattività. Se hai già un bucket S3, puoi usare quel bucket. Se non disponi di un bucket S3, consulta Creazione di un bucket nella Amazon S3 User Guide.

> **Note**
>
> Se il database è di grandi dimensioni e il trasferimento in un bucket S3 richiederebbe molto tempo, puoi ordinare un AWS Snow Family dispositivo e chiedere di eseguire il backup. AWS Dopo aver copiato i file sul dispositivo e averli restituiti al team di Snow Family, il team trasferisce le immagini di backup nel bucket S3. Per ulteriori informazioni, consulta la documentazione relativa ad AWS Snow Family.

- Un ruolo IAM per accedere al bucket S3: se disponi già di un ruolo IAM, puoi utilizzare quel ruolo. Se non hai un ruolo, vedi. Fase 2: Crea un ruolo IAM e allega la tua policy IAM
- Una policy IAM con relazioni di fiducia e autorizzazioni associate al tuo ruolo IAM: per ulteriori informazioni, consulta. Fase 1: Creazione di una policy IAM
- Il ruolo IAM aggiunto alla tua istanza DB RDS for Db2: per ulteriori informazioni, consulta. Passaggio 3: aggiungi il tuo ruolo IAM all'istanza DB RDS for Db2

Ripristino del database Db2

Dopo la configurazione per il backup e il ripristino nativi, sei pronto per ripristinare il database Db2 sull'istanza DB RDS for Db2.

Per ripristinare il database Db2 sull'istanza DB RDS for Db2

1. Connect alla tua istanza DB RDS for Db2. Per ulteriori informazioni, consulta Connessione all'istanza DB RDS for Db2.
2. (Facoltativo) Per assicurarti che il database sia configurato con le impostazioni ottimali per l'operazione di ripristino, puoi chiamare per the section called “rdsadmin.show_configuration” `RESTORE_DATABASE_PARALLELISM` verificare i valori di e. `RESTORE_DATABASE_NUM_BUFFERS` Chiama the section called “rdsadmin.set_configuration” per modificare questi valori, se necessario. L'impostazione esplicita di questi valori può migliorare le prestazioni durante il ripristino di database con grandi volumi di dati.
3. Ripristina il database chiamando. `rdsadmin.restore_database` Per ulteriori informazioni, consulta rdsadmin.restore_database.

Migrazione con tempi di inattività quasi nulli per Linux database Db2 basati

Con questo approccio di migrazione, esegui la migrazione di un database Db2 Linux basato su Db2 da un database Db2 autogestito (origine) ad Amazon RDS for Db2. Questo approccio comporta interruzioni o tempi di inattività minimi o nulli per l'applicazione o gli utenti. Questo approccio esegue il backup del database e lo ripristina con la riproduzione dei log, che aiuta a prevenire interruzioni delle operazioni in corso e garantisce un'elevata disponibilità del database.

Per ottenere una migrazione con tempi di inattività quasi pari a zero, RDS for Db2 implementa il ripristino con replay dei log. Questo approccio esegue un backup del database Db2 Linux basato sull'autogestione e lo ripristina sul server RDS for Db2. Con le stored procedure di Amazon RDS, applichi quindi i log delle transazioni successivi per aggiornare il database.

Argomenti

Limiti e raccomandazioni sulla migrazione con tempi di inattività quasi nulli

Le seguenti limitazioni si applicano all'utilizzo di una migrazione con tempi di inattività vicini allo zero:

- Amazon RDS richiede un backup online per una migrazione con tempi di inattività quasi pari a zero. Questo perché Amazon RDS mantiene il database in uno stato di attesa di rollforward durante il caricamento dei log delle transazioni archiviati. Per ulteriori informazioni, consulta the section called “Migrazione del database Db2”.
- Non puoi eseguire il ripristino da un bucket Amazon S3 in un Regione AWS ambiente diverso dalla regione in cui si trova l'istanza DB RDS for Db2.
- Non è possibile ripristinare un database se l'istanza DB RDS for Db2 contiene già un database.
- Amazon S3 limita la dimensione dei file caricati in un bucket S3 a 5 TB. Se il file di backup del database supera i 5 TB, suddividi il file di backup in file più piccoli.
- Amazon RDS non supporta routine esterne non limitate, ripristini incrementali o ripristini. Delta
- Non puoi eseguire il ripristino da un database di origine crittografato, ma puoi eseguire il ripristino in un'istanza database di Amazon RDS crittografata.

Quando ripristini il database, Amazon RDS copia il backup e quindi lo estrae sull'istanza DB RDS for Db2. Ti consigliamo di fornire uno spazio di archiviazione per l'istanza DB RDS for Db2 uguale o superiore alla somma delle dimensioni del backup più le dimensioni del database originale su disco.

La dimensione massima del database ripristinato è la dimensione massima del database supportata meno la dimensione del backup. Ad esempio, se la dimensione massima del database supportata è 64 TiB e la dimensione del backup è 30 TiB, la dimensione massima del database ripristinato è 34 TiB.

```
64 TiB - 30 TiB = 34 TiB
```

Configurazione per una migrazione con tempi di inattività quasi nulli

Per una migrazione con tempi di inattività vicini allo zero, sono necessari i seguenti componenti: AWS

- Un bucket Amazon S3 per archiviare i tuoi file di backup: carica tutti i file di backup che desideri migrare su Amazon RDS. Amazon RDS richiede un backup online per una migrazione con tempi di inattività quasi pari a zero. Se hai già un bucket S3, puoi usare quel bucket. Se non disponi di un bucket S3, consulta Creazione di un bucket nella Amazon S3 User Guide.

  > **Note**
  >
  > Se il database è di grandi dimensioni e il trasferimento in un bucket S3 richiederebbe molto tempo, puoi ordinare un AWS Snow Family dispositivo e chiedere di eseguire il backup. AWS Dopo aver copiato i file sul dispositivo e averli restituiti al team di Snow Family, il team trasferisce le immagini di backup nel bucket S3. Per ulteriori informazioni, consulta la documentazione relativa ad AWS Snow Family.

- Un ruolo IAM per accedere al bucket S3: se disponi già di un ruolo AWS Identity and Access Management (IAM), puoi utilizzare quel ruolo. Se non hai un ruolo, vedi. Fase 2: Crea un ruolo IAM e allega la tua policy IAM
- Una policy IAM con relazioni di fiducia e autorizzazioni associate al tuo ruolo IAM: per ulteriori informazioni, consultaFase 1: Creazione di una policy IAM.
- Il ruolo IAM aggiunto alla tua istanza DB RDS for Db2: per ulteriori informazioni, consulta. Passaggio 3: aggiungi il tuo ruolo IAM all'istanza DB RDS for Db2

Migrazione del database Db2

Dopo la configurazione per una migrazione con tempi di inattività quasi pari a zero, sei pronto per migrare il database Db2 all'istanza DB RDS for Db2.

Per eseguire una migrazione con tempi di inattività vicini allo zero

1. Esegui un backup online del database di origine. Per ulteriori informazioni, consulta il BACKUP DATABASEcomando nella IBM Db2 documentazione.
2. Copia il backup del database in un bucket Amazon S3. Per informazioni sull'uso di Amazon S3, consulta la Guida per l'utente di Amazon Simple Storage Service.
3. Connettiti al `rdsadmin` server con *`master_username e master_password`* per la tua istanza DB RDS for Db2.

   ```
   db2 connect to rdsadmin user master_username using master_password
   ```

4. (Facoltativo) Per assicurarti che il database sia configurato con le impostazioni ottimali per l'operazione di ripristino, puoi chiamare per the section called "rdsadmin.show_configuration" verificare i valori di and. `RESTORE_DATABASE_PARALLELISM` `RESTORE_DATABASE_NUM_BUFFERS` Chiama the section called "rdsadmin.set_configuration" per modificare questi valori, se necessario. L'impostazione esplicita di questi valori può migliorare le prestazioni durante il ripristino di database con grandi volumi di dati.
5. Ripristina il backup sul server RDS for Db2 chiamando. `rdsadmin.restore_database` Imposta `backup_type` su `ONLINE`. Per ulteriori informazioni, consulta rdsadmin.restore_database.
6. Copia i log di archivio dal server di origine al bucket S3. Per ulteriori informazioni, consulta Archive logging nella documentazione. IBM Db2
7. Applica i log di archivio tutte le volte che è necessario chiamando. `rdsadmin.rollforward_database` Impostato `complete_rollforward` `FALSE` per mantenere il database in uno `ROLL-FORWARD PENDING` stato. Per ulteriori informazioni, consulta rdsadmin.rollforward_database.
8. Dopo aver applicato tutti i log di archivio, porta il database online `rdsadmin.complete_rollforward` chiamando. Per ulteriori informazioni, consulta rdsadmin.complete_rollforward.
9. Passa le connessioni delle applicazioni al server RDS for Db2 aggiornando gli endpoint dell'applicazione per il database o aggiornando gli endpoint DNS per reindirizzare il traffico al

server RDS for Db2. È inoltre possibile utilizzare la funzionalità di reindirizzamento automatico del client Db2 sul database Db2 autogestito con l'endpoint del database RDS for Db2. Per ulteriori informazioni, consulta la descrizione e la configurazione del reindirizzamento automatico del client nella documentazione. IBM Db2

10. (Facoltativo) Chiudi il database di origine.

Migrazione una tantum da AIX o Windows verso gli ambienti Linux

Con questo approccio di migrazione, utilizzi strumenti Db2 nativi per eseguire il backup del tuo database Db2 autogestito su un bucket Amazon S3. Gli strumenti Db2 nativi includono l'`export`utilità, il comando di sistema o il comando `db2move` di sistema. `db2look` Il tuo database Db2 può essere gestito autonomamente o in Amazon Elastic Compute Cloud (Amazon EC2). Puoi spostare i dati dal tuo Windows sistema AIX o al tuo bucket Amazon S3. Quindi, utilizza un client Db2 per caricare i dati direttamente dal bucket S3 al database RDS for Db2. I tempi di inattività dipendono dalla dimensione del database. Per ulteriori informazioni sull'uso di Amazon S3, consulta. Integrazione di un'istanza DB RDS per Db2 con Amazon S3

Per migrare il database Db2 a RDS for Db2

1. Preparati a eseguire il backup del tuo database. Configura una quantità di spazio di archiviazione sufficiente per archiviare il backup sul tuo sistema Db2 autogestito.
2. Esegui il backup del database.
    a. Esegui il comando di db2look sistema per estrarre il file DDL (Data Definition Language) per tutti gli oggetti.
    b. Eseguite l'utilità di esportazione Db2, il comando di db2move sistema o un'CREATE EXTERNAL TABLEistruzione per scaricare i dati della tabella Db2 nell'archivio del sistema Db2.
3. Sposta il backup in un bucket Amazon S3. Per ulteriori informazioni, consulta Integrazione di un'istanza DB RDS per Db2 con Amazon S3.

> **Note**
>
> Se il database è di grandi dimensioni e il trasferimento in un bucket S3 richiederebbe molto tempo, puoi ordinare un AWS Snow Family dispositivo e chiedere di AWS eseguire il backup. Dopo aver copiato i file sul dispositivo e averli restituiti al team di Snow

Family, il team trasferisce le immagini di backup nel bucket S3. Per ulteriori informazioni, consulta la documentazione relativa ad AWS Snow Family.

4. Utilizza un client Db2 per caricare i dati direttamente dal bucket S3 al database RDS for Db2.

Migrazioni sincrone da Linux un ambiente all'altro Linux

Con questo approccio di migrazione, configuri la replica tra il database Db2 autogestito e l'istanza DB RDS for Db2. Le modifiche apportate al database autogestito vengono replicate sull'istanza DB RDS for Db2 quasi in tempo reale. Questo approccio può garantire una disponibilità continua e ridurre al minimo i tempi di inattività durante il processo di migrazione.

Usando AWS Database Migration Service (AWS DMS)

Puoi utilizzarlo AWS DMS per migrazioni singole e poi sincronizzare da Db2 su Linux, Unix e Windows ad Amazon RDS for Db2. Per ulteriori informazioni, consulta What is? AWS Database Migration Service .

Strumenti Db2 nativi

Puoi utilizzare diversi strumenti, utilità e comandi Db2 nativi per spostare i dati da un database Db2 a un database Amazon RDS for Db2. Per utilizzare questi strumenti Db2 nativi, devi essere in grado di connettere la tua macchina client a un'istanza DB RDS for Db2. Per ulteriori informazioni, consulta Connessione di un computer client a un'istanza DB RDS for Db2.

Nome dello strumento	Caso d'uso	Limitazioni
db2look	Copia dei metadati da un database Db2 autogestito a un database RDS per Db2.	• È necessario modificare la sintassi per la creazione di pool di buffer, la creazione di tablespace e la creazione di ruoli in modo che corrisponda alla sintassi utilizzata da. Procedure memorizzate RDS per Db2
Comando IMPORT	Migrazione di tabelle piccole e tabelle con oggetti di grandi	• Più lenta dell'LOADutilità a causa delle operazioni

Nome dello strumento	Caso d'uso	Limitazioni
	dimensioni (LOB) da un computer client all'istanza DB RDS for Db2.	di registrazione. `INSERT` `DELETE` • Prestazioni scadenti con larghezza di banda di rete limitata.
INGESTutilità	Streaming continuo di dati da file e pipe senza oggetti di grandi dimensioni (LOB) sul computer client all'istanza DB RDS for Db2. Supporti e operazioni. `INSERT` `MERGE`	• Impossibile eseguire lo streaming di file di dati che contengono LOB. Utilizzate invece il `IMPORT` comando. • Connettività richiesta tra il database Db2 autogestito e il database RDS for Db2.
Comando INSERT	Copia dei dati in piccole tabelle da un database Db2 autogestito a un database RDS per Db2.	• Connettività richiesta tra il database Db2 autogestito e il database RDS for Db2. • Prestazioni scadenti con larghezza di banda di rete limitata.
Comando LOAD	Migrazione di tabelle di piccole dimensioni senza oggetti di grandi dimensioni (LOB) da un computer client all'istanza DB RDS for Db2.	• Impossibile migrare file di dati che contengono LOB. Utilizzate invece il `IMPORT` comando. • Prestazioni scadenti con larghezza di banda di rete limitata.

Connessione di un computer client a un'istanza DB RDS for Db2

Per utilizzare uno degli strumenti Db2 nativi per spostare i dati da un database Db2 a un database Amazon RDS per Db2, devi prima connettere la macchina client a un'istanza DB RDS for Db2.

La macchina client può essere una delle seguenti:

- Un'istanza Amazon Elastic Compute Cloud (Amazon EC2) su, o. Linux Windows macOS Questa istanza deve trovarsi nello stesso cloud privato virtuale (VPC) dell'istanza DB RDS for Db2, oppure. AWS Cloud9 AWS CloudShell
- Un'istanza Db2 autogestita in un'istanza Amazon EC2. Le istanze devono trovarsi nello stesso VPC.
- Un'istanza Db2 autogestita in un'istanza Amazon EC2. Le istanze possono trovarsi in diversi VPC se hai abilitato il peering VPC. Per ulteriori informazioni, consulta Creare una connessione peering VPC nella Amazon Virtual Private Cloud VPC Peering Guide.
- Una macchina locale in esecuzione o in un Linux ambiente Windows macOS autogestito. È necessario disporre di connettività pubblica a RDS for Db2 o abilitare la connettività VPN tra istanze Db2 autogestite e. AWS

Per connettere il computer client all'istanza DB RDS for Db2, accedi al computer client con. IBM Db2 Data Management Console Per ulteriori informazioni, consultare Creazione di un'istanza database Amazon RDS e IBM Db2 Data Management Console.

È possibile utilizzare AWS Database Migration Service (AWS DMS) per eseguire query sul database, eseguire un piano di esecuzione SQL e monitorare il database. Per ulteriori informazioni, consulta Cos'è il AWS Database Migration Service? nella Guida AWS Database Migration Service per l'utente.

Dopo aver collegato correttamente il computer client all'istanza DB RDS for Db2, è possibile utilizzare qualsiasi strumento Db2 nativo per copiare i dati. Per ulteriori informazioni, consulta Strumenti Db2 nativi.

db2lookstrumento

`db2look`è uno strumento Db2 nativo che estrae file DDL (Data Definition Language), oggetti, autorizzazioni, configurazioni, WLM e layout di database. È possibile utilizzare `db2look` per copiare i metadati del database da un database Db2 autogestito a un database RDS per Db2. Per ulteriori informazioni, consulta Imitazione dei database utilizzando db2look nella documentazione. IBM Db2

Per copiare i metadati del database

1. Esegui lo `db2look` strumento sul tuo sistema Db2 autogestito per estrarre il file DDL. Nell'esempio seguente, sostituisci *`database_name con il nome del tuo`* database Db2.

```
db2look -d database_name -e -l -a -f -wlm -cor -createdb -printdbcfg -o db2look.sql
```

2. Se il computer client ha accesso al database di origine (Db2 autogestito) e all'istanza DB RDS for Db2, è possibile creare il `db2look.sql` file sul computer client collegandolo direttamente all'istanza remota. Quindi cataloga l'istanza Db2 remota autogestita.

   a. Cataloga il nodo. Nell'esempio seguente, sostituite *dns_ip_address* e *port* con il nome DNS o l'indirizzo IP e il numero di porta del database Db2 autogestito.

   ```
   db2 catalog tcpip node srcnode REMOTE dns_ip_address server port
   ```

   b. Catalogare il database. Nell'esempio seguente, sostituite *source_database_name e source_database_alias con il nome del database Db2 autogestito e l'alias* che desiderate utilizzare per questo database.

   ```
   db2 catalog database source_database_name as source_database_alias at node
    srcnode \
       authentication server_encrypt
   ```

   c. Collega al database di origine. Nell'esempio seguente, sostituite *source_database_alias*, *user_id e user_password con l'alias creato nel passaggio precedente e l'ID utente e la password* per il database Db2 autogestito.

   ```
   db2look -d source_database_alias -i user_id -w user_password -e -l -a -f -wlm \
       -cor -createdb -printdbcfg -o db2look.sql
   ```

3. Se non riesci ad accedere al database Db2 remoto autogestito dal computer client, copia il file sul computer client. `db2look.sql` Quindi cataloga l'istanza DB RDS for Db2.

   a. Cataloga il nodo. Nell'esempio seguente, sostituite *dns_ip_address* e *port* con il nome DNS o l'indirizzo IP e il numero di porta dell'istanza DB RDS for Db2.

   ```
   db2 catalog tcpip node remnode REMOTE dns_ip_address server port
   ```

   b. Catalogare il database. Nell'esempio seguente, sostituite *rds_database_name e rds_database_alias con il nome del database RDS for Db2 e l'alias* che desiderate utilizzare per questo database.

   ```
   db2 catalog database rds_database_name as rds_database_alias at node remnode \
       authentication server_encrypt
   ```

c. Cataloga il database di amministrazione che gestisce RDS for Db2. Non puoi utilizzare questo database per archiviare dati.

```
db2 catalog database rdsadmin as rdsadmin at node remnode authentication
 server_encrypt
```

4. Crea pool di buffer e tablespace. L'amministratore non dispone dei privilegi per creare pool di buffer o tablespace. Tuttavia, puoi utilizzare le stored procedure di Amazon RDS per crearle.

a. Trova i nomi e le definizioni dei buffer pool e delle tablespace nel file. `db2look.sql`

b. Connect ad Amazon RDS utilizzando il nome utente principale e la password principale per l'istanza DB RDS for Db2. Nell'esempio seguente, sostituisci *`master_username`* e *`master_password`* con le tue informazioni.

```
db2 connect to rdsadmin user master_username using master_password
```

c. Crea un buffer pool chiamando. `rdsadmin.create_bufferpool` Per ulteriori informazioni, consulta rdsadmin.create_bufferpool.

```
db2 "call rdsadmin.create_bufferpool(
    'database_name',
    'buffer_pool_name',
    buffer_pool_size,
    'immediate',
    'automatic',
    page_size,
    number_block_pages,
    block_size)"
```

d. Crea un tablespace chiamando. `rdsadmin.create_tablespace` Per ulteriori informazioni, consulta rdsadmin.create_tablespace.

```
db2 "call rdsadmin.create_tablespace(
    'database_name',
    'tablespace_name',
    'buffer_pool_name',
    tablespace_initial_size,
    tablespace_increase_size,
    'tablespace_type')"
```

e. Ripeti i passaggi c o d per ogni buffer pool o tablespace aggiuntivo che desideri aggiungere.

f. Termina la connessione.

```
db2 terminate
```

5. Crea tabelle e oggetti.

a. Connect al database RDS for Db2 utilizzando il nome utente principale e la password principale per l'istanza DB RDS for Db2. *Nell'esempio seguente, sostituisci rds_database_name, master_username e master_password con le tue informazioni.*

```
db2 connect to rds_database_name user master_username using master_password
```

b. Esegui il file `db2look.sql`.

```
db2 -tvf db2look.sql
```

c. Termina la connessione.

```
db2 terminate
```

IMPORTcomando con una macchina client

Puoi utilizzare il `IMPORT` comando da un computer client per importare i dati nel server Amazon RDS for Db2.

> **Important**
>
> Il metodo di `IMPORT` comando è utile per migrare tabelle di piccole dimensioni e tabelle che includono oggetti di grandi dimensioni (LOB). Il `IMPORT` comando è più lento dell'`LOAD`Utilità a causa delle operazioni di registrazione`INSERT`. `DELETE` Se la larghezza di banda della rete tra il computer client e RDS for Db2 è limitata, si consiglia di utilizzare un approccio di migrazione diverso. Per ulteriori informazioni, consulta Strumenti Db2 nativi.

Per importare dati nel server RDS for Db2

1. Accedi al tuo computer client con. IBM Db2 Data Management Console Per ulteriori informazioni, consulta Connessione all'istanza DB RDS for Db2 con IBM Db2 Data Management Console.

2. Cataloga il database RDS for Db2 sul computer client.

   a. Cataloga il nodo. Nell'esempio seguente, sostituite *dns_ip_address* e *port* con il nome DNS o l'indirizzo IP e il numero di porta del database Db2 autogestito.

   ```
   db2 catalog tcpip node srcnode REMOTE dns_ip_address server port
   ```

   b. Catalogare il database. Nell'esempio seguente, sostituite *source_database_name e source_database_alias con il nome del database Db2 autogestito e l'alias* che desiderate utilizzare per questo database.

   ```
   db2 catalog database source_database_name as source_database_alias at node
    srcnode \
       authentication server_encrypt
   ```

3. Collega al database di origine. Nell'esempio seguente, sostituite *source_database_alias, user_id e user_password con l'alias creato nel passaggio precedente e l'ID utente e la password* per il database Db2 autogestito.

   ```
   db2look -d source_database_alias -i user_id -w user_password -e -l -a -f -wlm \
       -cor -createdb -printdbcfg -o db2look.sql
   ```

4. Genera il file di dati utilizzando il comando sul tuo sistema Db2 autogestito. `EXPORT` Nell'esempio seguente, sostituite la *directory* con la directory sul computer client in cui si trova il file di dati. Sostituite *file_name* e *table_name* con il nome del file di dati e il nome della tabella.

   ```
   db2 "export to /directory/file_name.txt of del lobs to /directory/lobs/ \
       modified by coldel\| select * from table_name"
   ```

5. Connect al database RDS for Db2 utilizzando il nome utente principale e la password principale per l'istanza DB RDS for Db2. *Nell'esempio seguente, sostituisci rds_database_alias, master_username e master_password con le tue informazioni.*

   ```
   db2 connect to rds_database_alias user master_username using master_password
   ```

6. Utilizzate il `IMPORT` comando per importare i dati da un file sul computer client nel database remoto RDS for Db2. Per ulteriori informazioni, vedete IMPORTil comando nella IBM Db2 documentazione. Nell'esempio seguente, sostituite *directory* e *nome_file* con la directory

sul computer client in cui esiste il file di dati e il nome del file di dati. Sostituite *SCHEMA_NAME* e *TABLE_NAME con il nome* dello schema e della tabella.

```
db2 "IMPORT from /directory/file_name.tbl OF DEL LOBS FROM /directory/lobs/ \
    modified by coldel\| replace into SCHEMA_NAME.TABLE_NAME"
```

7. Termina la connessione.

```
db2 terminate
```

INGESTutilità

Puoi utilizzare l'INGESTutilità per lo streaming continuo di dati da file e pipe su una macchina client verso un'istanza database Amazon RDS for Db2 di destinazione. L'INGESTutilità supporta INSERT e funziona. MERGE Per ulteriori informazioni, vedete l'utilità Ingest nella IBM Db2 documentazione.

Poiché l'INGESTutilità supporta i nickname, è possibile utilizzarla per trasferire dati dal database Db2 autogestito a un database RDS for Db2. Questo approccio funziona finché esiste la connettività di rete tra i due database.

> **Important**
>
> L'INGESTutilità non supporta oggetti di grandi dimensioni (LOB). Utilizzate invece il IMPORTcomando.

Per utilizzare la RESTARTABLE funzionalità dell'INGESTutilità, esegui il comando seguente nel database RDS for Db2.

```
db2 "call sysproc.sysinstallobjects('INGEST','C',NULL,NULL)"
```

INSERTcomando da un database Db2 autogestito a un database Amazon RDS per Db2

È possibile utilizzare il INSERT comando da un server Db2 autogestito per inserire i dati in un database RDS for Db2. Con questo approccio di migrazione, si utilizza un soprannome per l'istanza RDS for Db2 DB remota. Il database Db2 autogestito (origine) deve essere in grado di connettersi al database RDS for Db2 (destinazione).

> **Important**
>
> Il metodo di `INSERT` comando è utile per la migrazione di tabelle di piccole dimensioni. Se la larghezza di banda di rete tra il database Db2 autogestito e il database RDS for Db2 è limitata, si consiglia di utilizzare un approccio di migrazione diverso. Per ulteriori informazioni, consulta Strumenti Db2 nativi.

Per copiare i dati da un database Db2 autogestito a un database RDS for Db2

1. Cataloga l'istanza DB RDS for Db2 sull'istanza Db2 autogestita.

   a. Cataloga il nodo. Nell'esempio seguente, sostituite *dns_ip_address* e *port* con il nome DNS o l'indirizzo IP e il numero di porta del database Db2 autogestito.

   ```
   db2 catalog tcpip node remnode REMOTE dns_ip_address SERVER port
   ```

   b. Catalogare il database. Nell'esempio seguente, sostituisci *rds_database_name con il nome* del database sulla tua istanza DB RDS for Db2.

   ```
   db2 catalog database rds_database_name as remdb at node remnode \
       authentication server_encrypt
   ```

2. Abilita la federazione sull'istanza Db2 autogestita. Nell'esempio seguente, sostituisci *source_database_name con il nome* del tuo database sull'istanza Db2 autogestita.

   ```
   db2 update dbm cfg using FEDERATED YES source_database_name
   ```

3. Crea tabelle sull'istanza DB RDS for Db2.

   a. Cataloga il nodo. Nell'esempio seguente, sostituite *dns_ip_address* e *port* con il nome DNS o l'indirizzo IP e il numero di porta del database Db2 autogestito.

   ```
   db2 catalog tcpip node srcnode REMOTE dns_ip_address server port
   ```

   b. Catalogare il database. Nell'esempio seguente, sostituite *source_database_name e source_database_alias con il nome del database Db2 autogestito e l'alias* che desiderate utilizzare per questo database.

```
db2 catalog database source_database_name as source_database_alias at node
 srcnode \
    authentication server_encrypt
```

4. Collega al database di origine. Nell'esempio seguente, sostituite *source_database_alias*, *user_id e user_password con l'alias creato nel passaggio precedente e l'ID utente e la password* per il database Db2 autogestito.

```
db2look -d source_database_alias -i user_id -w user_password -e -l -a -f -wlm \
    -cor -createdb -printdbcfg -o db2look.sql
```

5. Imposta la federazione e crea un nickname per la tabella del database RDS for Db2 sull'istanza Db2 autogestita.

   a. Connect al database locale. Nell'esempio seguente, sostituisci *source_database_name con il nome* del database sulla tua istanza Db2 autogestita.

```
db2 connect to source_database_name
```

   b. Crea un wrapper per accedere alle sorgenti dati Db2.

```
db2 create wrapper drda
```

   c. Definisci una fonte di dati su un database federato. Nell'esempio seguente, sostituisci *admin* e *admin_password* con le tue credenziali per l'istanza Db2 autogestita. Sostituisci *rds_database_name con il nome* del database sulla tua istanza DB RDS for Db2.

```
db2 "create server rdsdb2 type DB2/LUW version '11.5.9.0' \
    wrapper drda authorization "admin" password "admin_password" \
    options( dbname 'rds_database_name', node 'remnode')"
```

   d. Mappa gli utenti sui due database. Nell'esempio seguente, sostituite *master_username e master_password* con le credenziali per l'istanza DB RDS for Db2.

```
db2 "create user mapping for user server rdsdb2 \
    options (REMOTE_AUTHID 'master_username', REMOTE_PASSWORD
 'master_password')"
```

   e. Verifica la connessione al server RDS for Db2.

```
db2 set passthru rdsdb2
```

f. Crea un soprannome per la tabella nel database remoto RDS for Db2. Nell'esempio seguente, sostituite *NICKNAME* e *TABLE_NAME con un soprannome per la tabella e il nome* della tabella.

```
db2 create nickname REMOTE.NICKNAME for RDSDB2.TABLE_NAME.NICKNAME
```

6. Inserite i dati nella tabella nel database remoto RDS for Db2. Usa il soprannome in un'`select`istruzione sulla tabella locale nell'istanza Db2 autogestita. Nell'esempio seguente, sostituite *NICKNAME* e *TABLE_NAME con un soprannome per la tabella e il nome* della tabella.

```
db2 "INSERT into REMOTE.NICKNAME select * from RDS2DB2.TABLE_NAME.NICKNAME"
```

LOADcomando con una macchina client

È possibile utilizzare il `LOAD CLIENT` comando per caricare i dati da un file al server RDS for Db2. Poiché non esiste alcuna connettività SSH per il server Amazon RDS for Db2, puoi utilizzare il `LOAD CLIENT` comando sul tuo server Db2 autogestito o sul tuo computer client Db2.

> **Important**
>
> Il metodo di `LOAD` comando è utile per migrare tabelle di piccole dimensioni. Se la larghezza di banda di rete tra il client e RDS for Db2 è limitata, si consiglia di utilizzare un approccio di migrazione diverso. Per ulteriori informazioni, consulta Strumenti Db2 nativi.
> Se il file di dati include riferimenti a nomi di file di oggetti di grandi dimensioni, il `LOAD` comando non funzionerà perché gli oggetti di grandi dimensioni (LOB) devono risiedere sul server Db2. Se tenti di caricare i LOB dal computer client al server RDS for Db2, riceverai un errore. SQL3025N Utilizzate invece il comando. IMPORT

Per caricare dati sul server RDS for Db2

1. Accedi al tuo computer client con. IBM Db2 Data Management Console Per ulteriori informazioni, consulta Connessione all'istanza DB RDS for Db2 con IBM Db2 Data Management Console.
2. Cataloga il database RDS for Db2 sul computer client.

a. Cataloga il nodo. Nell'esempio seguente, sostituite *dns_ip_address* e *port* con il nome DNS o l'indirizzo IP e il numero di porta del database Db2 autogestito.

```
db2 catalog tcpip node srcnode REMOTE dns_ip_address server port
```

b. Catalogare il database. Nell'esempio seguente, sostituite *source_database_name e source_database_alias con il nome del database Db2 autogestito e l'alias* che desiderate utilizzare per questo database.

```
db2 catalog database source_database_name as source_database_alias at node
 srcnode \
    authentication server_encrypt
```

3. Collega al database di origine. Nell'esempio seguente, sostituite *source_database_alias*, *user_id e user_password con l'alias creato nel passaggio precedente e l'ID utente e la password* per il database Db2 autogestito.

```
db2look -d source_database_alias -i user_id -w user_password -e -l -a -f -wlm \
    -cor -createdb -printdbcfg -o db2look.sql
```

4. Genera il file di dati utilizzando il comando sul tuo sistema Db2 autogestito. EXPORT Nell'esempio seguente, sostituite la *directory* con la directory sul computer client in cui si trova il file di dati. Sostituite *file_name* e *TABLE_NAME* con il nome del file di dati e il nome della tabella.

```
db2 "export to /directory/file_name.txt of del modified by coldel\| \
    select * from TPCH.TABLE_NAME"
```

5. Connect al database RDS for Db2 utilizzando il nome utente principale e la password principale per l'istanza DB RDS for Db2. *Nell'esempio seguente, sostituisci rds_database_alias, master_username e master_password con le tue informazioni.*

```
db2 connect to rds_database_alias user master_username using master_password
```

6. Utilizzate il LOAD comando per caricare i dati da un file sul computer client nel database remoto RDS for Db2. Per ulteriori informazioni, vedete LOADIl comando nella IBM Db2 documentazione. Nell'esempio seguente, sostituite la *directory* con la directory sul computer client in cui si

trova il file di dati. Sostituite *file_name* e *TABLE_NAME* con il nome del file di dati e il nome della tabella.

```
db2 "LOAD CLIENT from /directory/file_name.txt \
    modified by coldel\| replace into TPCH.TABLE_NAME \
    nonrecoverable without prompting"
```

7. Termina la connessione.

```
db2 terminate
```

Opzioni per RDS per istanze DB Db2

Di seguito vengono illustrate le opzioni o le funzionalità aggiuntive disponibili per le istanze Amazon RDS che eseguono il motore DB Db2. Per abilitare queste opzioni, puoi aggiungerle a un gruppo di opzioni personalizzato e quindi associare il gruppo di opzioni all'istanza database. Per ulteriori informazioni sull'utilizzo di gruppi di opzioni, consulta Uso di gruppi di opzioni.

Amazon RDS supporta le seguenti opzioni per Db2:

Opzione	ID opzione
Registrazione di controllo Db2	`DB2_AUDIT`

Registrazione di controllo Db2

Con la registrazione di controllo Db2, Amazon RDS registra l'attività del database, inclusi gli utenti che accedono al database e le query eseguite sul database. RDS carica i log di controllo completati nel tuo bucket Amazon S3, utilizzando il ruolo AWS Identity and Access Management (IAM) che fornisci.

Argomenti

Configurazione della registrazione di controllo Db2

Per abilitare la registrazione di controllo per un database RDS for Db2, si abilita l'`DB2_AUDIT`opzione sull'istanza DB RDS for Db2. Quindi, configura una politica di controllo per abilitare la funzionalità per il database specifico. Per abilitare l'opzione sull'istanza DB RDS for Db2, configurate le impostazioni delle opzioni per l'`DB2_AUDIT`opzione. A tale scopo, fornisci gli Amazon Resource Names (ARN) per il tuo bucket Amazon S3 e il ruolo IAM con le autorizzazioni per accedere al tuo bucket.

Per configurare la registrazione di controllo Db2 per un database RDS for Db2, completa i seguenti passaggi.

Argomenti

Fase 1: creazione di un bucket Amazon S3

Se non l'hai già fatto, crea un bucket Amazon S3 in cui Amazon RDS può caricare i file di log di controllo del database RDS for Db2. Per il bucket S3 utilizzato come destinazione dei file dell'audit valgono le seguenti restrizioni:

- Deve essere uguale alla tua istanza DB RDS Regione AWS for Db2.
- Non deve essere aperto al pubblico.
- Non può utilizzare S3 Object Lock.
- Il proprietario del bucket deve essere anche il proprietario del ruolo IAM.

Per informazioni su come creare un bucket Amazon S3, consulta Creating a bucket nella Amazon S3 User Guide.

Dopo aver abilitato la registrazione di controllo, Amazon RDS invia automaticamente i log dall'istanza DB alle seguenti posizioni:

- Registri a livello di istanza DB: *bucket_name*`/db2-audit-logs/`*dbi_resource_id*`/`*date_time_utc*`/`
- Registri a livello di database: *bucket_name*`/db2-audit-logs/`*dbi_resource_id*`/`*date_time_utc*`/`*db_name*`/`

Prendi nota dell'Amazon Resource Name (ARN) per il tuo bucket. Queste informazioni sono necessarie per completare i passaggi successivi.

Fase 2: Creare una policy IAM

Crea una policy IAM con le autorizzazioni necessarie per trasferire i file di log di controllo dall'istanza DB al bucket Amazon S3. Questo passaggio presuppone che tu disponga di un bucket S3.

Prima di creare la politica, raccogli le seguenti informazioni:

- L'ARN per il tuo bucket.
- L'ARN per la tua chiave AWS Key Management Service (AWS KMS), se il tuo bucket utilizza la crittografia. SSE-KMS

Crea una policy IAM che includa le seguenti autorizzazioni:

```
"s3:ListBucket",
 "s3:GetBucketACL",
 "s3:GetBucketLocation",
 "s3:PutObject",
 "s3:ListMultipartUploadParts",
 "s3:AbortMultipartUpload",
 "s3:ListAllMyBuckets"
```

> **Note**
>
> Amazon RDS necessita dell'`s3:ListAllMyBuckets`azione interna per verificare che lo stesso sia Account AWS proprietario sia del bucket S3 che dell'istanza DB RDS for Db2.

Se il tuo bucket utilizza la SSE-KMS crittografia, includi anche le seguenti autorizzazioni:

```
"kms:GenerateDataKey",
 "kms:Decrypt"
```

Puoi creare una policy IAM utilizzando AWS Management Console o il AWS Command Line Interface ()AWS CLI.

Console

Per creare una policy IAM per consentire ad Amazon RDS l'accesso a un bucket Amazon S3

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, seleziona Policy.
3. Scegli Crea policy, quindi scegli JSON.
4. In Aggiungi azioni, filtra per S3. Aggiungi accesso ListBucketGetBucketAcl, e GetBucketLocation.
5. Per Aggiungi una risorsa, scegli Aggiungi. Per Tipo di risorsa, scegli bucket e inserisci il nome del bucket. Quindi, scegli Aggiungi risorsa.
6. Scegli Aggiungi nuova dichiarazione.
7. In Aggiungi azioni, filtra per S3. Aggiungi accesso PutObjectListMultipartUploadParts, e AbortMultipartUpload.

8. Per Aggiungi una risorsa, scegli Aggiungi. Per Tipo di risorsa, scegli l'oggetto e *inserisci il nome del bucket/**. Quindi, scegli Aggiungi risorsa.
9. Scegli Aggiungi nuova dichiarazione.
10. In Aggiungi azioni, filtra per S3. Aggiungi accesso ListAllMyBuckets.
11. Per Aggiungi una risorsa, scegli Aggiungi. Per Tipo di risorsa, scegli Tutte le risorse. Quindi, scegli Aggiungi risorsa.
12. Se utilizzi le tue chiavi KMS per crittografare i dati:
    1. Scegli Aggiungi nuova dichiarazione.
    2. In Aggiungi azioni, filtra per KMS. Aggiungi accesso GenerateDataKeye decrittografa.
    3. Per Aggiungi una risorsa, scegli Aggiungi. Per Tipo di risorsa, scegli Tutte le risorse. Quindi, scegli Aggiungi risorsa.
13. Seleziona Avanti.
14. Per Nome della politica, inserisci un nome per questa politica.
15. (Facoltativo) In Descrizione, inserire una descrizione per questa politica.
16. Scegli Crea policy.

AWS CLI

Per creare una policy IAM per consentire ad Amazon RDS l'accesso a un bucket Amazon S3

1. Esegui il comando create-policy. Nell'esempio seguente, sostituisci *iam_policy_name e s3_bucket_name con un nome per la tua policy IAM e il nome del bucket* Amazon S3 di destinazione.

   PerUnix, omacOS: Linux

```
aws iam create-policy \
    --policy-name iam_policy_name \
    --policy-document '{
        "Version": "2012-10-17",
        "Statement": [
            {
                "Sid": "Statement1",
                "Effect": "Allow",
                "Action": [
                    "s3:ListBucket",
```

```
                    "s3:GetBucketAcl",
                    "s3:GetBucketLocation"
                ],
                "Resource": [
                    "arn:aws:s3:::s3_bucket_name"
                ]
            },
            {
                "Sid": "Statement2",
                "Effect": "Allow",
                "Action": [
                    "s3:PutObject",
                    "s3:ListMultipartUploadParts",
                    "s3:AbortMultipartUpload"
                ],
                "Resource": [
                    "arn:aws:s3:::s3_bucket_name/*"
                ]
            },
            {
                "Sid": "Statement3",
                "Effect": "Allow",
                "Action": [
                    "s3:ListAllMyBuckets"
                ],
                "Resource": [
                    "*"
                ]
            },
            {
                "Sid": "Statement4",
                "Effect": "Allow",
                "Action": [
                    "kms:GenerateDataKey",
                    "kms:Decrypt"
                ],
                "Resource": [
                    "*"
                ]
            }
        ]
}'
```

Per Windows:

```
aws iam create-policy ^
    --policy-name iam_policy_name ^
    --policy-document '{
        "Version": "2012-10-17",
        "Statement": [
            {
                "Sid": "Statement1",
                "Effect": "Allow",
                "Action": [
                    "s3:ListBucket",
                    "s3:GetBucketAcl",
                    "s3:GetBucketLocation"
                ],
                "Resource": [
                    "arn:aws:s3:::s3_bucket_name"
                ]
            },
            {
                "Sid": "Statement2",
                "Effect": "Allow",
                "Action": [
                    "s3:PutObject",
                    "s3:ListMultipartUploadParts",
                    "s3:AbortMultipartUpload"
                ],
                "Resource": [
                    "arn:aws:s3:::s3_bucket_name/*"
                ]
            },
            {
                "Sid": "Statement3",
                "Effect": "Allow",
                "Action": [
                    "s3:ListAllMyBuckets"
                ],
                "Resource": [
                    "*"
                ]
            },
            {
                "Sid": "Statement4",
```

```
                "Effect": "Allow",
                "Action": [
                    "kms:GenerateDataKey",
                    "kms:Decrypt"
                ],
                "Resource": [
                    "*"
                ]
            }
        ]
}'
```

2. Dopo aver creato la policy, annota l'ARN della policy. Ti serve l'ARN per. Fase 3: Crea un ruolo IAM e allega la tua policy IAM

Per informazioni sulla creazione di una policy IAM, consulta Creating IAM policies nella IAM User Guide.

Fase 3: Crea un ruolo IAM e allega la tua policy IAM

Questo passaggio presuppone che tu abbia creato la policy IAM inFase 2: Creare una policy IAM. In questo passaggio, crei un ruolo IAM per la tua istanza DB RDS for Db2 e quindi alleghi la tua policy IAM al ruolo.

Puoi creare un ruolo IAM per la tua istanza DB utilizzando la console o il. AWS CLI

Console

Per creare un ruolo IAM e allegare ad esso la tua policy IAM

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione, seleziona Roles (Ruoli).
3. Selezionare Create role (Crea ruolo).
4. Per il tipo di entità affidabile, seleziona Servizio AWS.
5. Per Servizio o caso d'uso, seleziona RDS, quindi seleziona RDS — Aggiungi ruolo al database.
6. Seleziona Avanti.
7. Per le politiche di autorizzazione, cerca e seleziona il nome della policy IAM che hai creato.
8. Seleziona Avanti.

9. In Role name, (Nome ruolo), inserisci un nome.
10. (Facoltativo) In Description (Descrizione), inserisci una descrizione per il nuovo ruolo.
11. Scegli Crea ruolo.

AWS CLI

Per creare un ruolo IAM e allegare ad esso la tua policy IAM

1. Esegui il comando create-role. Nell'esempio seguente, sostituisci *iam_role_name con un nome* per il tuo ruolo IAM.

   Per, o: Linux macOS Unix

```
aws iam create-role \
    --role-name iam_role_name \
    --assume-role-policy-document '{
      "Version": "2012-10-17",
      "Statement": [
        {
          "Effect": "Allow",
          "Principal": {
            "Service": "rds.amazonaws.com"
          },
          "Action": "sts:AssumeRole"
        }
      ]
    }'
```

   Per Windows:

```
aws iam create-role ^
    --role-name iam_role_name ^
    --assume-role-policy-document '{
      "Version": "2012-10-17",
      "Statement": [
        {
          "Effect": "Allow",
          "Principal": {
            "Service": "rds.amazonaws.com"
          },
          "Action": "sts:AssumeRole"
```

```
        }
      ]
    }'
```

2. Dopo aver creato il ruolo, annota l'ARN di questo ruolo. È necessario questo ARN per il passaggio successivo,. Fase 4: Configurare un gruppo di opzioni per la registrazione di controllo Db2
3. Esegui il comando attach-role-policy. Nell'esempio seguente, sostituisci *iam_policy_arn con l'ARN* della policy IAM in cui hai creato. Fase 2: Creare una policy IAM Sostituisci *iam_role_name con il nome* del ruolo IAM che hai appena creato.

   Per, o: Linux macOS Unix

   ```
   aws iam attach-role-policy \
      --policy-arn iam_policy_arn \
      --role-name iam_role_name
   ```

   Per Windows:

   ```
   aws iam attach-role-policy ^
      --policy-arn iam_policy_arn ^
      --role-name iam_role_name
   ```

Per ulteriori informazioni, consulta la pagina relativa alla creazione di un ruolo per delegare le autorizzazioni a un utente IAM nella Guida per l'utente IAM.

Fase 4: Configurare un gruppo di opzioni per la registrazione di controllo Db2

Il processo per aggiungere l'opzione di registrazione di controllo Db2 a un'istanza DB RDS for Db2 è il seguente:

1. Creare un nuovo gruppo di opzioni oppure copiare o modificare un gruppo di opzioni esistente.
2. Aggiungere e configurare tutte le opzioni necessarie.
3. Associare il gruppo di opzioni a questa istanza database.

Dopo aver aggiunto l'opzione di registrazione di controllo Db2, non è necessario riavviare l'istanza DB. Non appena il gruppo di opzioni diventa attivo, è possibile creare audit e archiviarne i log nel bucket S3.

Per aggiungere e configurare la registrazione di controllo Db2 sul gruppo di opzioni di un'istanza DB

1. Scegliere una delle seguenti opzioni:
   - Utilizzare un gruppo di opzioni esistente.
   - Crea un gruppo di opzioni DB personalizzato e usa quel gruppo di opzioni. Per ulteriori informazioni, consulta Creazione di un gruppo di opzioni.
2. Aggiungete l'opzione DB2_AUDIT al gruppo di opzioni e configurate le impostazioni delle opzioni. Per ulteriori informazioni sull'aggiunta di opzioni, consulta Aggiunta di un'opzione a un gruppo di opzioni.
   - Per IAM_ROLE_ARN, inserisci l'ARN del ruolo IAM in cui hai creato. the section called “Crea un ruolo IAM e allega la tua policy IAM”
   - Per S3_BUCKET_ARN, inserisci l'ARN del bucket S3 da utilizzare per i log di controllo Db2. Il bucket deve trovarsi nella stessa regione dell'istanza DB RDS for Db2. La policy associata al ruolo IAM che hai inserito deve consentire le operazioni richieste su questa risorsa.
3. Applicare il gruppo di opzioni a un'istanza database nuova o esistente. Scegliere una delle seguenti opzioni:
   - Se si sta creando una nuova istanza database, applicare il gruppo di opzioni quando viene avviata l'istanza.
   - In un'istanza database esistente, applicare il gruppo di opzioni modificando l'istanza e poi collegando il nuovo gruppo di opzioni. Per ulteriori informazioni, consulta Modifica di un'istanza database Amazon RDS.

Fase 5: Configurare la politica di controllo

Per configurare la politica di controllo per il database RDS for Db2, connettiti al `rdsadmin` database utilizzando il nome utente e la password principale per l'istanza DB RDS for Db2. Quindi, richiama la `rdsadmin.configure_db_audit` stored procedure con il nome DB del database e i valori dei parametri applicabili.

L'esempio seguente si connette al database e configura una politica di controllo per `testdb` le categorie AUDIT, CHECKING, OBJMAINT, SECMAINT, SYSADMIN e VALIDATE. Il valore di status `BOTH` registra gli esiti positivi e negativi e, per impostazione predefinita, è impostato su. `ERROR TYPE NORMAL` Per ulteriori informazioni su come utilizzare questa stored procedure, vedere. the section called “rdsadmin.configure_db_audit”

```
db2 "connect to rdsadmin user master_user using master_password"
db2 "call rdsadmin.configure_db_audit('testdb', 'ALL', 'BOTH', ?)"
```

Fase 6: Verificare la configurazione di controllo

Per assicurarti che la tua politica di controllo sia impostata correttamente, controlla lo stato della configurazione di controllo.

Per verificare la configurazione, connettiti al `rdsadmin` database utilizzando il nome utente principale e la password principale per l'istanza DB RDS for Db2. Quindi, esegui la seguente istruzione SQL con il nome DB del tuo database. Nell'esempio seguente, il nome del DB è *testdb.*

```
db2 "select task_id, task_type, database_name, lifecycle,
    varchar(bson_to_json(task_input_params), 500) as task_params,
    cast(task_output as varchar(500)) as task_output
    from table(rdsadmin.get_task_status(null,'testdb','CONFIGURE_DB_AUDIT'))"

Sample Output

TASK_ID              TASK_TYPE            DATABASE_NAME       LIFECYCLE
-------------------- -------------------- --------------- -------------
                   2  CONFIGURE_DB_AUDIT            DB2DB       SUCCESS

... continued ...
TASK_PARAMS
------------------------------------------------------------
{ "AUDIT_CATEGORY" : "ALL", "CATEGORY_SETTING" : "BOTH" }

... continued ...
                                        TASK_OUTPUT
--------------------------------------------------
2023-12-22T20:27:03.029Z Task execution has started.

2023-12-22T20:27:04.285Z Task execution has completed successfully.
```

Gestione della registrazione di controllo Db2

Dopo aver configurato la registrazione di controllo di Db2, è possibile modificare la politica di controllo per un database specifico o disabilitare la registrazione di controllo a livello di database o per l'intera istanza DB. Puoi anche modificare il bucket Amazon S3 in cui vengono caricati i file di registro.

Argomenti

- Modifica di una politica di audit Db2
- Modifica della posizione dei file di registro
- Disabilitazione della registrazione di controllo Db2

Modifica di una politica di audit Db2

Per modificare la politica di controllo per uno specifico database RDS for Db2, esegui la stored procedure. `rdsadmin.configure_db_audit` Con questa procedura memorizzata, è possibile modificare le categorie, le impostazioni delle categorie e la configurazione del tipo di errore della politica di controllo. Per ulteriori informazioni, consulta the section called “rdsadmin.configure_db_audit”.

Modifica della posizione dei file di registro

Per modificare il bucket Amazon S3 in cui vengono caricati i file di registro, esegui una delle seguenti operazioni:

- Modifica il gruppo di opzioni corrente collegato alla tua istanza DB RDS for Db2: aggiorna l'`S3_BUCKET_ARN`impostazione dell'`DB2_AUDIT`opzione in modo che punti al nuovo bucket. Inoltre, assicuratevi di aggiornare la policy IAM allegata al ruolo IAM specificato dall'`IAM_ROLE_ARN`impostazione nel gruppo di opzioni allegato. Questa policy IAM deve fornire al nuovo bucket le autorizzazioni di accesso richieste. Per informazioni sulle autorizzazioni richieste nella policy IAM, consulta. Creazione di una policy IAM
- Collega la tua istanza DB RDS for Db2 a un gruppo di opzioni diverso: modifica l'istanza DB per cambiare il gruppo di opzioni ad essa associato. Assicurati che il nuovo gruppo di opzioni sia configurato con le impostazioni `S3_BUCKET_ARN` e `IAM_ROLE_ARN` corrette. Per informazioni su come configurare queste impostazioni per l'`DB2_AUDIT`opzione, consultaConfigurare un gruppo di opzioni.

Quando modificate il gruppo di opzioni, assicuratevi di applicare immediatamente le modifiche. Per ulteriori informazioni, consulta the section called “Modifica di un'istanza database”.

Disabilitazione della registrazione di controllo Db2

Per disabilitare la registrazione di controllo Db2, effettuate una delle seguenti operazioni:

- Disattiva la registrazione di controllo per l'istanza DB RDS for Db2: modifica l'istanza DB e rimuovi il gruppo di opzioni che contiene l'opzione. `DB2_AUDIT` Per ulteriori informazioni, consulta the section called “Modifica di un'istanza database”.
- Disabilita la registrazione di controllo per un database specifico: interrompi la registrazione di controllo e rimuovi la politica di controllo chiamando `rdsadmin.disable_db_audit` con il nome DB del database. Per ulteriori informazioni, consulta the section called “rdsadmin.disable_db_audit”.

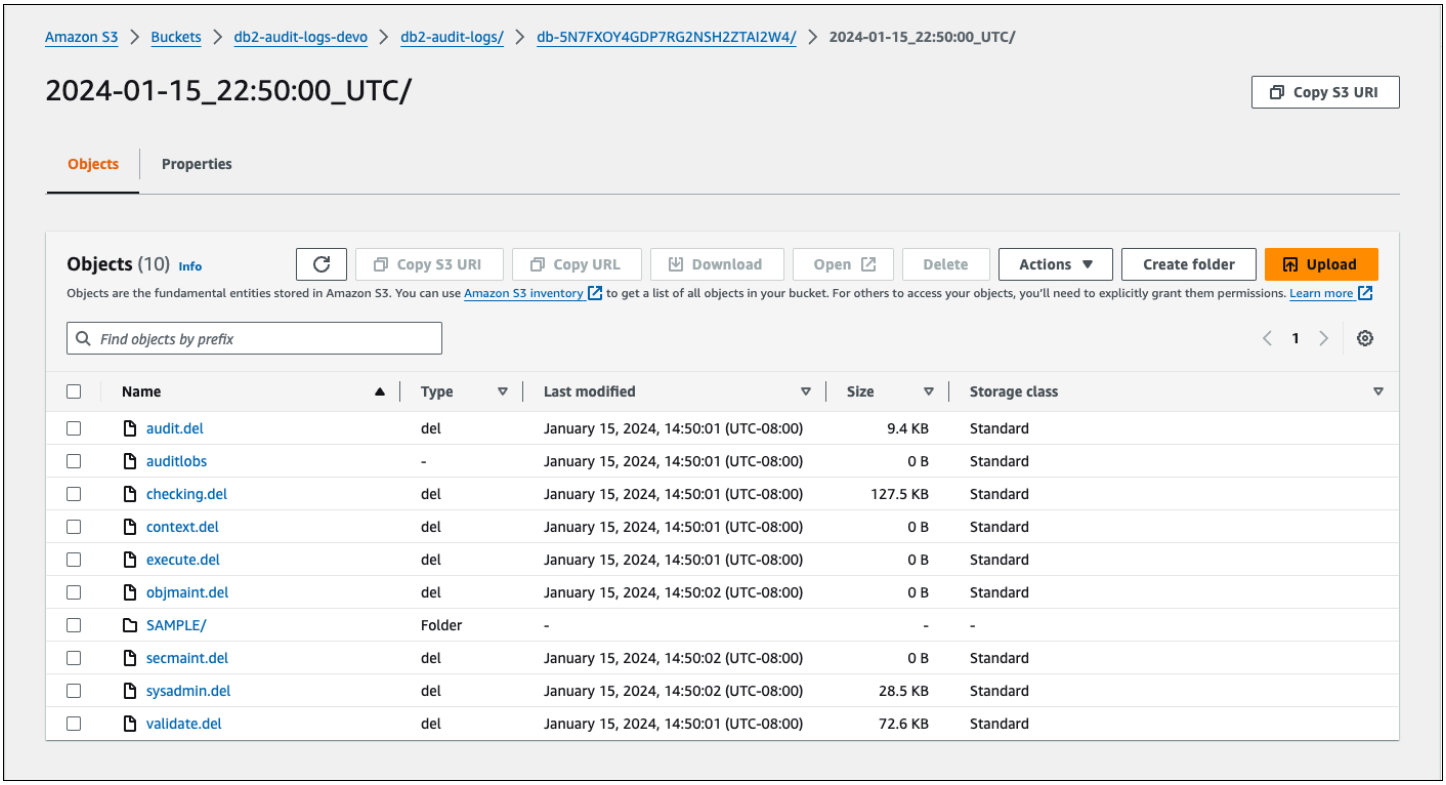
```
db2 "call rdsadmin.disable_db_audit(
    'db_name')"
```

Visualizzazione dei log di audit

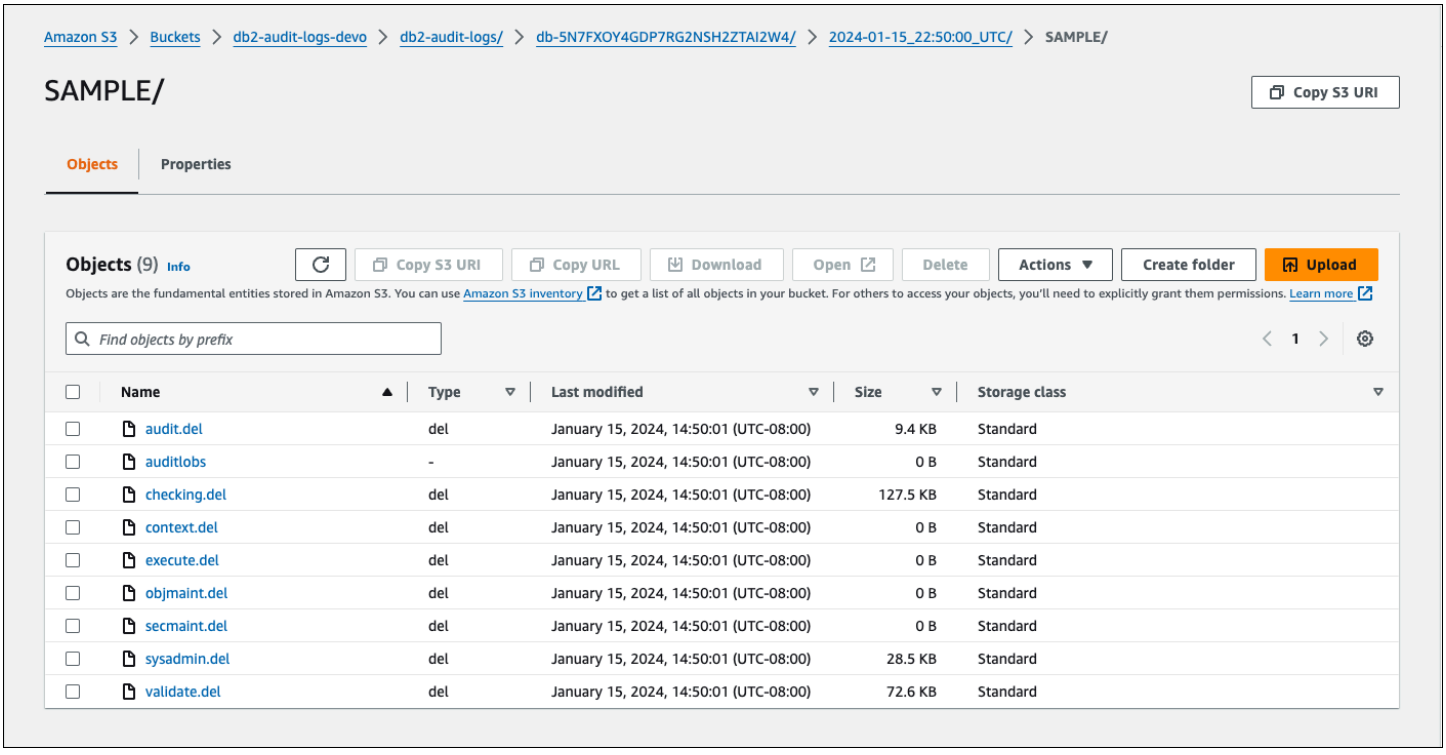
Dopo aver abilitato la registrazione di audit Db2, attendi almeno un'ora prima di visualizzare i dati di audit nel tuo bucket Amazon S3. Amazon RDS invia automaticamente i log dalla tua istanza DB RDS per Db2 alle seguenti posizioni:

- Registri a livello di istanza DB: *bucket_name*`/db2-audit-logs/`*dbi_resource_id*`/`*date_time_utc*`/`
- Registri a livello di database: *bucket_name*`/db2-audit-logs/`*dbi_resource_id*`/`*date_time_utc*`/`*db_name*`/`

La seguente schermata di esempio della console Amazon S3 mostra un elenco di cartelle per i file di registro a livello di istanza RDS for Db2 DB.



La seguente schermata di esempio della console Amazon S3 mostra i file di log a livello di database per l'istanza DB RDS for Db2.



Risoluzione dei problemi relativi alla registrazione di audit di Db2

Utilizzate le seguenti informazioni per risolvere i problemi più comuni relativi alla registrazione di controllo Db2.

Impossibile configurare la politica di controllo

Se la chiamata alla stored procedure `rdsadmin.configure_db_audit` restituisce un errore, è possibile che il gruppo di opzioni con l'`DB2_AUDIT`opzione non sia associato all'istanza DB RDS for Db2. Modifica l'istanza DB per aggiungere il gruppo di opzioni, quindi prova a chiamare nuovamente la stored procedure. Per ulteriori informazioni, consulta Modifica di un'istanza database Amazon RDS.

Nessun dato nel bucket Amazon S3

Se nel bucket Amazon S3 mancano dati di registrazione, verifica quanto segue:

- Il bucket Amazon S3 si trova nella stessa regione dell'istanza DB RDS for Db2.
- Il ruolo specificato nell'impostazione dell'`IAM_ROLE_ARN`opzione è configurato con le autorizzazioni necessarie per caricare i log nel bucket Amazon S3. Per ulteriori informazioni, consulta Creazione di una policy IAM.
- Gli ARN per le impostazioni `IAM_ROLE_ARN` e le `S3_BUCKET_ARN` opzioni sono corretti nel gruppo di opzioni associato all'istanza DB RDS for Db2. Per ulteriori informazioni, consulta Configurare un gruppo di opzioni.

È possibile verificare lo stato delle attività della configurazione della registrazione di controllo collegandosi al database ed eseguendo un'istruzione SQL. Per ulteriori informazioni, consulta Verificare la configurazione dell'audit.

Puoi anche controllare gli eventi per scoprire di più sul motivo per cui i log potrebbero mancare. Per informazioni su come visualizzare gli eventi, consultathe section called “Visualizzazione di registri, eventi e flussi nella console Amazon RDS”.

Procedure archiviate esterne per RDS for Db2

È possibile creare routine esterne e registrarle nei database RDS for Db2 come stored procedure esterne. Attualmente, RDS for Db2 supporta solo routine basate su Java per stored procedure esterne.

Procedure archiviate esterne basate su Java

Le stored procedure esterne basate su Java sono routine Java esterne registrate nel database RDS for Db2 come stored procedure esterne.

Argomenti

- Limitazioni per le stored procedure esterne basate su Java
- Configurazione di stored procedure esterne basate su Java

Limitazioni per le stored procedure esterne basate su Java

Prima di sviluppare una routine esterna, considerate le seguenti limitazioni e restrizioni.

Per creare una routine esterna, assicuratevi di utilizzare il Java Development Kit (JDK) fornito da Db2. Per ulteriori informazioni, consultate Supporto software Java per i prodotti di database Db2.

Il tuo programma Java può creare file solo nella `/tmp` directory e Amazon RDS non supporta l'abilitazione delle autorizzazioni eseguibili o Set User ID (SUID) su questi file. Inoltre, il tuo programma Java non può utilizzare le chiamate di sistema socket o le seguenti chiamate di sistema:

- _sysctl
- acct
- afs_syscall
- bpf
- capset
- chown
- chroot
- create_module
- delete_module

- fanotify_init
- fanotify_mark
- finit_module
- fsconfig
- fsopen
- fspick
- get_kernel_syms
- getpmsg
- init_module
- mount
- move_mount
- nfsservctl
- open_by_handle_at
- open_tree
- pivot_root
- putpmsg
- query_module
- quotactl
- reboot
- security
- setdomainname
- setfsuid
- sethostname
- sysfs
- tuxcall
- umount2
- uselib
- ustat
- vhangup
- vserver

Per ulteriori restrizioni sulle routine esterne per Db2, consulta Restrizioni sulle routine esterne nella documentazione. IBM Db2

Configurazione di stored procedure esterne basate su Java

Per configurare una stored procedure esterna, create un file.jar con la routine esterna, installatelo nel database RDS for Db2, quindi registratelo come stored procedure esterna.

Argomenti

- Passaggio 1: abilitare le stored procedure esterne
- Fase 2: Installare il file.jar con la routine esterna
- Fase 3: Registrare la stored procedure esterna
- Fase 4: Convalida della stored procedure esterna

Passaggio 1: abilitare le stored procedure esterne

Per abilitare le stored procedure esterne, in un gruppo di parametri personalizzato associato all'istanza DB, imposta il parametro `db2_alternate_authz_behaviour` su uno dei seguenti valori:

- `EXTERNAL_ROUTINE_DBADM`— Concede implicitamente l'autorizzazione a qualsiasi utente, gruppo o ruolo con `DBADM` autorità. `CREATE_EXTERNAL_ROUTINE`
- `EXTERNAL_ROUTINE_DBAUTH`— Consente a un utente autorizzato di concedere `DBADM` l'`CREATE_EXTERNAL_ROUTINE`autorizzazione a qualsiasi utente, gruppo o ruolo. In questo caso, a nessun utente, gruppo o ruolo viene implicitamente concessa questa autorizzazione, nemmeno a un utente con `DBADM` autorità.

Per ulteriori informazioni su questa impostazione, vedere l'istruzione GRANT (autorità del database) nella IBM Db2 documentazione.

Puoi creare e modificare un gruppo di parametri personalizzato utilizzando l' AWS Management Console AWS CLI, la o l'API Amazon RDS.

Console

Per configurare il parametro db2_alternate_authz_behavior in un gruppo di parametri personalizzato

1. Se desideri utilizzare un gruppo di parametri DB personalizzati diverso da quello utilizzato dall'istanza DB, crea un nuovo gruppo di parametri DB. Se utilizzi il modello Bring Your Own License (BYOL), assicurati che il nuovo gruppo di parametri personalizzati includa gli IBM ID. Per informazioni su questi ID, consulta. the section called “IBMID per Bring Your Own License for Db2” Per ulteriori informazioni sulla creazione di un gruppo di parametri database, consulta Creazione di un gruppo di parametri del database.
2. Imposta il valore per il `db2_alternate_authz_behaviour` parametro nel tuo gruppo di parametri personalizzato. Per ulteriori informazioni sulla modifica di un gruppo di parametri, vedereModifica di parametri in un gruppo di parametri del database.

AWS CLI

Per configurare il parametro db2_alternate_authz_behavior in un gruppo di parametri personalizzato

1. Se desideri utilizzare un gruppo di parametri DB personalizzati diverso da quello utilizzato dall'istanza DB, crea un gruppo di parametri personalizzato eseguendo il comando. create-db-parameter-group Se utilizzi il modello Bring Your Own License (BYOL), assicurati che il nuovo gruppo di parametri personalizzati includa gli IBM ID. Per informazioni su questi ID, consulta. the section called “IBMID per Bring Your Own License for Db2”

   Includi le seguenti opzioni obbligatorie:

   - `--db-parameter-group-name`— Un nome per il gruppo di parametri che state creando.
   - `--db-parameter-group-family`— L'edizione e la versione principale del motore Db2. I valori validi sono `db2-se-11.5` e `db2-ae-11.5`.
   - `--description`— Una descrizione per questo gruppo di parametri.

   Per ulteriori informazioni sulla creazione di un gruppo di parametri database, consulta Creazione di un gruppo di parametri del database.

   L'esempio seguente mostra come creare un gruppo di parametri personalizzato denominato `MY_EXT_SP_PARAM_GROUP` per la famiglia del gruppo di parametri`db2-se-11.5`.

   Per LinuxmacOS, oUnix:

```
aws rds create-db-parameter-group \
--region us-east-1 \
--db-parameter-group-name MY_EXT_SP_PARAM_GROUP \
--db-parameter-group-family db2-se-11.5 \
--description "test db2 external routines"
```

Per Windows:

```
aws rds create-db-parameter-group ^
--region us-east-1 ^
--db-parameter-group-name MY_EXT_SP_PARAM_GROUP ^
--db-parameter-group-family db2-se-11.5 ^
--description "test db2 external routines"
```

2. Modifica il `db2_alternate_authz_behaviour` parametro nel gruppo di parametri personalizzato eseguendo il modify-db-parameter-groupcomando.

   Includi le seguenti opzioni obbligatorie:

   - `--db-parameter-group-name`— Il nome del gruppo di parametri creato.
   - `--parameters`— Una matrice di nomi di parametri, valori e metodi di applicazione per l'aggiornamento dei parametri.

   Per ulteriori informazioni sulla modifica di un gruppo di parametri, vedereModifica di parametri in un gruppo di parametri del database.

   L'esempio seguente mostra come modificare il gruppo di parametri `MY_EXT_SP_PARAM_GROUP` impostando il valore di `db2_alternate_authz_behaviour` to`EXTERNAL_ROUTINE_DBADM`.

   Per LinuxmacOS, oUnix:

```
aws rds modify-db-parameter-group \
    --db-parameter-group-name MY_EXT_SP_PARAM_GROUP \
    --parameters
 "ParameterName='db2_alternate_authz_behaviour',ParameterValue='EXTERNAL_ROUTINE_DBADM',App
```

   Per Windows:

```
aws rds modify-db-parameter-group ^
```

```
    --db-parameter-group-name MY_EXT_SP_PARAM_GROUP ^
    --parameters
"ParameterName='db2_alternate_authz_behaviour',ParameterValue='EXTERNAL_ROUTINE_DBADM',App
```

API RDS

Per configurare il parametro db2_alternate_authz_behavior in un gruppo di parametri personalizzato

1. Se desideri utilizzare un gruppo di parametri DB personalizzato diverso da quello utilizzato dall'istanza DB, crea un nuovo gruppo di parametri DB utilizzando l'CreateDBParameterGroupoperazione API Amazon RDS. Se utilizzi il modello Bring Your Own License (BYOL), assicurati che il nuovo gruppo di parametri personalizzati includa gli ID. IBM Db2 Per informazioni su questi ID, consulta. the section called “IBMID per Bring Your Own License for Db2”

   Includi i parametri obbligatori seguenti:

   - `DBParameterGroupName`
   - `DBParameterGroupFamily`
   - `Description`

   Per ulteriori informazioni sulla creazione di un gruppo di parametri database, consulta Creazione di un gruppo di parametri del database.
2. Modifica il `db2_alternate_authz_behaviour` parametro nel gruppo di parametri personalizzato che hai creato utilizzando l'ModifyDBParameterGroupoperazione dell'API RDS.

   Includi i parametri obbligatori seguenti:

   - `DBParameterGroupName`
   - `Parameters`

   Per ulteriori informazioni sulla modifica di un gruppo di parametri, vedere. Modifica di parametri in un gruppo di parametri del database

Fase 2: Installare il file.jar con la routine esterna

Dopo aver creato la routine Java, create il file.jar ed eseguitelo `db2 "call sqlj.install_jar('file:`*`file_path`*`',`*`jar_ID`*`)"` per installarlo nel database RDS for Db2.

L'esempio seguente mostra come creare una routine Java e installarla su un database RDS for Db2. L'esempio include un codice di esempio per una routine semplice che è possibile utilizzare per testare il processo. In questo esempio si basano i seguenti presupposti:

- Il codice Java è compilato su un server in cui è installato Db2. Si tratta di una procedura consigliata perché la mancata compilazione con il JDK fornito da IBM può causare errori inspiegabili.
- Il database RDS for Db2 è catalogato localmente sul server.

Se desideri provare il processo con il seguente codice di esempio, copialo e salvalo in un file denominato. `MYJAVASP.java`

```
import java.sql.*;
public class MYJAVASP
{
public static void my_JAVASP (String inparam) throws SQLException, Exception
{
try
{
// Obtain the calling context's connection details.
Connection myConn = DriverManager.getConnection("jdbc:default:connection");
String myQuery = "INSERT INTO TEST.TEST_TABLE VALUES (?, CURRENT DATE)";
PreparedStatement myStmt = myConn.prepareStatement(myQuery);
myStmt.setString(1, inparam);
myStmt.executeUpdate();
}
catch (SQLException sql_ex)
{
throw sql_ex;
}
catch (Exception ex)
{
throw ex;
}
}
```

Il comando seguente compila la routine Java.

```
~/sqllib/java/jdk64/bin/javac MYJAVASP.java
```

Il comando seguente crea il file.jar.

```
~/sqllib/java/jdk64/bin/jar cvf MYJAVASP.jar MYJAVASP.class
```

I comandi seguenti si connettono al database denominato `MY_DB2_DATABASE` e installano il file.jar.

```
db2 "connect to MY_DB2_DATABASE user master_username using master_password"

db2 "call sqlj.install_jar('file:/tmp/MYJAVASP.jar','MYJAVASP')"
db2 "call sqlj.refresh_classes()"
```

Fase 3: Registrare la stored procedure esterna

Dopo aver installato il file.jar nel database RDS for Db2, registralo come stored procedure eseguendo il `db2 CREATE PROCEDURE` comando or. `db2 REPLACE PROCEDURE`

L'esempio seguente mostra come connettersi al database e registrare la routine Java creata nel passaggio precedente come stored procedure.

```
db2 "connect to MY_DB2_DATABASE user master_username using master_password"

create procedure TESTSP.MYJAVASP (in input char(6))
specific myjavasp
dynamic result sets 0
deterministic
language java
parameter style java
no dbinfo
fenced
threadsafe
modifies sql data
program type sub
external name 'MYJAVASP!my_JAVASP';
```

Fase 4: Convalida della stored procedure esterna

Utilizzate i seguenti passaggi per testare la procedura di memorizzazione esterna del campione registrata nel passaggio precedente.

Per convalidare la stored procedure esterna

1. Crea una tabella come `TEST.TEST_TABLE` nell'esempio seguente.

```
db2 "create table TEST.TEST_TABLE(C1 char(6), C2 date)"
```

2. Richiama la nuova stored procedure esterna. La chiamata restituisce lo stato di `0`.

```
db2 "call TESTSP.MYJAVASP('test')"
Return Status = 0
```

3. Interroga la tabella creata nel passaggio 1 per verificare i risultati della chiamata alla stored procedure.

```
db2 "SELECT * from TEST.TEST_TABLE"
```

   La query produce un output simile all'esempio seguente:

```
C1     C2
------ ----------
test   02/05/2024
```

Problemi e limitazioni noti per Amazon RDS for Db2

I seguenti elementi sono problemi e limitazioni noti relativi all'utilizzo di Amazon RDS for Db2.

Argomenti

Limitazione dell'autenticazione

Amazon RDS imposta `DB2AUTH` su. `JCC_ENFORCE_SECMEC` Poiché non `JCC_ENFORCE_SECMEC` può essere modificato, Amazon RDS applica la crittografia delle password sulle connessioni JDBC.

Routine non recintate

RDS per Db2 non supporta la creazione di routine non recintate. Per verificare se il database contiene routine non recintate, esegui il seguente comando SQL:

```
SELECT 'COUNT:' || count(*) FROM SYSCAT.ROUTINES where fenced='N' and routineschema not
 in ('SQLJ','SYSCAT','SYSFUN','SYSIBM','SYSIBMADM','SYSPROC','SYSTOOLS')
```

tablespace di archiviazione non automatici durante la migrazione

RDS per Db2 non supporta la creazione di nuovi tablespace di archiviazione non automatici. Quando si utilizza il ripristino nativo per una migrazione unica del database, RDS per Db2 converte automaticamente i tablespace di archiviazione non automatici in tablespace di archiviazione automatici, quindi ripristina il database in RDS per Db2. Per informazioni Migrazione una tantum da AIX o Windows verso gli ambienti Linux sulle migrazioni una Migrazione una tantum da un ambiente Linux all'altro Linux tantum, consulta e.

Riferimento alla procedura memorizzata RDS per Db2

Questi argomenti descrivono le procedure memorizzate nel sistema disponibili per le istanze Amazon RDS che eseguono il motore RDS for Db2. Per eseguire queste procedure, l'utente master deve prima connettersi al database. `rdsadmin`

Argomenti

Concessione e revoca dei privilegi

Le seguenti stored procedure concedono e revocano i privilegi per i database Amazon RDS for Db2. Per eseguire queste procedure, l'utente master deve prima connettersi al database. `rdsadmin`

Argomenti

rdsadmin.create_role

Crea un ruolo.

Sintassi

```
db2 "call rdsadmin.create_role(
    'database_name',
    'role_name')"
```

Parametri

I parametri seguenti sono obbligatori:

`database_name`

Il nome del database su cui verrà eseguito il comando. Il tipo di dati è`varchar`.

role_name

Il nome del ruolo che desideri creare. Il tipo di dati è`varchar`.

Note per l'utilizzo

Per informazioni sulla verifica dello stato della creazione di un ruolo, vedererdsadmin.get_task_status.

Esempi

L'esempio seguente crea un ruolo chiamato `MY_ROLE` database`DB2DB`.

```
db2 "call rdsadmin.create_role(
    'DB2DB',
    'MY_ROLE')"
```

rdsadmin.grant_role

Assegna un ruolo a un ruolo, utente o gruppo.

Sintassi

```
db2 "call rdsadmin.grant_role(
    ?,
    'database_name',
    'role_name',
    'grantee',
    'admin_option')"
```

Parametri

È richiesto il seguente parametro di output:

?

Un indicatore di parametro che restituisce l'identificatore univoco dell'attività. Questo parametro accetta solo. ?

Sono richiesti i seguenti parametri di input:

database_name

Il nome del database su cui verrà eseguito il comando. Il tipo di dati è`varchar`.

role_name

Il nome del ruolo che desideri creare. Il tipo di dati è`varchar`.

beneficiario

Il ruolo, l'utente o il gruppo a cui ricevere l'autorizzazione. Il tipo di dati è`varchar`. Valori validi: `ROLE`, `USER`, `GROUP`, `PUBLIC`.

Il formato deve essere un valore seguito dal nome. Separa più valori e nomi con virgole. Esempio: 'USER *user1*, *user2*, GROUP *group1*, *group2*'. Sostituisci i nomi con le tue informazioni.

Il seguente parametro di input è facoltativo:

admin_option

Speciifica se il beneficiario è autorizzato ad `ROLE` assegnare ruoli. `DBADM` Il tipo di dati è. `char` Il valore predefinito è `N`.

Note per l'utilizzo

Per informazioni sulla verifica dello stato dell'assegnazione di un ruolo, vederer[dsadmin.get_task_status](#).

Esempi

L'esempio seguente assegna un ruolo chiamato `ROLE_TEST` for database `TESTDB` al ruolo chiamato`role1`, all'utente chiamato `user1` e al gruppo chiamato. `group1` `ROLE_TEST`riceve l'autorizzazione di amministratore per assegnare ruoli.

```
db2 "call rdsadmin.grant_role(
    ?,
    'TESTDB',
    'ROLE_TEST',
    'ROLE role1, USER user1, GROUP group1',
    'Y')"
```

L'esempio seguente assegna un ruolo chiamato database `ROLE_TEST` `TESTDB` a. `PUBLIC` `ROLE_TEST`non riceve l'autorizzazione di amministratore per assegnare ruoli.

```
db2 "call rdsadmin.grant_role(
    ?,
    'TESTDB',
    'ROLE_TEST',
    'PUBLIC')"
```

rdsadmin.revoke_role

Revoca un ruolo da un ruolo, utente o gruppo.

Sintassi

```
db2 "call rdsadmin.revoke_role(
    ?,
    'database_name',
    'role_name',
    'grantee')"
```

Parametri

È richiesto il seguente parametro di output:

?

Un indicatore di parametro che restituisce l'identificatore univoco dell'attività. Questo parametro accetta solo?.

Sono richiesti i seguenti parametri di input:

database_name

Il nome del database su cui verrà eseguito il comando. Il tipo di dati è`varchar`.

role_name

Il nome del ruolo che desideri revocare. Il tipo di dati è. `varchar`

beneficiario

Il ruolo, l'utente o il gruppo a cui perdere l'autorizzazione. Il tipo di dati è`varchar`. Valori validi: `ROLE`, `USER`, `GROUP`, `PUBLIC`.

Il formato deve essere un valore seguito dal nome. Separa più valori e nomi con virgole. Esempio: 'USER *user1*, *user2*, GROUP *group1*, *group2*'. Sostituisci i nomi con le tue informazioni.

Note per l'utilizzo

Per informazioni sulla verifica dello stato dell'assegnazione di un ruolo, vedererdsadmin.get_task_status.

Esempi

L'esempio seguente revoca un ruolo chiamato `ROLE_TEST` per il database `TESTDB` dal ruolo chiamato`role1`, dall'utente chiamato e dal `user1` gruppo chiamato. `group1`

```
db2 "call rdsadmin.revoke_role(
    ?,
    'TESTDB',
    'ROLE_TEST',
    'ROLE role1, USER user1, GROUP group1')"
```

L'esempio seguente revoca un ruolo chiamato `ROLE_TEST` per il database da. `TESTDB` `PUBLIC`

```
db2 "call rdsadmin.revoke_role(
    ?,
    'TESTDB',
    'ROLE_TEST',
    'PUBLIC')"
```

rdsadmin.add_user

Aggiunge un utente a un elenco di autorizzazioni.

Sintassi

```
db2 "call rdsadmin.add_user(
    'username',
    'password',
```

```
    'group_name,group_name')"
```

Parametri

I parametri seguenti sono obbligatori:

username

Il nome utente di un utente. Il tipo di dati è`varchar`.

password

La password di un utente. Il tipo di dati è`varchar`.

Il parametro seguente è facoltativo:

group_name

Il nome di un gruppo a cui desideri aggiungere l'utente. Il tipo di dati è`varchar`. L'impostazione predefinita è una stringa vuota o nulla.

Note per l'utilizzo

È possibile aggiungere un utente a uno o più gruppi separando i nomi dei gruppi con virgole.

È possibile creare un gruppo quando si crea un nuovo utente o quando si aggiunge un gruppo a un utente esistente. Non puoi creare un gruppo da solo.

> **Note**
>
> Il numero massimo di utenti che puoi aggiungere chiamando `rdsadmin.add_user` è 5.000.

Per informazioni sulla verifica dello stato dell'aggiunta di un utente, consultardsadmin.get_task_status.

Esempi

L'esempio seguente crea un utente chiamato `jorge_souza` e lo assegna ai gruppi chiamati `sales` and`inside_sales`.

```
db2 "call rdsadmin.add_user(
```

```
    'jorge_souza',
    '*******',
    'sales,inside_sales')"
```

rdsadmin.change_password

Modifica la password di un utente.

Sintassi

```
db2 "call rdsadmin.change_password(
    'username',
    'new_password')"
```

Parametri

I parametri seguenti sono obbligatori:

username

Il nome utente di un utente. Il tipo di dati è`varchar`.

new_password

Una nuova password per l'utente. Il tipo di dati è`varchar`.

Note per l'utilizzo

Per informazioni sulla verifica dello stato della modifica di una password, vedererdsadmin.get_task_status.

Esempi

L'esempio seguente modifica la password per`jorge_souza`.

```
db2 "call rdsadmin.change_password(
    'jorge_souza',
    '*******')"
```

rdsadmin.list_users

Elenca gli utenti in un elenco di autorizzazioni.

Sintassi

```
db2 "call rdsadmin.list_users()"
```

Note per l'utilizzo

Per informazioni sulla verifica dello stato degli utenti delle inserzioni, consultardsadmin.get_task_status.

rdsadmin.remove_user

Rimuove l'utente dall'elenco delle autorizzazioni.

Sintassi

```
db2 "call rdsadmin.remove_user('username')"
```

Parametri

Il parametro seguente è obbligatorio:

username

Il nome utente di un utente. Il tipo di dati è`varchar`.

Note per l'utilizzo

Per informazioni sulla verifica dello stato della rimozione di un utente, vedererdsadmin.get_task_status.

Esempi

L'esempio seguente rimuove la possibilità `jorge_souza` di accedere ai database in RDS per le istanze DB Db2.

```
db2 "call rdsadmin.remove_user('jorge_souza')"
```

rdsadmin.add_groups

Aggiunge gruppi a un utente.

Sintassi

```
db2 "call rdsadmin.add_groups(
    'username',
    'group_name,group_name')"
```

Parametri

I parametri seguenti sono obbligatori:

username

Il nome utente di un utente. Il tipo di dati è`varchar`.

group_name

Il nome di un gruppo a cui desideri aggiungere l'utente. Il tipo di dati è`varchar`. L'impostazione predefinita è una stringa vuota.

Note per l'utilizzo

È possibile aggiungere uno o più gruppi a un utente separando i nomi dei gruppi con virgole. Per informazioni sulla verifica dello stato dell'aggiunta di gruppi, consulta. rdsadmin.get_task_status

Esempi

L'esempio seguente aggiunge i `b2b_sales` gruppi `direct_sales` e all'utente`jorge_souza`.

```
db2 "call rdsadmin.add_groups(
    'jorge_souza',
    'direct_sales,b2b_sales')"
```

rdsadmin.remove_groups

Rimuove i gruppi da un utente.

Sintassi

```
db2 "call rdsadmin.remove_groups(
    'username',
    'group_name,group_name')"
```

Parametri

I parametri seguenti sono obbligatori:

username

Il nome utente di un utente. Il tipo di dati è`varchar`.

group_name

Il nome di un gruppo da cui si desidera rimuovere l'utente. Il tipo di dati è`varchar`.

Note per l'utilizzo

È possibile rimuovere uno o più gruppi da un utente separando i nomi dei gruppi con virgole.

Per informazioni sulla verifica dello stato della rimozione dei gruppi, vedere. rdsadmin.get_task_status

Esempi

L'esempio seguente rimuove i `b2b_sales` gruppi `direct_sales` and dall'utente`jorge_souza`.

```
db2 "call rdsadmin.remove_groups(
    'jorge_souza',
    'direct_sales,b2b_sales')"
```

rdsadmin.dbadm_grant

Concede `DBADM` o `DATAACCESS` autorizza un ruolo, utente o gruppo. `ACCESSCTRL`

Sintassi

```
db2 "call rdsadmin.dbadm_grant(
    ?,
    'database_name',
    'authorization',
    'grantee')"
```

Parametri

È richiesto il seguente parametro di output:

?

Un indicatore di parametro che restituisce l'identificatore univoco dell'attività. Questo parametro accetta solo. ?

Sono richiesti i seguenti parametri di input:

database_name

Il nome del database su cui verrà eseguito il comando. Il tipo di dati è`varchar`.

authorization

Il tipo di autorizzazione da concedere. Il tipo di dati è`varchar`. Valori validi: `DBADM`, `ACCESSCTRL`, `DATAACCESS`.

Separa più tipi con virgole.

beneficiario

Il ruolo, l'utente o il gruppo a cui ricevere l'autorizzazione. Il tipo di dati è`varchar`. Valori validi: `ROLE`, `USER`, `GROUP`.

Il formato deve essere un valore seguito dal nome. Separa più valori e nomi con virgole. Esempio: 'USER *user1*, *user2*, GROUP *group1*, *group2*'. Sostituisci i nomi con le tue informazioni.

Note per l'utilizzo

Il ruolo per ricevere l'accesso deve esistere.

Per informazioni sulla verifica dello stato della concessione dell'accesso da amministratore del database, consultardsadmin.get_task_status.

Esempi

L'esempio seguente concede all'amministratore del database l'accesso al database denominato `TESTDB` per il ruolo. `ROLE_DBA`

```
db2 "call rdsadmin.dbadm_grant(
    ?,
    'TESTDB',
    'DBADM',
```

```
    'ROLE ROLE_DBA')"
```

L'esempio seguente concede all'amministratore del database l'accesso al database denominato andTESTDB. `user1` `group1`

```
db2 "call rdsadmin.dbadm_grant(
    ?,
    'TESTDB',
    'DBADM',
    'USER user1, GROUP group1')"
```

L'esempio seguente concede all'amministratore del database l'accesso al database denominato `TESTDB`user1, `user2`group1, e. `group2`

```
db2 "call rdsadmin.dbadm_grant(
    ?,
    'TESTDB',
    'DBADM',
    'USER user1, user2, GROUP group1, group2')"
```

rdsadmin.dbadm_revoke

Revoca `DBADM` o `DATAACCESS` autorizza un ruolo, utente o gruppo. `ACCESSCTRL`

Sintassi

```
db2 "call rdsadmin.dbadm_revoke(
    ?,
    'database_name',
    'authorization',
    'grantee')"
```

Parametri

È richiesto il seguente parametro di output:

?

Identificatore univoco dell'attività. Questo parametro accetta solo?.

Sono richiesti i seguenti parametri di input:

database_name

Il nome del database su cui verrà eseguito il comando. Il tipo di dati è`varchar`.

authorization

Il tipo di autorizzazione da revocare. Il tipo di dati è. `varchar` Valori validi: `DBADM`, `ACCESSCTRL`, `DATAACCESS`.

Separa più tipi con virgole.

beneficiario

Il ruolo, l'utente o il gruppo a cui revocare l'autorizzazione. Il tipo di dati è. `varchar` Valori validi: `ROLE`, `USER`, `GROUP`.

Il formato deve essere un valore seguito dal nome. Separa più valori e nomi con virgole. Esempio: 'USER *user1*, *user2*, GROUP *group1*, *group2*'. Sostituisci i nomi con le tue informazioni.

Note per l'utilizzo

Per informazioni sulla verifica dello stato della revoca dell'accesso all'amministratore del database, vedere. rdsadmin.get_task_status

Esempi

L'esempio seguente revoca l'accesso dell'amministratore del database al database denominato `TESTDB` per il ruolo. `ROLE_DBA`

```
db2 "call rdsadmin.dbadm_revoke(
    ?,
    'TESTDB',
    'DBADM',
    'ROLE ROLE_DBA')"
```

L'esempio seguente revoca l'accesso dell'amministratore del database al database denominato and. `TESTDB` `user1` `group1`

```
db2 "call rdsadmin.dbadm_revoke(
    ?,
    'TESTDB',
    'DBADM',
```

```
    'USER user1, GROUP group1')"
```

L'esempio seguente revoca l'accesso dell'amministratore del database al database denominato `TESTDB` per`user1`,, `user2` e. `group1` `group2`

```
db2 "call rdsadmin.dbadm_revoke(
    ?,
    'TESTDB',
    'DBADM',
    'USER user1, user2, GROUP group1, group2')"
```

Gestione dei buffer pool

Le seguenti stored procedure gestiscono i buffer pool per i database Amazon RDS for Db2. Per eseguire queste procedure, l'utente master deve prima connettersi al database. `rdsadmin`

Argomenti

- rdsadmin.create_bufferpool
- rdsadmin.alter_bufferpool
- rdsadmin.drop_bufferpool

rdsadmin.create_bufferpool

Crea un pool di buffer.

Sintassi

```
db2 "call rdsadmin.create_bufferpool(
    'database_name',
    'buffer_pool_name',
    buffer_pool_size,
    'immediate',
    'automatic',
    page_size,
    number_block_pages,
    block_size)"
```

Parametri

I parametri seguenti sono obbligatori:

database_name

Il nome del database su cui eseguire il comando. Il tipo di dati è`varchar`.

buffer_pool_name

Il nome del buffer pool da creare. Il tipo di dati è`varchar`.

I parametri seguenti sono facoltativi:

buffer_pool_size

La dimensione del buffer pool in numero di pagine. Il tipo di dati è`integer`. Il valore predefinito è `-1`.

immediato

Speciifica se il comando viene eseguito immediatamente. Il tipo di dati è`char`. Il valore predefinito è Y.

automatico

Speciifica se impostare il pool di buffer su automatico. Il tipo di dati è. `char` Il valore predefinito è Y.

dimensione della pagina

La dimensione della pagina del buffer pool. Il tipo di dati è`integer`. Valori validi: `4096`, `8192`, `16384`, `32768`. Il valore predefinito è `8192`.

number_block_pages

Il numero di pagine bloccate nei buffer pool. Il tipo di dati è`integer`. Il valore predefinito è `0`.

block_size

La dimensione del blocco per le pagine bloccate. Il tipo di dati è`integer`. Valori validi: da `2` a `256`. Il valore predefinito è `32`.

Note per l'utilizzo

Per informazioni sulla verifica dello stato della creazione di un pool di buffer, vedererdsadmin.get_task_status.

Esempi

L'esempio seguente crea un pool di buffer chiamato `BP8` per un database chiamato `TESTDB` con parametri predefiniti, pertanto il pool di buffer utilizza una dimensione di pagina di 8 KB.

```
db2 "call rdsadmin.create_bufferpool(
    'TESTDB',
    BP8')"
```

L'esempio seguente crea un buffer pool chiamato `BP16` per un database denominato `TESTDB` che utilizza una dimensione di pagina di 16 KB con un numero iniziale di pagine di 1.000 ed è impostato

su automatico. Db2 esegue il comando immediatamente. Se si utilizza un numero iniziale di pagine pari a -1, Db2 utilizzerà l'allocazione automatica delle pagine.

```
db2 "call rdsadmin.create_bufferpool(
    'TESTDB',
    'BP16',
    1000,
    'Y',
    'Y',
    16384)"
```

L'esempio seguente crea un pool di buffer chiamato BP16 per un database chiamato. TESTDB Questo pool di buffer ha una dimensione di pagina di 16 KB con un numero iniziale di pagine di 10.000. Db2 esegue immediatamente il comando utilizzando 500 pagine di blocchi con una dimensione del blocco di 512.

```
db2 "call rdsadmin.create_bufferpool(
    'TESTDB',
    'BP16',
    10000,
    'Y',
    'Y',
    16384,
    500,
    512)"
```

rdsadmin.alter_bufferpool

Modifica un pool di buffer.

Sintassi

```
db2 "call rdsadmin.alter_bufferpool(
    'database_name',
    'buffer_pool_name',
    buffer_pool_size,
    'immediate',
    'automatic',
    change_number_blocks,
    number_block_pages,
    block_size)"
```

Parametri

I parametri seguenti sono obbligatori:

database_name

Il nome del database su cui eseguire il comando. Il tipo di dati è`varchar`.

buffer_pool_name

Il nome del buffer pool da modificare. Il tipo di dati è`varchar`.

buffer_pool_size

La dimensione del buffer pool in numero di pagine. Il tipo di dati è`integer`.

I parametri seguenti sono facoltativi:

immediato

Speciifica se il comando viene eseguito immediatamente. Il tipo di dati è`char`. Il valore predefinito è `Y`.

automatico

Speciifica se impostare il pool di buffer su automatico. Il tipo di dati è. `char` Il valore predefinito è `N`.

change_number_blocks

Speciifica se è stata apportata una modifica al numero di pagine bloccate nel pool di buffer. Il tipo di dati è. `char` Il valore predefinito è `N`.

number_block_pages

Il numero di pagine bloccate nei buffer pool. Il tipo di dati è`integer`. Il valore predefinito è `0`.

block_size

La dimensione del blocco per le pagine bloccate. Il tipo di dati è`integer`. Valori validi: da `2` a `256`. Il valore predefinito è `32`.

Note per l'utilizzo

Per informazioni sulla verifica dello stato di modifica di un buffer pool, vedere. rdsadmin.get_task_status

Esempi

L'esempio seguente modifica un pool di buffer chiamato non automatico BP16 per un database chiamato non automatico e ne modifica la dimensione TESTDB a 10.000 pagine. Db2 esegue questo comando immediatamente.

```
db2 "call rdsadmin.alter_bufferpool(
    'TESTDB',
    'BP16',
    10000,
    'Y',
    'N')"
```

rdsadmin.drop_bufferpool

Elimina un buffer pool.

Sintassi

```
db2 "call rdsadmin.drop_bufferpool(
    'database_name',
    'buffer_pool_name'"
```

Parametri

I parametri seguenti sono obbligatori:

database_name

Il nome del database a cui appartiene il buffer pool. Il tipo di dati è`varchar`.

buffer_pool_name

Il nome del buffer pool da eliminare. Il tipo di dati è`varchar`.

Note per l'utilizzo

Per informazioni sulla verifica dello stato dell'eliminazione di un pool di buffer, vedere. rdsadmin.get_task_status

Esempi

L'esempio seguente elimina un pool di buffer chiamato BP16 per un database chiamato. TESTDB

```
db2 "call rdsadmin.drop_bufferpool(
    'TESTDB',
    'BP16')"
```

Gestione dei database

Le seguenti stored procedure gestiscono i database per Amazon RDS for Db2. Per eseguire queste procedure, l'utente master deve prima connettersi al `rdsadmin` database.

Argomenti

rdsadmin.create_database

Crea un database.

Sintassi

```
db2 "call rdsadmin.create_database('database_name')"
```

Parametri

> **Note**
>
> Questa procedura memorizzata non convalida la combinazione dei parametri richiesti. Quando si chiamardsadmin.get_task_status, la funzione definita dall'utente potrebbe restituire un errore a causa di una combinazione didatabase_codeset,database_territory,

e `database_collation` tale errore non è valido. Per ulteriori informazioni, consulta Scelta della tabella codici, del territorio e delle regole di confronto per il database nella documentazione. IBM Db2

Il parametro seguente è obbligatorio:

`database_name`

Il nome del database da creare. Il tipo di dati è`varchar`.

I parametri seguenti sono facoltativi:

`database_page_size`

La dimensione predefinita della pagina del database. Valori validi: `4096`, `8192`, `16384`, `32768`. Il tipo di dati è`integer`. Il valore predefinito è `8192`.

> **Important**
>
> Amazon RDS supporta l'atomicità di scrittura per pagine da 4 KiB, 8 KiB e 16 KiB. Al contrario, le pagine da 32 KB rischiano scritture strappate o dati parziali che vengano scritti sulla scrivania. Se utilizzi pagine da 32 KiB, ti consigliamo di abilitare il point-in-time ripristino e i backup automatici. Altrimenti, corri il rischio di non riuscire a recuperare le pagine danneggiate. Per ulteriori informazioni, consulta the section called “Introduzione ai backup” e the section called “oint-in-time Ripristino P”.

`database_code_set`

Il set di codici per il database. Il tipo di dati è`varchar`. Il valore predefinito è `UTF-8`.

`database_territory`

Il codice del paese a due lettere per il database. Il tipo di dati è`varchar`. Il valore predefinito è `US`.

`database_collation`

La sequenza di confronto che determina il modo in cui le stringhe di caratteri memorizzate nel database vengono ordinate e confrontate. Il tipo di dati è. `varchar`

Valori validi:

- `COMPATIBILITY`— Una sequenza di confronto IBM Db2 versione 2.
- `EBCDIC_819_037`— Tabella codici ISO in latino, fascicolazione; CCSID 037 (EBCDIC US English).
- `EBCDIC_819_500`— Tabella codici ISO in latino, raccolta; CCSID 500 (EBCDIC International).
- `EBCDIC_850_037`— Tabella codici latini ASCII, fascicolazione; CCSID 037 (EBCDIC US English).
- `EBCDIC_850_500`— Tabella codici in alfabeto latino ASCII, fascicolazione; CCSID 500 (EBCDIC International).
- `EBCDIC_932_5026`— Tabella codici in giapponese ASCII, fascicolazione; CCSID 037 (EBCDIC in inglese americano).
- `EBCDIC_932_5035`— Tabella codici in giapponese ASCII, fascicolazione; CCSID 500 (EBCDIC International).
- `EBCDIC_1252_037`— Tabella codici in latino di Windows, confronto; CCSID 037 (EBCDIC in inglese americano).
- `EBCDIC_1252_500`— Tabella codici in latino di Windows, confronto; CCSID 500 (EBCDIC International).
- `IDENTITY`— Collazione predefinita. Le stringhe vengono confrontate byte per byte.
- `IDENTITY_16BIT`— Lo schema di codifica di compatibilità per UTF-16: sequenza di confronto a 8 bit (CESU-8). Per ulteriori informazioni, vedere Unicode Technical Report #26 sul sito Web di Unicode Consortium.
- `NLSCHAR`— Da utilizzare solo con la tabella codici thailandese (CP874).
- `SYSTEM`— Se si utilizza`SYSTEM`, il database utilizza automaticamente la sequenza di confronto per e. `database_codeset` `database_territory`

Il valore predefinito è `IDENTITY`.

Inoltre, RDS for Db2 supporta i seguenti gruppi di regole di confronto: e. `language-aware-collation` `locale-sensitive-collation` Per ulteriori informazioni, consulta Scelta di un confronto per un database Unicode nella documentazione. IBM Db2

`database_autoconfigure_str`

La `AUTOCONFIGURE` sintassi del comando, ad esempio,. `'AUTOCONFIGURE APPLY DB'` Il tipo di dati è`varchar`. L'impostazione predefinita è una stringa vuota o nulla.

Per ulteriori informazioni, consulta AUTOCONFIGUREil comando nella IBM Db2 documentazione.

Note per l'utilizzo

Puoi creare un database chiamando `rdsadmin.create_database` se non hai specificato il nome del database quando hai creato l'istanza DB RDS for Db2 utilizzando la console Amazon RDS o il. AWS CLI Per ulteriori informazioni, consulta Creazione di un'istanza database.

Considerazioni speciali:

- Il `CREATE DATABASE` comando inviato all'istanza Db2 utilizza l'`RESTRICTIVE`opzione.
- RDS per Db2 utilizza solo. `AUTOMATIC STORAGE`
- RDS for Db2 utilizza i valori predefiniti per e. `NUMSEGS` `DFT_EXTENT_SZ`
- RDS per Db2 utilizza la crittografia dell'archiviazione e non supporta la crittografia del database.

Per ulteriori informazioni su queste considerazioni, consulta il CREATE DATABASEcomando nella documentazione. IBM Db2

Prima di chiamare`rdsadmin.create_database`, è necessario connettersi al `rdsadmin` database. Nell'esempio seguente, sostituite *`master_username e master_password`* con le informazioni sull'istanza DB RDS for Db2:

```
db2 connect to rdsadmin user master_username using master_password
```

Per informazioni sulla verifica dello stato della creazione di un database, vedere. rdsadmin.get_task_status

Esempi

L'esempio seguente crea un database chiamato TESTJP con una combinazione corretta dei parametri database_code_set, database_territory e database_collation per il Giappone:

```
db2 "call rdsadmin.create_database('TESTJP', 4096, 'IBM-437', 'JP', 'SYSTEM')"
```

rdsadmin.drop_database

Rimuove un database.

Sintassi

```
db2 "call rdsadmin.drop_database('database_name')"
```

Parametri

Il parametro seguente è obbligatorio:

database_name

Il nome del database da eliminare. Il tipo di dati è`varchar`.

Note per l'utilizzo

È possibile eliminare un database chiamando `rdsadmin.drop_database` solo se sono soddisfatte le seguenti condizioni:

- Non hai specificato il nome del database quando hai creato l'istanza DB RDS for Db2 utilizzando la console Amazon RDS o il. AWS CLI Per ulteriori informazioni, consulta Creazione di un'istanza database.
- Hai creato il database chiamando la the section called “rdsadmin.create_database” stored procedure.
- È stato ripristinato il database da un'immagine offline o di cui è stato eseguito il backup richiamando la the section called “rdsadmin.restore_database” stored procedure.

Prima di chiamare`rdsadmin.drop_database`, è necessario connettersi al `rdsadmin` database. Nell'esempio seguente, sostituite *master_username e master_password* con le informazioni sull'istanza DB RDS for Db2:

```
db2 connect to rdsadmin user master_username using master_password
```

Per informazioni sulla verifica dello stato dell'eliminazione di un database, consulta. rdsadmin.get_task_status

Esempi

L'esempio seguente elimina un database chiamato`TESTDB`:

```
db2 "call rdsadmin.drop_database('TESTDB')"
```

Esempi di risposta

Se si passa un nome di database errato, la stored procedure restituisce il seguente esempio di risposta:

```
SQL0438N Application raised error or warning with diagnostic text: "Cannot
drop database. Database with provided name does not exist". SQLSTATE=99993
```

Se hai creato il database utilizzando la console Amazon RDS o AWS CLI, la stored procedure restituisce il seguente esempio di risposta:

```
Return Status = 0
```

Dopo la ricezione`Return Status = 0`, richiama la the section called “rdsadmin.get_task_status” stored procedure. Una risposta simile all'esempio seguente spiega lo stato:

```
1 ERROR DROP_DATABASE RDSDB 2023-10-10-16.33.03.744122 2023-10-10-16.33.30.143797 -
 2023-10-10-16.33.30.098857 Task execution has started.
2023-10-10-16.33.30.143797 Caught exception during executing task id 1, Aborting task.
Reason Dropping database created via rds CreateDBInstance api is not allowed.
Only database created using rdsadmin.create_database can be dropped
```

rdsadmin.update_db_param

Aggiorna i parametri del database.

Sintassi

```
db2 "call rdsadmin.update_db_param(
    'database_name',
    'parameter_to_modify',
    'changed_value')"
```

Parametri

I parametri seguenti sono obbligatori:

database_name

Il nome del database per cui eseguire l'attività. Il tipo di dati è`varchar`.

parametro_da_modificare

Il nome del parametro da modificare. Il tipo di dati è`varchar`. Per ulteriori informazioni, consulta RDS per i parametri Db2.

changed_value

Il valore in cui modificare il valore del parametro. Il tipo di dati è`varchar`.

Note per l'utilizzo

Per informazioni sulla verifica dello stato dell'aggiornamento dei parametri del database, vedererdsadmin.get_task_status.

Esempi

L'esempio seguente aggiorna il `archretrydelay` parametro `100` per un database chiamato`TESTDB`:

```
db2 "call rdsadmin.update_db_param(
    'TESTDB',
    'archretrydelay',
    '100')"
```

L'esempio seguente posticipa la convalida degli oggetti creati su un database chiamato `TESTDB` per evitare il controllo delle dipendenze:

```
db2 "call rdsadmin.update_db_param(
    'TESTDB',
    'auto_reval',
    'deferred_force')"
```

rdsadmin.set_configuration

Configura impostazioni specifiche per il database.

Sintassi

```
db2 "call rdsadmin.set_configuration(
```

```
    'name',
    'value)"
```

Parametri

I parametri seguenti sono obbligatori:

name

Il nome dell'impostazione di configurazione. Il tipo di dati è`varchar`.

value

Il valore per l'impostazione di configurazione. Il tipo di dati è`varchar`.

Note per l'utilizzo

La tabella seguente mostra le impostazioni di configurazione con cui è possibile controllare`rdsadmin.set_configuration`.

Nome	Descrizione
`RESTORE_DATABASE_NUM_BUFFERS`	Il numero di buffer da creare durante un'operazione di ripristino. Questo valore deve essere inferiore alla dimensione totale della memoria della classe di istanze DB. Se questa impostazione non è configurata, Db2 determina il valore da utilizzare durante l'operazione di ripristino. Per ulteriori informazioni, consulta la documentazione relativa ad IBM Db2.
`RESTORE_DATABASE_PARALLELISM`	Il numero di manipolatori di buffer da creare durante un'operazione di ripristino. Questo valore deve essere inferiore al doppio del numero di vCPU per l'istanza DB. Se questa impostazione non è configurata, Db2 determina il valore da utilizzare durante l'operazione di ripristino. Per ulteriori informazioni, consulta la documentazione relativa ad IBM Db2.

Esempi

L'esempio seguente imposta la `RESTORE_DATABASE_PARALLELISM` configurazione su. 8

```
db2 "call rdsadmin.set_configuration(
    'RESTORE_DATABASE_PARALLELISM',
    '8')"
```

L'esempio seguente imposta la RESTORE_DATABASE_NUM_BUFFERS configurazione su150.

```
db2 "call rdsadmin.set_configuration(
    'RESTORE_DATABASE_NUM_BUFFERS',
    '150')"
```

rdsadmin.show_configuration

Restituisce le impostazioni correnti che è possibile impostare utilizzando la stored procedurerdsadmin.set_configuration.

Sintassi

```
db2 "call rdsadmin.show_configuration(
    'name')"
```

Parametri

Il parametro seguente è facoltativo:

name

Il nome dell'impostazione di configurazione su cui restituire informazioni. Il tipo di dati èvarchar.

I seguenti nomi di configurazione sono validi:

- RESTORE_DATABASE_NUM_BUFFERS — Il numero di buffer da creare durante un'operazione di ripristino.
- RESTORE_DATABASE_PARALLELISM — Il numero di manipolatori di buffer da creare durante un'operazione di ripristino.

Note per l'utilizzo

Se non si specifica il nome di un'impostazione di configurazione, rdsadmin.show_configuration restituisce informazioni per tutte le impostazioni di configurazione che è possibile impostare utilizzando la stored procedure. rdsadmin.set_configuration

Esempi

L'esempio seguente restituisce informazioni sulla `RESTORE_DATABASE_PARALLELISM` configurazione corrente.

```
db2 "call rdsadmin.show_configuration(
    'RESTORE_DATABASE_PARALLELISM')"
```

rdsadmin.restore_database

Ripristina un database.

Sintassi

```
db2 "call rdsadmin.restore_database(
    ?,
    'database_name',
    's3_bucket_name',
    's3_prefix',
    restore_timestamp,
    'backup_type')"
```

Parametri

È richiesto il seguente parametro di output:

?

Un indicatore di parametro che emette un messaggio di errore. Questo parametro accetta solo. ?

Sono richiesti i seguenti parametri di input:

database_name

Il nome del database da ripristinare. Questo nome deve corrispondere al nome del database nell'immagine di backup. Il tipo di dati è`varchar`.

s3_bucket_name

Il nome del bucket Amazon S3 in cui risiede il backup. Il tipo di dati è. `varchar`

s3_prefix

Il prefisso da usare per la corrispondenza dei file durante il download. Il tipo di dati è`varchar`.

Se questo parametro è vuoto, verranno scaricati tutti i file nel bucket Amazon S3. Di seguito è riportato un esempio di prefisso:

```
backupfolder/SAMPLE.0.rdsdb.DBPART000.20230615010101
```

restore_timestamp

Il timestamp dell'immagine di backup del database. Il tipo di dati è. `varchar`

Il timestamp è incluso nel nome del file di backup. Ad esempio, `20230615010101` è il timestamp del nome del file. `SAMPLE.0.rdsdb.DBPART000.20230615010101.001`

tipo_backup

Il tipo di backup. Il tipo di dati è`varchar`. Valori validi: `OFFLINE`, `ONLINE`.

Utilizzalo `ONLINE` per migrazioni con tempi di inattività prossimi allo zero. Per ulteriori informazioni, consulta Migrazione con tempi di inattività quasi nulli per Linux database Db2 basati.

Note per l'utilizzo

Puoi ripristinare un database chiamando `rdsadmin.restore_database` se non hai specificato il nome del database quando hai creato l'istanza DB RDS for Db2 utilizzando la console Amazon RDS o il. AWS CLI Per ulteriori informazioni, consulta Creazione di un'istanza database.

Prima di ripristinare un database, devi fornire uno spazio di archiviazione per l'istanza DB RDS for Db2 uguale o superiore alla somma delle dimensioni del backup e del database Db2 originale su disco. Quando ripristini il backup, Amazon RDS estrae il file di backup sulla tua istanza DB RDS for Db2.

Ogni file di backup deve avere una dimensione pari o inferiore a 5 TB. Se un file di backup supera i 5 TB, devi dividerlo in file più piccoli.

Per ripristinare tutti i file utilizzando la procedura `rdsadmin.restore_database` memorizzata, non includere il suffisso numerico del file dopo il timestamp nei nomi dei file. Ad esempio, il prefisso *backupfolder/SAMPLE.0.rdsdb.DBPART000.20230615010101s3_ripristina* i seguenti file:

```
SAMPLE.0.rdsdb.DBPART000.20230615010101.001
SAMPLE.0.rdsdb.DBPART000.20230615010101.002
```

```
SAMPLE.0.rdsdb.DBPART000.20230615010101.003
SAMPLE.0.rdsdb.DBPART000.20230615010101.004
SAMPLE.0.rdsdb.DBPART000.20230615010101.005
```

Per migliorare le prestazioni delle operazioni di ripristino del database, è possibile configurare il numero di buffer e manipolatori di buffer da utilizzare con RDS. Per verificare la configurazione corrente, utilizzare. the section called “rdsadmin.show_configuration” Per modificare la configurazione, utilizzarethe section called “rdsadmin.set_configuration”.

Per informazioni sulla verifica dello stato del ripristino del database, vedererdsadmin.get_task_status.

Per portare il database online e applicare registri delle transazioni aggiuntivi dopo il ripristino del database, vedere. rdsadmin.rollforward_database

Esempi

L'esempio seguente ripristina un backup offline con uno o più file con il prefisso s3_prefix: backupfolder/SAMPLE.0.rdsdb.DBPART000.20230615010101

```
db2 "call rdsadmin.restore_database(
    ?,
    'SAMPLE',
    'myS3bucket',
    'backupfolder/SAMPLE.0.rdsdb.DBPART000.20230615010101',
    20230615010101,
    'OFFLINE')"
```

rdsadmin.rollforward_database

Porta il database online e applica registri delle transazioni aggiuntivi dopo aver ripristinato un database tramite chiamata. rdsadmin.restore_database

Sintassi

```
db2 "call rdsadmin.rollforward_database(
    ?,
    'database_name',
    's3_bucket_name',
    s3_prefix,
    'rollfoward_to_option',
    'complete_rollforward')"
```

Parametri

È richiesto il seguente parametro di output:

?

Un indicatore di parametro che emette un messaggio di errore. Questo parametro accetta solo. ?

Sono richiesti i seguenti parametri di input:

database_name

Il nome del database su cui eseguire l'operazione. Il tipo di dati è`varchar`.

s3_bucket_name

Il nome del bucket Amazon S3 in cui risiede il backup. Il tipo di dati è. `varchar`

s3_prefix

Il prefisso da usare per la corrispondenza dei file durante il download. Il tipo di dati è`varchar`.

Se questo parametro è vuoto, verranno scaricati tutti i file nel bucket S3. L'esempio seguente è un esempio di prefisso:

```
backupfolder/SAMPLE.0.rdsdb.DBPART000.20230615010101
```

I seguenti parametri di input sono opzionali:

rollforward_to_option

Il punto verso il quale vuoi rotolare in avanti. Il tipo di dati è`varchar`. Valori validi: `END_OF_LOGS`, `END_OF_BACKUP`. Il valore predefinito è `END OF LOGS`.

complete_rollforward

Speciifica se completare il processo di roll-forward. Il tipo di dati è. `varchar` Il valore predefinito è `TRUE`.

Se`TRUE`, dopo il completamento, il database è online e accessibile. Se`FALSE`, allora il database rimane in uno `ROLL-FORWARD PENDING` stato.

Note per l'utilizzo

Dopo la chiamatardsadmin.restore_database, è necessario chiamare `rollforward_database` per applicare i log di archivio da un bucket S3. È inoltre possibile utilizzare questa procedura memorizzata per ripristinare i registri delle transazioni aggiuntivi dopo la chiamata. `rdsadmin.restore_database`

Se lo `complete_rollforward` impostiFALSE, il database è in uno `ROLL-FORWARD PENDING` stato e non in linea. Per portare il database online, è necessario chiamarerdsadmin.complete_rollforward.

Per informazioni sulla verifica dello stato del rollforward del database, vedererdsadmin.get_task_status.

Esempi

L'esempio seguente passa a un backup online del database con i registri delle transazioni e quindi porta il database online:

```
db2 "call rdsadmin.rollforward_database(
    ?,
    null,
    null,
    'END_OF_LOGS',
    'TRUE')"
```

L'esempio seguente passa a un backup online del database senza registri delle transazioni, quindi porta il database online:

```
db2 "call rdsadmin.rollforward_database(
    ?,
    'TESTDB',
    'S3Bucket',
    'logsfolder/,
    'END_OF_BACKUP',
    'TRUE')"
```

L'esempio seguente passa a un backup online del database con i registri delle transazioni e quindi non porta il database online:

```
db2 "call rdsadmin.rollforward_database(
```

```
    ?,
    'TESTDB',
    null,
    'onlinebackup/TESTDB',
    'END_OF_LOGS',
    'FALSE')"
```

L'esempio seguente passa a un backup online del database con registri delle transazioni aggiuntivi e quindi non porta il database online:

```
db2 "call rdsadmin.rollforward_database(
    ?,
    'TESTDB',
    'S3Bucket',
    'logsfolder/S0000155.LOG',
    'END_OF_LOGS',
    'FALSE')"
```

rdsadmin.complete_rollforward

Porta il database online da uno `ROLL-FORWARD PENDING` stato.

Sintassi

```
db2 "call rdsadmin.complete_rollforward(
    ?,
    'database_name')"
```

Parametri

È richiesto il seguente parametro di output:

?

Un indicatore di parametro che emette un messaggio di errore. Questo parametro accetta solo. ?

È richiesto il seguente parametro di input:

database_name

Il nome del database che desideri portare online. Il tipo di dati è`varchar`.

Note per l'utilizzo

Se hai chiamato rdsadmin.rollforward_database con `complete_rollforward` set to`FALSE`, il database è in uno `ROLL-FORWARD PENDING` stato e non in linea. Per completare il processo di roll-forward e portare il database online, chiama. `rdsadmin.complete_rollforward`

Per informazioni sulla verifica dello stato del completamento della procedura di roll-forward, vedere. rdsadmin.get_task_status

Esempi

L'esempio seguente porta il TESTDB database online:

```
db2 "call rdsadmin.complete_rollfoward(
    ?,
    'TESTDB')"
```

rdsadmin.db2pd_command

Raccoglie informazioni su un database RDS for Db2.

Sintassi

```
db2 "call rdsadmin.db2pd_command('db2pd_cmd')"
```

Parametri

È richiesto il seguente parametro di input:

db2pd_cmd

Il nome del db2pd comando che si desidera eseguire. Il tipo di dati è`varchar`.

Il parametro deve iniziare con un trattino. Per un elenco di parametri, consulta db2pd - Monitora e risolvi i problemi del comando del database Db2 nella documentazione IBM.

I seguenti parametri non possono essere utilizzati:

- `-rep` | `-repeat`
- `-fil` | `-file`

- `-db| -data | -database <dbname>` senza opzioni secondarie, ad esempio o `-apinfo -logs`
- `-inst | -instance`

Note per l'utilizzo

Questa procedura memorizzata raccoglie informazioni che possono contribuire al monitoraggio e alla risoluzione dei problemi dei database RDS per Db2.

La stored procedure utilizza l'IBM`db2pd`utilità per eseguire vari comandi. L'`db2pd`utilità richiede `SYSADM` l'autorizzazione, che l'utente master RDS for Db2 non dispone. Tuttavia, con la stored procedure di Amazon RDS, l'utente principale è in grado di utilizzare l'utilità per eseguire vari comandi. Per ulteriori informazioni sull'utilità, consulta db2pd - Monitor and troubleshoot Db2 database command nella documentazione IBM.

L'output è limitato a un massimo di 2 MB.

Per informazioni sulla verifica dello stato della raccolta di informazioni sul database, vedererdsadmin.get_task_status.

Esempi

L'esempio seguente restituisce l'uptime di un'istanza DB RDS for Db2:

```
db2 "call rdsadmin.db2pd_command('-')"
```

L'esempio seguente restituisce l'uptime di un database chiamato: `TESTDB`

```
db2 "call rdsadmin.db2pd_command('-db TESTDB -')"
```

L'esempio seguente restituisce l'utilizzo della memoria di un'istanza DB RDS for Db2:

```
db2 "call rdsadmin.db2pd_command('-dbptnmem')"
```

L'esempio seguente restituisce i set di memoria di un'istanza RDS for Db2 DB e di un database chiamato: `TESTDB`

```
db2 "call rdsadmin.db2pd_command('-inst -db TESTDB -memsets')"
```

rdsadmin.force_application

Forza la disattivazione delle applicazioni da un database RDS for Db2.

Sintassi

```
db2 "call rdsadmin.force_application(
    ?,
    'applications')"
```

Parametri

È richiesto il seguente parametro di output:

?

Un indicatore di parametro che emette un messaggio di errore. Questo parametro accetta solo. ?

È richiesto il seguente parametro di input:

applications

Le applicazioni che si desidera forzare la disattivazione da un database RDS for Db2. Il tipo di dati è. `varchar` Valori validi: `ALL` o *application_handle*.

Separa i nomi di più applicazioni con virgole. *Esempio: 'application_handle_1, application_handle_2'.*

Note per l'utilizzo

Questa procedura memorizzata espelle tutte le applicazioni da un database in modo da poter eseguire la manutenzione.

La stored procedure utilizza il IBM `FORCE APPLICATION` comando. Il `FORCE APPLICATION` comando richiede `SYSADMSYSMAINT`, o `SYSCTRL` autorizza, che l'utente master RDS for Db2 non dispone. Tuttavia, con la stored procedure di Amazon RDS, l'utente principale è in grado di utilizzare il comando. Per ulteriori informazioni, consulta il comando FORCE APPLICATION nella documentazione IBM.

Per informazioni sulla verifica dello stato dell'esclusione forzata delle applicazioni da un database, consultardsadmin.get_task_status.

Esempi

L'esempio seguente forza la disattivazione di tutte le applicazioni da un database RDS for Db2:

```
db2 "call rdsadmin.force_application(
    ?,
    'ALL')"
```

L'esempio seguente impone la 1192 disattivazione e la disattivazione degli handle 9991 delle applicazioni da un database RDS for Db2: 8891

```
db2 "call rdsadmin.force_application(
    ?,
    '9991, 8891, 1192')"
```

rdsadmin.set_archive_log_retention

Configura la quantità di tempo (in ore) per conservare i file di registro di archivio per il database RDS for Db2 specificato.

Sintassi

```
db2 "call rdsadmin.set_archive_log_retention(
    ?,
    'database_name',
    'archive_log_retention_hours')"
```

Parametri

È richiesto il seguente parametro di output:

?

Un indicatore di parametro che emette un messaggio di errore. Questo parametro accetta solo. ?

Sono richiesti i seguenti parametri di input:

database_name

Il nome del database per cui configurare la conservazione dei registri di archivio. Il tipo di dati è`varchar`.

archive_log_retention_hours

Il numero di ore per conservare i file di registro dell'archivio. Il tipo di dati è`smallint`. L'impostazione predefinita è `0` e il massimo è `168` (7 giorni).

Se il valore è`0`, Amazon RDS non conserva i file di log di archivio.

Note per l'utilizzo

Puoi visualizzare l'impostazione corrente di conservazione dei log di archivio the section called “rdsadmin.show_archive_log_retention” chiamando.

Non è possibile configurare l'impostazione di conservazione dei log di archivio nel `rdsadmin` database.

Esempi

L'esempio seguente imposta il tempo di conservazione dei log di archivio per un database chiamato `TESTDB` a 24 ore.

```
db2 "call rdsadmin.set_archive_log_retention(
    ?,
    'TESTDB',
    '24')"
```

L'esempio seguente disattiva la conservazione dei registri di archivio per un database chiamato`TESTDB`.

```
db2 "call rdsadmin.set_archive_log_retention(
    ?,
    'TESTDB',
    '0')"
```

rdsadmin.show_archive_log_retention

Restituisce l'impostazione corrente di conservazione dei registri di archivio per il database specificato.

Sintassi

```
db2 "call rdsadmin.show_archive_log_retention(
```

```
    ?,
    'database_name')"
```

Parametri

È richiesto il seguente parametro di output:

?

Un indicatore di parametro che emette un messaggio di errore. Questo parametro accetta solo. ?

È richiesto il seguente parametro di input:

database_name

Il nome del database per cui mostrare l'impostazione di conservazione dei log di archivio. Il tipo di dati è`varchar`.

Esempi

L'esempio seguente mostra l'impostazione di conservazione dei registri di archivio per un database chiamato`TESTDB`.

```
db2 "call rdsadmin.show_archive_log_retention(
    ?
    'TESTDB')"
```

Gestione delle tablespace

Le seguenti stored procedure gestiscono i tablespace per i database Amazon RDS for Db2. Per eseguire queste procedure, l'utente master deve prima connettersi al database. `rdsadmin`

Argomenti

rdsadmin.create_tablespace

Crea un tablespace.

Sintassi

```
db2 "call rdsadmin.create_tablespace(
    'database_name',
    'tablespace_name',
    'buffer_pool_name',
    tablespace_page_size,
    tablespace_initial_size,
    tablespace_increase_size,
    'tablespace_type')"
```

Parametri

I parametri seguenti sono obbligatori:

`database_name`

Il nome del database in cui creare il tablespace. Il tipo di dati è. `varchar`

`tablespace_name`

Il nome del tablespace da creare. Il tipo di dati è. `varchar`

Il nome del tablespace presenta le seguenti restrizioni:

- Non può essere uguale al nome di un tablespace esistente in questo database.

- Può contenere solo i caratteri. `_$#@a-zA-Z0-9`
- Non può iniziare con _ o$.
- Non può iniziare con`SYS`.

I parametri seguenti sono facoltativi:

`buffer_pool_name`

Il nome del pool di buffer da assegnare al tablespace. Il tipo di dati è. `varchar` L'impostazione predefinita è una stringa vuota.

> **Important**
>
> È necessario disporre già di un pool di buffer della stessa dimensione di pagina da associare al tablespace.

`tablespace_page_size`

La dimensione della pagina del tablespace in byte. Il tipo di dati è. `integer` Valori validi: `4096`, `8192`, `16384`, `32768`. L'impostazione predefinita è la dimensione della pagina utilizzata quando è stato creato il database tramite chiamatardsadmin.create_database.

> **Important**
>
> Amazon RDS supporta l'atomicità di scrittura per pagine da 4 KiB, 8 KiB e 16 KiB. Al contrario, le pagine da 32 KB rischiano scritture strappate o dati parziali che vengano scritti sulla scrivania. Se utilizzi pagine da 32 KiB, ti consigliamo di abilitare il point-in-time ripristino e i backup automatici. Altrimenti, corri il rischio di non riuscire a recuperare le pagine danneggiate. Per ulteriori informazioni, consulta the section called “Introduzione ai backup” e the section called “oint-in-time Ripristino P”.

`tablespace_initial_size`

La dimensione iniziale del tablespace in kilobyte (KB). Il tipo di dati è. `integer` Valori validi: `48` o superiori. Il valore predefinito è null.

Se non imposti un valore, Db2 imposta un valore appropriato per te.

> **Note**
>
> Questo parametro non è applicabile alle tablespace temporanee perché il sistema gestisce le tablespace temporanee.

tablespace_incremente_size

La percentuale di cui aumentare la tablespace quando è piena. Il tipo di dati è. `integer` Valori validi: 1 —`100`. Il valore predefinito è null.

Se non imposti un valore, Db2 imposta un valore appropriato per te.

> **Note**
>
> Questo parametro non è applicabile alle tablespace temporanee perché il sistema gestisce le tablespace temporanee.

tablespace_type

Il tipo di tablespace. Il tipo di dati è. `char` Valori validi: U (per i dati utente) o T (per i dati temporanei). Il valore predefinito è U.

Note per l'utilizzo

RDS for Db2 crea sempre un database di dati di grandi dimensioni.

Per informazioni sulla verifica dello stato della creazione di un tablespace, consulta. rdsadmin.get_task_status

Esempi

L'esempio seguente crea un tablespace chiamato SP8 e assegna un pool di buffer chiamato per un database chiamato. BP8 TESTDB La tablespace ha una dimensione iniziale della pagina tablespace di 4.096 byte, una tablespace iniziale di 1.000 KB e un aumento delle dimensioni della tabella impostato al 50%.

```
db2 "call rdsadmin.create_tablespace(
```

```
    'TESTDB',
    'SP8',
    'BP8',
    4096,
    1000,
    50)"
```

L'esempio seguente crea un tablespace temporaneo chiamato. SP8 Assegna un pool di buffer chiamato BP8 della dimensione di 8 KB per un database chiamato. TESTDB

```
db2 "call rdsadmin.create_tablespace(
    'TESTDB',
    'SP8',
    'BP8',
    8192,
    NULL,
    NULL,
    'T')"
```

rdsadmin.alter_tablespace

Modifica un tablespace.

Sintassi

```
db2 "call rdsadmin.alter_tablespace(
    'database_name',
    'tablespace_name',
    'buffer_pool_name',
    tablespace_increase_size,
    'max_size',
    'reduce_max',
    'reduce_stop',
    'reduce_value',
    'lower_high_water',
    'lower_high_water_stop',
    'switch_online')"
```

Parametri

I parametri seguenti sono obbligatori:

database_name

Il nome del database che utilizza il tablespace. Il tipo di dati è. varchar

tablespace_name

Il nome del tablespace da modificare. Il tipo di dati è. varchar

I parametri seguenti sono facoltativi:

buffer_pool_name

Il nome del pool di buffer da assegnare al tablespace. Il tipo di dati è. varchar L'impostazione predefinita è una stringa vuota.

> **Important**
>
> È necessario disporre già di un pool di buffer della stessa dimensione di pagina da associare al tablespace.

tablespace_incremente_size

La percentuale di cui aumentare la tablespace quando è piena. Il tipo di dati è. integer Valori validi: 1 —100. Il valore predefinito è 0.

dimensione_massima

La dimensione massima per il tablespace. Il tipo di dati è. varchar Valori validi: *integer* K | M |G, orNONE. Il valore predefinito è NONE.

reduce_max

Specifica se ridurre la soglia massima al limite massimo. Il tipo di dati èchar. Il valore predefinito è N.

reduce_stop

Specifica se interrompere un comando o un comando precedente. reduce_max reduce_value Il tipo di dati è. char Il valore predefinito è N.

reduce_value

Il numero o la percentuale di cui ridurre il limite massimo consentito dalla tablespace. Il tipo di dati è. varchar Valori validi: *intero* K | M | G o 1 —100. Il valore predefinito è N.

lower_high_water

Specifica se eseguire il comando. `ALTER TABLESPACE LOWER HIGH WATER MARK` Il tipo di dati è`char`. Il valore predefinito è `N`.

lower_high_water_stop

Specifica se eseguire il comando. `ALTER TABLESPACE LOWER HIGH WATER MARK STOP` Il tipo di dati è`char`. Il valore predefinito è `N`.

switch_online

Speciifica se eseguire il comando. `ALTER TABLESPACE SWITCH ONLINE` Il tipo di dati è`char`. Il valore predefinito è `N`.

Note per l'utilizzo

I parametri opzionali`reduce_max`,`reduce_stop`,`reduce_value`, `lower_high_water`lower_high_water_stop, e si `switch_online` escludono a vicenda. Non è possibile combinarli con nessun altro parametro opzionale, ad esempio `buffer_pool_name` nel `rdsadmin.alter_tablespace` comando. Se si combinano questi parametri con qualsiasi altro parametro opzionale del `rdsadmin.alter_tablespace` comando, all'esecuzione`rdsadmin.get_task_status`, Db2 restituirà un errore simile al seguente:

```
DB21034E  The command was processed as an SQL statement because it was not a valid
 Command Line Processor command.  During SQL processing it returned:
SQL1763N  Invalid ALTER TABLESPACE statement for table space "TBSP_TEST" due to reason
 "12"
```

Per informazioni sulla verifica dello stato della modifica di un tablespace, vedere. rdsadmin.get_task_status

Esempi

L'esempio seguente modifica un tablespace chiamato SP8 e assegna un buffer pool chiamato a un database chiamato BP8 per abbassare il limite massimo. TESTDB

```
db2 "call rdsadmin.alter_tablespace(
    'TESTDB',
    'SP8',
    'BP8',
    NULL,
```

```
    NULL,
    'Y')"
```

L'esempio seguente esegue il `REDUCE MAX` comando su un tablespace chiamato nel database. `TBSP_TEST` `TESTDB`

```
db2 "call rdsadmin.alter_tablespace(
    'TESTDB',
    'TBSP_TEST',
    NULL,
    NULL,
    NULL,
    'Y')"
```

L'esempio seguente esegue il `REDUCE STOP` comando su una tablespace chiamata `TBSP_TEST` nel database. `TESTDB`

```
db2 "call rdsadmin.alter_tablespace(
    'TESTDB',
    'TBSP_TEST',
    NULL,
    NULL,
    NULL,
    NULL,
    'Y')"
```

rdsadmin.rename_tablespace

Rinomina un tablespace.

Sintassi

```
db2 "call rdsadmin.rename_tablespace(
    ?,
    'database_name',
    'source_tablespace_name',
    'target_tablespace_name')"
```

Parametri

I parametri seguenti sono obbligatori:

?

Un indicatore di parametro che emette un messaggio di errore. Questo parametro accetta solo?.

database_name

Il nome del database a cui appartiene il tablespace. Il tipo di dati è. `varchar`

source_tablespace_name

Il nome del tablespace da rinominare. Il tipo di dati è. `varchar`

target_tablespace_name

Il nuovo nome del tablespace. Il tipo di dati è. `varchar`

Il nuovo nome presenta le seguenti restrizioni:

- Non può essere uguale al nome di un tablespace esistente.
- Può contenere solo i caratteri. `_$#@a-zA-Z0-9`
- Non può iniziare con _ o$.
- Non può iniziare conSYS.

Note per l'utilizzo

Per informazioni sulla verifica dello stato della ridenominazione di una tablespace, consulta. rdsadmin.get_task_status

Non è possibile rinominare le tablespace che appartengono al database. `rdsadmin`

Esempi

L'esempio seguente rinomina un tablespace chiamato a in un database chiamato. `SP8` `SP9` `TESTDB`

```
db2 "call rdsadmin.rename_tablespace(
    ?,
    'TESTDB',
    'SP8'.
    'SP9')"
```

rdsadmin.drop_tablespace

Elimina un tablespace.

Sintassi

```
db2 "call rdsadmin.drop_tablespace(
    'database_name',
    'tablespace_name')"
```

Parametri

I parametri seguenti sono obbligatori:

database_name

Il nome del database a cui appartiene il tablespace. Il tipo di dati è. varchar

tablespace_name

Il nome del tablespace da eliminare. Il tipo di dati è. varchar

Note per l'utilizzo

Per informazioni sulla verifica dello stato dell'eliminazione di una tablespace, consulta. rdsadmin.get_task_status

Esempi

L'esempio seguente elimina un tablespace chiamato da un database chiamatoSP8. TESTDB

```
db2 "call rdsadmin.drop_tablespace(
    'TESTDB',
    'SP8')"
```

Gestione delle politiche di controllo

Le seguenti stored procedure gestiscono le politiche di controllo per i database Amazon RDS for Db2 che utilizzano la registrazione di controllo. Per ulteriori informazioni, consulta the section called “Registrazione di controllo Db2”. Per eseguire queste procedure, l'utente master deve prima connettersi al database. `rdsadmin`

Argomenti

- rdsadmin.configure_db_audit
- rdsadmin.disable_db_audit

rdsadmin.configure_db_audit

Configura la politica di controllo per il database RDS for Db2 specificato da db_name. Se la policy che stai configurando non esiste, la chiamata a questa stored procedure la crea. Se questa policy esiste, la chiamata a questa stored procedure la modifica con i valori dei parametri forniti dall'utente.

Sintassi

```
db2 "call rdsadmin.configure_db_audit(
    'db_name',
    'category',
    'category_setting',
    '?')"
```

Parametri

I seguenti parametri sono obbligatori.

db_name

Il nome DB del database RDS for Db2 per cui configurare la politica di controllo. Il tipo di dati è. `varchar`

categoria

Il nome della categoria per cui configurare questa politica di controllo. Il tipo di dati è`varchar`. Di seguito sono riportati i valori validi per questo parametro:

- `ALL`— Con`ALL`, Amazon RDS non include le `CONTEXT``EXECUTE`, o `ERROR` le categorie.

- AUDIT
- CHECKING
- CONTEXT
- ERROR
- EXECUTE— Puoi configurare questa categoria con o senza dati. Con data significa anche registrare i valori dei dati di input forniti per qualsiasi variabile host e indicatore di parametro. L'impostazione predefinita è senza dati. Per ulteriori informazioni, vedete la descrizione del parametro *category_setting* e il. the section called “Esempi”
- OBJMAINT
- SECMAINT
- SYSADMIN
- VALIDATE

Per ulteriori informazioni su queste categorie, consulta la documentazione. IBM Db2

categoria_impostazione

L'impostazione per la categoria di controllo specificata. Il tipo di dati è`varchar`.

La tabella seguente mostra i valori di impostazione delle categorie validi per ogni categoria.

Categoria	Impostazioni valide per le categorie
ALL	BOTH|FAILURE|SUCCESS|NONE
AUDIT	
CHECKING	
CONTEXT	
OBJMAINT	
SECMAINT	
SYSADMIN	
VALIDATE	

Categoria	Impostazioni valide per le categorie
ERROR	AUDIT|NORMAL . L'impostazione predefinita èNORMAL.
EXECUTE	BOTH,WITH|BOTH,WITHOUT|FAILURE,WITH|FAILURE,WITHOUT|SUCCESS,WITH|SUCCESS,WITHOUT|NONE

Note per l'utilizzo

Prima di chiamarerdsadmin.configure_db_audit, assicurati che l'istanza DB RDS for Db2 con il database per cui stai configurando la politica di controllo sia associata a un gruppo di opzioni che disponga dell'opzione. DB2_AUDIT Per ulteriori informazioni, consulta the section called "Configurazione della registrazione di controllo Db2".

Dopo aver configurato la politica di controllo, puoi verificare lo stato della configurazione di controllo per il database seguendo i passaggi riportati di seguito. Verificare la configurazione dell'audit

La ALL specificazione del category parametro non include le ERROR categorie CONTEXTEXECUTE, o. Per aggiungere queste categorie alla tua politica di controllo, chiama rdsadmin.configure_db_audit separatamente ogni categoria che desideri aggiungere. Per ulteriori informazioni, consulta the section called "Esempi".

Esempi

I seguenti esempi creano o modificano la politica di controllo per un database denominatoTESTDB. Negli esempi da 1 a 5, se la ERROR categoria non è stata configurata in precedenza, questa categoria è impostata su NORMAL (impostazione predefinita). Per modificare tale impostazione suAUDIT, seguiExample 6: Specifying the ERROR category.

Esempio 1: Specificazione della categoria **ALL**

```
db2 "call rdsadmin.configure_db_audit('TESTDB', 'ALL', 'BOTH', ?)"
```

Nell'esempio, la chiamata configura le VALIDATE categorie AUDITCHECKING,OBJMAINT, SECMAINTSYSADMIN, e nella politica di controllo. Specificare BOTH significa che gli eventi riusciti e quelli non riusciti verranno controllati per ciascuna di queste categorie.

Esempio 2: Specificazione della categoria con i dati **EXECUTE**

```
db2 "call rdsadmin.configure_db_audit('TESTDB', 'EXECUTE', 'SUCCESS,WITH', ?)"
```

Nell'esempio, la chiamata configura la `EXECUTE` categoria nella politica di controllo. Specificare `SUCCESS,WITH` significa che i log di questa categoria includeranno solo gli eventi riusciti e includeranno i valori dei dati di input forniti per le variabili host e i marker dei parametri.

Esempio 3: Specificazione della categoria senza dati **`EXECUTE`**

```
db2 "call rdsadmin.configure_db_audit('TESTDB', 'EXECUTE', 'FAILURE,WITHOUT', ?)"
```

Nell'esempio, la chiamata configura la `EXECUTE` categoria nella politica di controllo. Specificare `FAILURE,WITHOUT` significa che i log di questa categoria includeranno solo gli eventi con esito negativo e non includeranno i valori dei dati di input forniti per le variabili host e i marker dei parametri.

Esempio 4: Specificazione della categoria senza eventi di stato **`EXECUTE`**

```
db2 "call rdsadmin.configure_db_audit('TESTDB', 'EXECUTE', 'NONE', ?)"
```

Nell'esempio, la chiamata configura la `EXECUTE` categoria nella politica di controllo. Specificare `NONE` significa che nessun evento in questa categoria verrà controllato.

Esempio 5: Specificazione della categoria **`OBJMAINT`**

```
db2 "call rdsadmin.configure_db_audit('TESTDB', 'OBJMAINT', 'NONE', ?)"
```

Nell'esempio, la chiamata configura la `OBJMAINT` categoria nella politica di controllo. Specificare `NONE` significa che nessun evento in questa categoria verrà controllato.

Esempio 6: Specificazione della categoria **`ERROR`**

```
db2 "call rdsadmin.configure_db_audit('TESTDB', 'ERROR', 'AUDIT', ?)"
```

Nell'esempio, la chiamata configura la `ERROR` categoria nella politica di controllo. Specificare `AUDIT` significa che tutti gli errori, inclusi gli errori che si verificano all'interno della registrazione di controllo stessa, vengono registrati nei log. Il tipo di errore predefinito è. `NORMAL` Con`NORMAL`, gli errori generati dall'audit vengono ignorati e vengono acquisiti solo gli errori associati all'operazione eseguita. `SQLCODE`

rdsadmin.disable_db_audit

Interrompe la registrazione di controllo per il database RDS for Db2 specificato da *db_name* e rimuove la politica di controllo configurata per tale database.

> **Note**
>
> Questa procedura memorizzata rimuove solo le politiche di controllo configurate tramite chiamata. the section called “rdsadmin.configure_db_audit”

Sintassi

```
db2 "call rdsadmin.disable_db_audit('db_name')"
```

Parametri

I seguenti parametri sono obbligatori.

db_name

Il nome DB del database RDS for Db2 per cui disabilitare la registrazione di controllo. Il tipo di dati è. `varchar`

Note per l'utilizzo

`rdsadmin.disable_db_audit`La chiamata non disabilita la registrazione di controllo per l'istanza DB RDS for Db2. Per disabilitare la registrazione di controllo a livello di istanza DB, rimuovi il gruppo di opzioni dall'istanza DB. Per ulteriori informazioni, consulta Disabilitazione della registrazione di controllo Db2.

Esempi

L'esempio seguente disabilita la registrazione di controllo per un database denominato. `TESTDB`

```
db2 "call rdsadmin.disable_db_audit('TESTDB')"
```

Riferimento alla funzione definita dall'utente RDS per Db2

Questi argomenti descrivono le funzioni definite dall'utente disponibili per le istanze Amazon RDS che eseguono il motore RDS for Db2.

Argomenti

Verifica dello stato di un'attività

È possibile utilizzare la funzione `rdsadmin.get_task_status` definita dall'utente per verificare lo stato delle seguenti attività. L'elenco non è completo.

- Creazione, modifica o eliminazione di un pool di buffer
- Creare, modificare o eliminare una tablespace
- Creare o eliminare un database
- Ripristino di un backup del database da Amazon S3
- Inoltro dei log del database da Amazon S3

rdsadmin.get_task_status

Restituisce lo stato di un'attività.

Sintassi

```
db2 "select task_id, task_type, database_name, lifecycle,
    varchar(bson_to_json(task_input_params), 500) as task_params,
    cast(task_output as varchar(500)) as task_output
    from table(rdsadmin.get_task_status(task_id,'database_name','task_type'))"
```

Parametri

I seguenti parametri sono opzionali. Se non si fornisce alcun parametro, la funzione definita dall'utente restituisce lo stato di tutte le attività per tutti i database. Amazon RDS conserva la cronologia delle attività per 35 giorni.

`task_id`

L'ID dell'attività in esecuzione. Questo ID viene restituito quando si esegue un'attività. Default: `0`.

`database_name`

Il nome del database per il quale viene eseguita l'attività.

`tipo_attività`

Il tipo di attività da interrogare. Valori validi:
`ADD_GROUPSADD_USER,ALTER_BUFFERPOOL,ALTER_TABLESPACE,CHANGE_PASSWORD,COMPLETE_RO`
`CREATE_DATABASECREATE_ROLE,CREATE_TABLESPACE,DROP_BUFFERPOOL,DROP_DATABASE,DROP_`

Esempi

L'esempio seguente visualizza le colonne restituite quando `rdsadmin.get_task_status` viene chiamato.

```
db2 "describe select * from table(rdsadmin.get_task_status())"
```

L'esempio seguente elenca lo stato di tutte le attività.

```
db2 "select task_id, task_type, database_name, lifecycle,
    varchar(bson_to_json(task_input_params), 500) as task_params,
    cast(task_output as varchar(500)) as task_output
    from table(rdsadmin.get_task_status(null,null,null))"
```

L'esempio seguente elenca lo stato di un'attività specifica.

```
db2 "select task_id, task_type, database_name,
    varchar(bson_to_json(task_input_params), 500) as task_params
    from table(rdsadmin.get_task_status(1,null,null))"
```

L'esempio seguente elenca lo stato di un'attività e di un database specifici.

```
db2 "select task_id, task_type, database_name,
    varchar(bson_to_json(task_input_params), 500) as task_params
    from table(rdsadmin.get_task_status(2,'SAMPLE',null))"
```

L'esempio seguente elenca lo stato di tutte le `ADD_GROUPS` attività.

```
db2 "select task_id, task_type, database_name,
    varchar(bson_to_json(task_input_params), 500) as task_params
    from table(rdsadmin.get_task_status(null,null,'add_groups'))"
```

L'esempio seguente elenca lo stato di tutte le attività per un database specifico.

```
db2 "select task_id, task_type, database_name,
    varchar(bson_to_json(task_input_params), 500) as task_params
    from table(rdsadmin.get_task_status(null,'testdb', null))"
```

L'esempio seguente restituisce i valori JSON come colonne.

```
db2 "select varchar(r.task_type,25) as task_type, varchar(r.lifecycle,10) as lifecycle,
 r.created_at, u.* from
    table(rdsadmin.get_task_status(null,null,'restore_db')) as r,
 json_table(r.task_input_params, 'strict $' columns(s3_prefix varchar(500)
    null on empty, s3_bucket_name varchar(500) null on empty) error on error ) as U"
```

Risposta

La funzione `rdsadmin.get_task_status` definita dall'utente restituisce le seguenti colonne:

`TASK_ID`

L'ID dell'attività.

`TASK_TYPE`

Dipende dai parametri di input.

- `ADD_GROUPS`— Aggiunge gruppi.
- `ADD_USER`— Aggiunge un utente.
- `ALTER_BUFFERPOOL`— Modifica un buffer pool.
- `ALTER_TABLESPACE`— Modifica una tablespace.
- `CHANGE_PASSWORD` — Modifica la password di un utente.
- `COMPLETE_ROLLFORWARD`— Completa un'`rdsadmin.rollforward_database`attività e attiva un database.
- `CREATE_BUFFERPOOL`— Crea un pool di buffer.
- `CREATE_DATABASE`— Crea un database.
- `CREATE_ROLE`— Crea un ruolo Db2 per un utente.
- `CREATE_TABLESPACE`— Crea un tablespace.
- `DROP_BUFFERPOOL`— Elimina un buffer pool.
- `DROP_DATABASE`— Elimina un database.
- `DROP_TABLESPACE`— Elimina un tablespace.
- `LIST_USERS`— Elenca tutti gli utenti.
- `REMOVE_GROUPS`— Rimuove i gruppi.
- `REMOVE_USER`— Rimuove un utente.
- `RESTORE_DB`— Ripristina un database completo.

- ROLLFORWARD_DB_LOG— Esegue un'rdsadmin.rollforward_databaseoperazione sui registri del database.
- ROLLFORWARD_STATUS — Restituisce lo stato di un'rdsadmin.rollforward_databaseattività.
- UPDATE_DB_PARAM— Aggiorna i parametri dei dati.

DATABASE_NAME

Il nome del database a cui è associata l'attività.

COMPLETED_WORK_BYTES

Il numero di byte ripristinati dall'operazione.

DURATION_MINS

Il tempo impiegato per completare l'operazione.

LIFECYCLE

Lo stato dell'attività. Stati possibili:

- CREATED— Dopo l'invio di un'attività ad Amazon RDS, Amazon RDS imposta lo stato su. CREATED
- IN_PROGRESS— Dopo l'avvio di un'attività, Amazon RDS imposta lo stato suIN_PROGRESS. La modifica dello stato da a può richiedere fino a 5 minuti. CREATED IN_PROGRESS
- SUCCESS— Al termine di un'attività, Amazon RDS imposta lo stato su. SUCCESS
- ERROR— Se un'attività di ripristino non riesce, Amazon RDS imposta lo stato suERROR. Per ulteriori informazioni sull'errore, consultaTASK_OUPUT.

CREATED_BY

Quello authid che ha creato il comando.

CREATED_AT

La data e l'ora in cui è stata creata l'attività.

LAST_UPDATED_AT

La data e l'ora dell'ultimo aggiornamento dell'attività.

TASK_INPUT_PARAMS

I parametri variano in base al tipo di attività. Tutti i parametri di input sono rappresentati come un oggetto JSON. Ad esempio, le chiavi JSON per l'RESTORE_DBattività sono le seguenti:

- DBNAME
- RESTORE_TIMESTAMP
- S3_BUCKET_NAME
- S3_PREFIX

TASK_OUTPUT

Ulteriori informazioni sull'attività. Se si verifica un errore durante il ripristino nativo, questa colonna include informazioni sull'errore.

Esempi di risposte

Il seguente esempio di risposta mostra che un database chiamato TESTJP è stato creato con successo. Per ulteriori informazioni, vedere la the section called “rdsadmin.create_database” stored procedure.

```
`1 SUCCESS CREATE_DATABASE RDSDB 2023-10-24-18.32.44.962689 2023-10-24-18.34.50.038523
 1 TESTJP { "CODESET" : "IBM-437", "TERRITORY" : "JP", "COLLATION" : "SYSTEM",
 "AUTOCONFIGURE_CMD" : "", "PAGESIZE" : 4096 }
2023-10-24-18.33.30.079048 Task execution has started.

2023-10-24-18.34.50.038523 Task execution has completed successfully`.
```

Il seguente esempio di risposta spiega perché l'eliminazione di un database non è riuscita. Per ulteriori informazioni, vedere la the section called “rdsadmin.drop_database” stored procedure.

```
1 ERROR DROP_DATABASE RDSDB 2023-10-10-16.33.03.744122 2023-10-10-16.33.30.143797 -
 2023-10-10-16.33.30.098857 Task execution has started.
2023-10-10-16.33.30.143797 Caught exception during executing task id 1, Aborting task.
Reason Dropping database created via rds CreateDBInstance api is not allowed.
Only database created using rdsadmin.create_database can be dropped
```

Il seguente esempio di risposta mostra il corretto ripristino di un database. Per ulteriori informazioni, vedere la the section called “rdsadmin.restore_database” stored procedure.

```
1 RESTORE_DB  SAMPLE  SUCCESS

{ "S3_BUCKET_NAME" : "mybucket", "S3_PREFIX" :
 "SAMPLE.0.rdsdb3.DBPART000.20230413183211.001", "RESTORE_TIMESTAMP" :
```

```
 "20230413183211", "BACKUP_TYPE" : "offline" }




2023-11-06-18.31.03.115795 Task execution has started.
2023-11-06-18.31.04.300231 Preparing to download
2023-11-06-18.31.08.368827 Download complete. Starting Restore
2023-11-06-18.33.13.891356 Task Completed Successfully
```

Amazon RDS for MariaDB

Amazon RDS supporta le istanze database che eseguono le seguenti versioni di MariaDB:

- MariaDB 10.11
- MariaDB 10.6
- MariaDB 10.5
- MariaDB 10.4
- MariaDB 10.3 (fine del supporto RDS standard prevista per il 23 ottobre 2023)

Per ulteriori informazioni sul supporto delle versioni secondarie, consulta Versioni di MariaDB in Amazon RDS.

Per creare un'istanza database MariaDB, utilizza gli strumenti o le interfacce di gestione di Amazon RDS. Puoi quindi utilizzare gli strumenti Amazon RDS per eseguire azioni di gestione per l'istanza database. Queste includono le azioni seguenti:

- Riconfigurazione o ridimensionamento dell'istanza database
- Autorizzazione delle connessioni all'istanza database
- Creazione e ripristino da backup o snapshot
- Creazione di istanze secondarie Multi-AZ
- Creazione di repliche di lettura
- Monitoraggio delle prestazioni dell'istanza database

Per archiviare e accedere ai dati nell'istanza database, utilizza le utilità e le applicazioni MariaDB standard.

MariaDB è disponibile in tutte le Regioni AWS. Per ulteriori informazioni su Regioni AWS, consulta Regioni, zone di disponibilità e Local Zones.

Puoi utilizzare i database Amazon RDS for MariaDB allo scopo di creare applicazioni conformi ai requisiti HIPAA. Puoi archiviare informazioni sanitarie, inclusi dati sanitari protetti (PHI), in base a un Contratto di società in affari (BAA) con AWS. Per ulteriori informazioni, consulta Compliance HIPAA. AWS I servizi coperti dal programma di compliance sono stati integralmente valutati da un auditor

di terzi e generano una certificazione, un'attestazione di conformità o un'autorizzazione operativa (ATO). Per ulteriori informazioni, consulta Servizi AWS coperti dal programma di compliance.

Prima di creare un'istanza database, completa i passaggi in Configurazione di Amazon RDS. Quando crei un'istanza database, l'utente principale di RDS ottiene privilegi DBA, con alcune limitazioni. Utilizzare questo account per attività amministrative, ad esempio la creazione di account di database aggiuntivi.

Puoi creare:

- Istanze DB
- Snapshot DB
- Ripristini point-in-time
- Backup automatizzati
- Backup manuali

Puoi utilizzare istanze database che eseguono MariaDB all'interno di un cloud privato virtuale (VPC) basato su Amazon VPC. Inoltre, puoi abilitare varie opzioni per aggiungere altre funzionalità all'istanza database MariaDB. Amazon RDS supporta le implementazioni Multi-AZ per MariaDB come soluzione failover a elevata disponibilità.

> **Important**
>
> Per offrire un'esperienza di servizio gestito, Amazon RDS non fornisce accesso shell alle istanze database. Limita anche l'accesso ad alcune procedure di sistema e tabelle che richiedono privilegi avanzati. Puoi accedere al database utilizzando i client SQL standard come il client mysql. Tuttavia, non è possibile accedere direttamente all'host utilizzando Telnet o Secure Shell (SSH).

Argomenti

- Prestazioni di scrittura migliorate con Scritture ottimizzate per Amazon RDS per MariaDB
- Aggiornamento del motore di database MariaDB
- Importazione di dati in un'istanza database MariaDB
- Uso della replica MariaDB in Amazon RDS
- Opzioni per il motore di database MariaDB
- Parametri per MariaDB
- Migrazione dei dati da uno snapshot DB MySQL a un'istanza database MariaDB
- MariaDB sul riferimento SQL di Amazon RDS
- Fuso orario locale per le istanze database MariaDB
- Problemi e limitazioni note per RDS per MariaDB

Supporto funzionalità MariaDB su Amazon RDS

RDS per MariaDB supporta la maggior parte delle caratteristiche e delle funzionalità di MariaDB. Alcune funzionalità potrebbero avere un supporto o privilegi limitati.

Please change to "Puoi filtrare le nuove funzionalità Amazon RDS alla pagina Quali sono le novità del database?. Per Prodotti, scegli Amazon RDS. Quindi esegui la ricerca utilizzando parole chiave come **`MariaDB 2023`**.

> **Note**
>
> I seguenti elenchi non sono esaustivi.

Argomenti

- Supporto alle funzionalità MariaDB nelle versioni principali di Amazon RDS per MariaDB
- Motori di storage supportati per MariaDB in Amazon RDS
- Precaricamento della cache per MariaDB in Amazon RDS
- Funzionalità MariaDB non supportate da Amazon RDS

Supporto alle funzionalità MariaDB nelle versioni principali di Amazon RDS per MariaDB

Nelle seguenti sezioni sono disponibili informazioni sul supporto alle funzionalità MariaDB nelle versioni principali di Amazon RDS per MariaDB:

Argomenti

Per informazioni sulle versioni secondarie supportate di Amazon RDS for MariaDB, consulta Versioni di MariaDB in Amazon RDS.

Supporto per MariaDB 10.11 in Amazon RDS

Amazon RDS supporta le seguenti nuove funzionalità per le istanze database che eseguono MariaDB versione 10.11 o versioni successive:

- Plugin Password Reuse Check: puoi utilizzare il plug-in MariaDB Password Reuse Check per impedire agli utenti di riutilizzare le password e impostare il periodo di conservazione delle password. Per ulteriori informazioni, consultaPlugin Password Reuse Check (Utilità di controllo del riutilizzo delle password).
- Autorizzazione GRANT TO PUBLIC: puoi concedere i privilegi a tutti gli utenti che hanno accesso al tuo server. Per ulteriori informazioni, consulta GRANT TO PUBLIC.
- Separazione dei privilegi SUPER e READ ONLY ADMIN: puoi rimuovere i privilegi READ ONLY ADMIN da tutti gli utenti, anche dagli utenti che in precedenza avevano i privilegi SUPER.
- Sicurezza: ora puoi impostare l'opzione `--ssl` come impostazione predefinita per il tuo client MariaDB. MariaDB non disabilita più SSL automaticamente se la configurazione non è corretta.
- Comandi e funzioni SQL: ora puoi usare il comando `SHOW ANALYZE FORMAT=JSON` e le funzioni `ROW_NUMBER`,`SFORMAT` e `RANDOM_BYTES`. `SFORMAT` consente la formattazione delle stringhe ed è abilitata per impostazione predefinita. Puoi eseguire la conversione da partizione a tabella e da tabella a partizione con un solo comando. Ci sono anche diversi miglioramenti relativi alle funzioni

`JSON_*()`. Le funzioni `DES_ENCRYPT` e `DES_DECRYPT` sono obsolete per la versione 10.10 e successive. Per ulteriori informazioni, consulta SFORMAT.

- Miglioramenti di InnoDB: i miglioramenti includono i seguenti elementi:
  - Miglioramenti delle prestazioni nel redo log per ridurre l'amplificazione della scrittura e migliorare la concorrenza.
  - La possibilità di modificare la tablespace di undo senza reinizializzare la directory dei dati. Questo miglioramento riduce il sovraccarico del piano di controllo (control plane). Richiede il riavvio ma non la reinizializzazione dopo aver modificato la tablespace di undo.
  - Supporto interno per `CHECK TABLE … EXTENDED` e per gli indici decrescenti.
  - Miglioramenti dell'inserimento in blocco.
- Modifiche a binlog: queste modifiche includono i seguenti elementi:
  - Generazione di log di `ALTER` in due fasi per ridurre la latenza di replica. Il parametro `binlog_alter_two_phase` è disabilitato per impostazione predefinita, ma può essere abilitato tramite i gruppi di parametri.
  - Generazione di log di `explicit_defaults_for_timestamp`.
  - Non vengono più generati i log `INCIDENT_EVENT` se la transazione può essere ripristinata in modo sicuro.
- Miglioramenti della replica: le istanze database MariaDB versione 10.11 utilizzano la replica GTID per impostazione predefinita se il master la supporta. Inoltre,`Seconds_Behind_Master` è più preciso.
- Client: puoi utilizzare nuove opzioni della riga di comando per `mysqlbinglog` e `mariadb-dump`. Puoi usare `mariadb-dump` per scaricare e ripristinare i dati storici.
- Controllo delle versioni del sistema: è possibile modificare la cronologia. MariaDB crea automaticamente nuove partizioni.
- DDL atomico: `CREATE OR REPLACE` ora è atomico. L'istruzione ha esito positivo o è completamente invertita.
- Scrittura dei redo log: i redo log vengono scritti in modo asincrono.
- Funzioni archiviate: le funzioni archiviate ora supportano gli stessi parametri `IN`, `OUT` e `INOUT` delle stored procedure.
- Parametri obsoleti o rimossi: i seguenti parametri sono obsoleti o sono stati rimossi per le istanze database MariaDB versione 10.11:
  - innodb_change_buffering
  - innodb_disallow_writes

  - innodb_log_write_ahead_size
  - innodb_prefix_index_cluster_optimization
  - keep_files_on_create
  - old
- Parametri dinamici: i seguenti parametri ora sono dinamici per le istanze database diMariaDB versione 10.11:
  - innodb_log_file_size
  - innodb_write_io_threads
  - innodb_read_io_threads
- Nuovi valori predefiniti per i parametri: i seguenti parametri hanno nuovi valori predefiniti per le istanze database MariaDB versione 10.11:
  - Il valore predefinito del parametro explicit_defaults_for_timestampmodificato da `OFF` in `ON`.
  - Il valore predefinito del parametro optimizer_prune_level modificato da 1 in 2.
- Nuovi valori validi per i parametri: i seguenti parametri hanno nuovi valori validi per le istanze database MariaDB versione 10.11:
  - I valori validi per il parametro old sono stati uniti a quelli del parametro old_mode.
  - I valori validi del parametro histogram_type ora includono `JSON_HB`.
  - L'intervallo di valori valido per il parametro innodb_log_buffer_size è ora da `262144` a `4294967295`(da 256 KB a 4096 MB).
  - L'intervallo di valori valido per il parametro innodb_log_file_size è ora da `4194304` a `512GB` (da 4 MB a 512 GB).
  - I valori validi per il parametro optimizer_prune_level ora include2.
- Nuovi parametri: i seguenti parametri sono nuovi per le istanze database MariaDB versione 10.11:
  - Il parametro binlog_alter_two_phaseil parametro può migliorare le prestazioni della replica.
  - Il parametro log_slow_min_examed_row_limit può migliorare le prestazioni.
  - Il parametro log_slow_query e il parametro log_slow_query_file sono alias rispettivamente per `slow_query_log` e `slow_query_log_file`, .
  - optimizer_extra_pruning_depth
  - system_versioning_insert_history

Per un elenco di tutte le funzionalità e la documentazione, consulta le seguenti informazioni sul sito Web di MariaDB.

Versioni	Modifiche e miglioramenti	Note di rilascio
MariaDB 10.7	Modifiche e miglioramenti in MariaDB 10.7	Note di rilascio - MariaDB serie 10.7
MariaDB 10.8	Modifiche e miglioramenti in MariaDB 10.8	Note di rilascio - MariaDB serie 10.8
MariaDB 10.9	Modifiche e miglioramenti in MariaDB 10.9	Note di rilascio - MariaDB serie 10.9
MariaDB 10.10	Modifiche e miglioramenti in MariaDB 10.10	Note di rilascio - MariaDB serie 10.10
MariaDB 10.11	Modifiche e miglioramenti in MariaDB 10.11	Note di rilascio - MariaDB serie 10.11

Per un elenco delle funzionalità non supportate, consulta Funzionalità MariaDB non supportate da Amazon RDS.

Supporto per MariaDB 10.6 in Amazon RDS

Amazon RDS supporta le seguenti nuove funzionalità per le istanze database che eseguono MariaDB versione 10.6 o versioni successive:

- Motore di storage MyRocks: è possibile utilizzare il motore di storage MyRocks con RDS for MariaDB per ottimizzare il consumo di storage delle applicazioni Web ad alte prestazioni e a uso intensivo di scrittura. Per ulteriori informazioni, consulta Motori di storage supportati per MariaDB in Amazon RDS e MyRocks.
- autenticazione DB (IAM) AWS Identity and Access Management: puoi utilizzare l'autenticazione DB IAM per una migliore sicurezza e gestione centrale delle connessioni alle istanze database MariaDB. Per ulteriori informazioni, consulta Autenticazione del database IAM per MariaDB, MySQL e PostgreSQL.
- Opzioni di aggiornamento: ora è possibile eseguire l'aggiornamento a RDS per MariaDB versione 10.6 da qualsiasi versione precedente (10.3, 10.4, 10.5). Inoltre, puoi ripristinare uno snapshot

di un'istanza database di MySQL 5.6 o 5.7 esistente in un'istanza MariaDB 10.6. Per ulteriori informazioni, consulta Aggiornamento del motore di database MariaDB.

- Replica ritardata: ora è possibile impostare un periodo di tempo configurabile per cui una replica di lettura ritarda rispetto al database di origine. In una configurazione di replica MariaDB standard, esiste un ritardo di replica minimo tra l'origine e la replica. Con la replica ritardata, puoi impostare un ritardo intenzionale come strategia per il ripristino di emergenza. Per ulteriori informazioni, consulta Configurazione della replica ritardata con MariaDB.
- Compatibilità con Oracle PL/SQL: utilizzando RDS for MariaDB versione 10.6, puoi migrare più facilmente le tue applicazioni Oracle legacy ad Amazon RDS. Per ulteriori informazioni, consulta SQL_MODE=ORACLE.
- DDL atomico: le istruzioni Dynamic Data Language (DDL) possono essere relativamente protette con RDS for MariaDB versione 10.6. `CREATE TABLE`, `ALTER TABLE`, `RENAME TABLE`, `DROP TABLE`, `DROP DATABASE` e le relative istruzioni DDL sono ora atomiche. L'istruzione ha esito positivo o è completamente invertita. Per ulteriori informazioni, consulta DDL atomico.
- Altri miglioramenti: questi miglioramenti includono una funzione `JSON_TABLE` per trasformare i dati JSON in formato relazionale all'interno di SQL e un caricamento più rapido dei dati della tabella vuota con Innodb. Includono anche nuovi `sys_schema` per l'analisi e la risoluzione dei problemi, miglioramenti di Optimizer per ignorare gli indici inutilizzati e miglioramenti delle prestazioni. Per ulteriori informazioni, consulta JSON_TABLE.
- Nuovi valori predefiniti per i parametri: i seguenti parametri hanno nuovi valori predefiniti per le istanze database di MariaDB versione 10.6:
  - Il valore predefinito per i seguenti parametri è cambiato da `utf8` a `utf8mb3`:
    - character_set_client
    - character_set_connection
    - character_set_results
    - character_set_system

    Sebbene i valori predefiniti siano cambiati per questi parametri, non vi è alcuna modifica funzionale. Per ulteriori informazioni, consulta Set di caratteri e regole di confronto supportati nella documentazione di MariaDB.
  - Il valore predefinito del parametro collation_connection è stato modificato da `utf8_general_ci` a `utf8mb3_general_ci`. Sebbene il valore predefinito sia cambiato per questo parametri, non vi è alcuna modifica funzionale.

- Il valore predefinito del parametro old_mode è stato modificato da non impostato a `UTF8_IS_UTF8MB3`. Sebbene il valore predefinito sia cambiato per questo parametri, non vi è alcuna modifica funzionale.

Per un elenco di tutte le funzionalità di MariaDB 10.6 e la relativa documentazione, consulta Modifiche e miglioramenti in MariaDB 10.6 e Note di rilascio - MariaDB 10.6 Series sul sito Web di MariaDB.

Per un elenco delle funzionalità non supportate, consulta Funzionalità MariaDB non supportate da Amazon RDS.

Supporto per MariaDB 10.5 in Amazon RDS

Amazon RDS supporta le seguenti nuove funzionalità per le istanze database che eseguono MariaDB versione 10.5 o successive:

- Miglioramenti InnoDB – MariaDB versione 10.5 include miglioramenti InnoDB. Per ulteriori informazioni, consulta InnoDB: Miglioramenti delle prestazioni ecc. nella documentazione di MariaDB.
- Aggiornamenti dello schema delle prestazioni – MariaDB versione 10.5 include aggiornamenti dello schema delle prestazioni. Per ulteriori informazioni, consulta Aggiornamenti dello schema delle prestazioni che corrispondono alla strumentazione e alle tabelle MySQL 5.7 nella documentazione di MariaDB.
- Un file nel log redo InnoDB – Nelle versioni di MariaDB prima della versione 10.5, il valore del parametro `innodb_log_files_in_group` era impostato su 2. In MariaDB versione 10.5, il valore di questo parametro è impostato su 1.

  Se si esegue l'aggiornamento da una versione precedente a MariaDB versione 10.5 e non si modificano i parametri, il valore del parametro `innodb_log_file_size` rimane invariato. Tuttavia, si applica a un file di log anziché a due. Il risultato è che l'istanza di MariaDB versione 10.5 aggiornata utilizza metà delle dimensioni del log redo che stava utilizzando prima dell'aggiornamento. Questa modifica può avere un notevole impatto sulle prestazioni. Per risolvere questo problema, è possibile raddoppiare il valore del parametro `innodb_log_file_size`. Per ulteriori informazioni sulla modifica dei parametri, consulta Modifica di parametri in un gruppo di parametri del database.
- Comando SHOW SLAVE STATUS non supportato – Nelle versioni di MariaDB precedenti alla versione 10.5, il comando `SHOW SLAVE STATUS` richiedeva il privilegio `REPLICATION SLAVE`.

In MariaDB versione 10.5, il comando `SHOW REPLICA STATUS` equivalente richiede il privilegio `REPLICATION REPLICA ADMIN`. Questo nuovo privilegio non viene concesso all'utente master RDS.

Anziché utilizzare il comando `SHOW REPLICA STATUS`, eseguire la nuova procedura memorizzata `mysql.rds_replica_status` per restituire informazioni simili. Per ulteriori informazioni, consulta mysql.rds_replica_status.

- Comando SHOW RELAYLOG EVENTS non supportato – Nelle versioni di MariaDB precedenti alla versione 10.5, il comando `SHOW RELAYLOG EVENTS` richiedeva il privilegio `REPLICATION SLAVE`. In MariaDB versione 10.5, questo comando richiede il privilegio `REPLICATION REPLICA ADMIN`. Questo nuovo privilegio non viene concesso all'utente master RDS.
- Nuovi valori predefiniti per i parametri – I seguenti parametri hanno nuovi valori predefiniti per le istanze database di MariaDB versione 10.5:
  - Il valore predefinito del parametro max_connections è stato modificato in `LEAST({DBInstanceClassMemory/25165760},12000)`. Per informazioni sulla funzione del parametro `LEAST`, consulta Funzioni dei parametri database.
  - Il valore predefinito del parametro innodb_adaptive_hash_index è stato modificato in `OFF` (`0`).
  - Il valore predefinito del parametro innodb_checksum_algorithm è stato modificato in `full_crc32`.
  - Il valore predefinito del parametro innodb_log_file_size è stato modificato in 2 GB.

Per un elenco di tutte le funzionalità di MariaDB 10.5 e la relativa documentazione, consulta Modifiche e miglioramenti in MariaDB 10.5 e Note di rilascio - MariaDB 10.5 Series sul sito Web di MariaDB.

Per un elenco delle funzionalità non supportate, consulta Funzionalità MariaDB non supportate da Amazon RDS.

Supporto per MariaDB 10.4 in Amazon RDS

Amazon RDS supporta le seguenti nuove funzionalità per le istanze database che eseguono MariaDB versione 10.4 o successive:

- Miglioramenti in termini di sicurezza dell'account utente – Scadenza password e miglioramenti del blocco account
- Miglioramenti di Optimizer – Funzionalità Optimizer Trace

- Miglioramenti di InnoDB – Supporto Instant DROP COLUMN ed estensione VARCHAR istant per ROW_FORMAT=DYNAMIC e ROW_FORMAT=COMPACT
- Nuovi parametri – Inclusi tcp_nodedelay, tls_version e gtid_cleanup_batch_size

Per un elenco di tutte le funzionalità di MariaDB 10.4 e la relativa documentazione, consulta Changes & Improvements in MariaDB 10.4 e Release Notes - MariaDB 10.4 Series sul sito Web di MariaDB.

Per un elenco delle funzionalità non supportate, consulta Funzionalità MariaDB non supportate da Amazon RDS.

Supporto per MariaDB 10.3 in Amazon RDS

Amazon RDS supporta le seguenti nuove funzionalità per le istanze database che eseguono MariaDB 10.3 o versione successiva:

- Compatibilità con Oracle – Parser di compatibilità PL/SQL, sequenze, integrazione di INTERSECT ed EXCEPT con UNION, nuove dichiarazioni TYPE OF e ROW TYPE OF e colonne invisibili
- Elaborazione di dati temporali – Tabelle con versione di sistema per consentire l'elaborazione di query degli stati presenti e passati del database
- Flessibilità – Aggregazioni definite dall'utente, compressione delle colonne indipendente dallo storage e supporto del protocollo proxy per l'inoltro dell''indirizzo IP del client direttamente al server
- Gestibilità – Operazioni ADD COLUMN immediate e DDL (Data Definition Language) con risposta immediata agli errori.

Per un elenco di tutte le funzionalità di MariaDB 10.3 e la relativa documentazione, consulta Changes & Improvements in MariaDB 10.3 e Release Notes - MariaDB 10.3 Series sul sito Web di MariaDB.

Per un elenco delle funzionalità non supportate, consulta Funzionalità MariaDB non supportate da Amazon RDS.

Motori di storage supportati per MariaDB in Amazon RDS

RDS per MariaDB supporta i seguenti motori di storage.

Argomenti

- Motore di storage InnoDB
- Il motore di storage MyRocks

Altri motori di storage non sono attualmente supportati da Amazon RDS for MariaDB.

Motore di storage InnoDB

Sebbene MariaDB supporti più motori di storage con funzionalità diverse, non tutti sono ottimizzati per il recupero e per la durata dei dati. InnoDB è il motore di storage consigliato per istanze di database MariaDB su Amazon RDS. Le funzionalità di ripristino point-in-time e ripristino di snapshot di Amazon RDS richiedono un motore di archiviazione recuperabile e sono disponibili solo per il motore di storage suggerito per la versione di MariaDB.

Per ulteriori informazioni, consulta InnoDB.

Il motore di storage MyRocks

Il motore di storage MyRocks è disponibile in RDS per MariaDB versione 10.6 e versioni successive. Prima di utilizzare il motore di storage MyRocks in un database di produzione, ti consigliamo di eseguire benchmark e test approfonditi per verificare eventuali potenziali vantaggi rispetto a InnoDB per il tuo caso d'uso.

Il gruppo di parametri predefinito per MariaDB versione 10.6 include i parametri MyRocks. Per ulteriori informazioni, consulta Parametri per MariaDB e Utilizzo di gruppi di parametri.

Per creare una tabella che utilizza il motore di storage MyRocks, specifica `ENGINE=RocksDB` nell'istruzione `CREATE TABLE`. Nell'esempio seguente viene creata una tabella che utilizza il motore di storage MyRocks.

```
CREATE TABLE test (a INT NOT NULL, b CHAR(10)) ENGINE=RocksDB;
```

Sconsigliamo vivamente di non eseguire transazioni che coprono sia le tabelle InnoDB che MyRocks. MariaDB non garantisce ACID (Atomicity, Consistency, Isolation, Durability) per le transazioni tra i motori di storage. Sebbene sia possibile avere sia le tabelle InnoDB che MyRocks in un'istanza database, questo approccio non è consigliato se non durante la migrazione da un motore di storage all'altro. Quando entrambe le tabelle InnoDB e MyRocks esistono in un'istanza database, ogni motore di storage ha il proprio pool di buffer, che potrebbe causare un deterioramento delle prestazioni.

MyRocks non supporta i blocchi gap o di isolamento `SERIALIZABLE`. Pertanto, in genere non è possibile utilizzare MyRocks con la replica basata sulle istruzioni. Per ulteriori informazioni, consulta MyRocks e replica.

Attualmente, è possibile modificare soltanto i seguenti parametri MyRocks:

- rocksdb_block_cache_size
- rocksdb_bulk_load
- rocksdb_bulk_load_size
- rocksdb_deadlock_detect
- rocksdb_deadlock_detect_depth
- rocksdb_max_latest_deadlocks

Il motore di storage MyRocks e il motore di storage InnoDB possono competere per la memoria in base alle impostazioni per i parametri `rocksdb_block_cache_size` e `innodb_buffer_pool_size`. In alcuni casi, potresti voler utilizzare soltanto il motore di storage MyRocks su una particolare istanza database. In tal caso, ti consigliamo di impostare il parametro `innodb_buffer_pool_size minimal` su un valore minimo e `rocksdb_block_cache_size` il più in alto possibile.

È possibile accedere ai file di log di MyRocks utilizzando le operazioni `DescribeDBLogFiles` e `DownloadDBLogFilePortion`.

Per ulteriori informazioni su MyRocks, consulta MyRocks sul sito Web di MariaDB.

Precaricamento della cache per MariaDB in Amazon RDS

Il precaricamento della cache InnoDB può offrire vantaggi in termini di prestazioni per l'istanza database MariaDB salvando lo stato corrente del pool di buffer quando l'istanza database viene arrestata e quindi ricaricando il pool di buffer con le informazioni salvate quando l'istanza database si avvia. Questo approccio elimina la necessità di preparare il pool di buffer a partire da un utilizzo normale del database e consente invece di precaricare il pool di buffer con le pagine per le query comuni note. Per ulteriori informazioni sul precaricamento della cache, consulta Dump e ripristino del pool di buffer nella documentazione di MariaDB.

Il precaricamento della cache è abilitato per impostazione predefinita nelle istanze database MariaDB 10.3 e versioni successive. Per abilitarlo, imposta i parametri `innodb_buffer_pool_dump_at_shutdown` e `innodb_buffer_pool_load_at_startup` su 1 nel gruppo di parametri per l'istanza database. La modifica dei valori di questi parametri in un gruppo di parametri ha effetto su tutte le istanze database MariaDB che utilizzano tale gruppo di parametri. Per abilitare il precaricamento della cache per istanze database MariaDB specifiche, potrebbe essere necessario creare un nuovo gruppo di parametri per tali istanze database. Per informazioni sui gruppi di parametri, consulta Utilizzo di gruppi di parametri.

Il precaricamento della cache fornisce principalmente un vantaggio in termini di prestazioni per le istanze database che utilizzano lo storage standard. Se utilizzi lo storage PIOPS probabilmente non riscontrerai un vantaggio significativo in termini di prestazioni.

> **Important**
>
> Se l'istanza database MariaDB non si arresta normalmente, ad esempio durante un failover, lo stato del pool di buffer non è salvato su disco. In questo caso, al riavvio dell'istanza database, MariaDB carica il file del pool di buffer disponibile. Ciò non comporta alcun problema, ma il pool di buffer ripristinato potrebbe non riflettere lo stato più recente del pool di buffer prima del riavvio. Per assicurarti di disporre di uno stato recente del pool di buffer per precaricare la cache all'avvio, consigliamo di eseguire periodicamente un dump del pool di buffer "on demand". Puoi eseguire il dump del pool di buffer o caricarlo on demand.
> Puoi creare un evento per eseguire il dump del pool di buffer automaticamente e a intervalli regolari. L'istruzione seguente crea ad esempio un evento denominato `periodic_buffer_pool_dump` che esegue il dump del pool di buffer ogni ora.
>
> ```
> CREATE EVENT periodic_buffer_pool_dump
>    ON SCHEDULE EVERY 1 HOUR
>    DO CALL mysql.rds_innodb_buffer_pool_dump_now();
> ```
>
> Per ulteriori informazioni, consulta Events nella documentazione di MariaDB.

Dump e caricamento del pool di buffer on demand

Puoi salvare e caricare la cache on demand mediante le seguenti stored procedure:

- Per eseguire il dump dello stato corrente del pool di buffer su disco, chiama la stored procedure mysql.rds_innodb_buffer_pool_dump_now.
- Per caricare lo stato salvato del pool di buffer dal disco, chiama la stored procedure mysql.rds_innodb_buffer_pool_load_now.
- Per annullare un'operazione di caricamento in corso, chiama la stored procedure mysql.rds_innodb_buffer_pool_load_abort.

Funzionalità MariaDB non supportate da Amazon RDS

Le seguenti funzionalità di MariaDB non sono supportate in Amazon RDS:

- Motore di storage S3
- Plug-in di autenticazione – GSSAPI
- Plug-in di autenticazione – Unix Socket
- AWSPlug-in di crittografia Key Management
- Replica ritardata per le versioni di MariaDB inferiori alla 10.6
- Crittografia MariaDB nativa dei dati inattivi per InnoDB e Aria

  Puoi abilitare la crittografia dei dati inattivi per un'istanza database MariaDB seguendo le istruzioni in Crittografia delle risorse Amazon RDS.
- HandlerSocket
- Tipo di tabella JSON per le versioni MariaDB inferiori alla 10.6
- MariaDB ColumnStore
- MariaDB Galera Cluster
- Replica multi-source
- Motore di storage MyRocks per le versioni MariaDB inferiori alla 10.6
- Plugin convalida password, `simple_password_check` e `cracklib_password_check`
- Filtri di storage Spider
- Filtri di storage Sphinx
- Motore di storage TokuDB
- Attributi di oggetti specifici per il motore di storage, come descritto nell'argomento relativo ai nuovi attributi tabella/campo/indice definiti dal motore nella documentazione di MariaDB.
- Crittografia di tabelle e spazi tabelle
- Plugin per la gestione delle chiavi Hashicorp
- Esecuzione di due aggiornamenti in parallelo

Per offrire un'esperienza di servizio gestito, Amazon RDS non fornisce accesso shell alle istanze database e limita l'accesso a determinate procedure e tabelle di sistema che richiedono privilegi avanzati. Amazon RDS supporta l'accesso ai database su un'istanza database mediante qualsiasi applicazione client SQL standard. Amazon RDS non consente l'accesso host diretto a un'istanza database tramite Telnet, Secure Shell (SSH) o Windows Remote Desktop Connection.

Versioni di MariaDB in Amazon RDS

Per MariaDB, i numeri di versione sono organizzati come versione X.Y.Z. Nella terminologia di Amazon RDS; X.Y indica la versione principale e Z è il numero di versione secondaria. Per le implementazioni di Amazon RDS, una modifica di versione è considerata principale se cambia il numero di versione principale, ad esempio nel caso di un passaggio dalla versione 10.5 alla 10.6. Una modifica di versione è considerata secondaria se cambia solo il numero della versione secondaria, ad esempio passando dalla versione 10.6.14 alla 10.6.16.

Argomenti

Versioni secondarie di MariaDB supportate in Amazon RDS

Attualmente Amazon RDS supporta le seguenti versioni secondarie di MariaDB.

> **Note**
>
> Le date con solo un mese e un anno sono approssimative e vengono aggiornate con una data esatta quando nota.

Versione del motore di MariaDB	Data di rilascio nella community	Data di rilascio per RDS	Data di fine del supporto standard RDS
10.11			
10.11.7	7 febbraio 2024	26 febbraio 2024	Marzo 2025
10.11.6	13 novembre 2023	12 dicembre 2023	Marzo 2025
10.11.5	14 agosto 2023	7 settembre 2023	Settembre 2024
10.11.4	7 giugno 2023	21 agosto 2023	Settembre 2024

Versione del motore di MariaDB	Data di rilascio nella community	Data di rilascio per RDS	Data di fine del supporto standard RDS
10.6			
10,6,17	7 febbraio 2024	26 febbraio 2024	Marzo 2025
10.6.16	13 novembre 2023	12 dicembre 2023	Marzo 2025
10.6.15	14 agosto 2023	7 settembre 2023	Settembre 2024
10,6,14	7 giugno 2023	22 giugno 2023	Settembre 2024
10,6,13	10 maggio 2023	15 giugno 2023	Settembre 2024
10.5			
10,5,24	7 febbraio 2024	26 febbraio 2024	Marzo 2025
10.5.23	13 novembre 2023	12 dicembre 2023	Marzo 2025
10.5.22	14 agosto 2023	7 settembre 2023	Settembre 2024
10,5,21	7 giugno 2023	22 giugno 2023	Settembre 2024
10,5,20	10 maggio 2023	15 giugno 2023	Settembre 2024
10.4			
10,4,33	7 febbraio 2024	26 febbraio 2024	agosto 2024
10.4.32	13 novembre 2023	12 dicembre 2023	agosto 2024
10.4.31	14 agosto 2023	7 settembre 2023	agosto 2024
10.4.30	7 giugno 2023	22 giugno 2023	agosto 2024
10.4.29	10 maggio 2023	15 giugno 2023	agosto 2024

Quando crei una nuova istanza database, puoi specificare qualsiasi versione di MariaDB attualmente supportata. Puoi specificare la versione principale (come MariaDB 10.5) e qualsiasi versione secondaria supportata per la versione principale specificata. Se non viene specificata alcuna versione, Amazon RDS utilizza per impostazione predefinita una versione supportata, in genere la più recente. Se viene specificata una versione principale ma non una secondaria, per impostazione predefinita Amazon RDS utilizza una release recente della versione principale specificata. Per visualizzare un elenco delle versioni supportate, nonché le impostazioni predefinite per le istanze DB appena create, usa il comando. `describe-db-engine-versions` AWS CLI

Ad esempio, per elencare le versioni del motore supportate per RDS per MariaDB, esegui il comando CLI seguente:

```
aws rds describe-db-engine-versions --engine mariadb --query "*[].
{Engine:Engine,EngineVersion:EngineVersion}" --output text
```

La versione predefinita di MariaDB potrebbe variare in base alla Regione AWS. Per creare un'istanza DB con una versione secondaria specifica, specifica la versione secondaria durante la creazione dell'istanza DB. È possibile determinare la versione secondaria predefinita di an Regione AWS utilizzando il seguente comando: AWS CLI

```
aws rds describe-db-engine-versions --default-only --engine mariadb
 --engine-version major-engine-version --region region --query "*[].
{Engine:Engine,EngineVersion:EngineVersion}" --output text
```

Sostituisci *major-engine-version*con la versione principale del motore e sostituisci la *regione* con Regione AWS. Ad esempio, il AWS CLI comando seguente restituisce la versione predefinita del motore secondario MariaDB per la versione principale 10.5 e gli Stati Uniti occidentali (Oregon) (us-west-2): Regione AWS

```
aws rds describe-db-engine-versions --default-only --engine mariadb --engine-version
 10.5 --region us-west-2 --query "*[].{Engine:Engine,EngineVersion:EngineVersion}" --
output text
```

Versioni principali di MariaDB supportate in Amazon RDS

Le versioni principali di RDS per MariaDB restano disponibili almeno fino alla fine del ciclo di vita della community per la versione della community corrispondente. È possibile utilizzare le date seguenti per pianificare i cicli di test e aggiornamento. Se Amazon estende il supporto per una versione di RDS

per MariaDB più a lungo di quanto inizialmente previsto, questa tabella verrà aggiornata in base alla nuova data.

> **Note**
>
> Le date con solo un mese e un anno sono approssimative e vengono aggiornate con una data esatta quando nota.

Versione principale di MariaDB	Data di rilascio nella community	Data di rilascio per RDS	Data di fine vita nella community	Data di fine del supporto standard RDS
MariaDB 10.11	16 febbraio 2023	21 agosto 2023	16 febbraio 2028	Febbraio 2028
MariaDB 10.6	6 luglio 2021	3 febbraio 2022	6 luglio 2026	Luglio 2026
MariaDB 10.5	24 giugno 2020	21 gennaio 2021	24 giugno 2025	Giugno 2025
MariaDB 10.4	18 giugno 2019	6 aprile 2020	18 giugno 2024	agosto 2024

Versioni obsolete per Amazon RDS for MariaDB

Le versioni di Amazon RDS for MariaDB 10.0, 10.1, 10.2 e 10.3 sono obsolete.

Per informazioni sulla policy di deprecazione di Amazon RDS for MariaDB, consulta la pagina Domande frequenti su Amazon RDS.

Connessione a un'istanza database che esegue il motore di database MariaDB

Dopo che Amazon RDS ha effettuato il provisioning dell'istanza database, puoi utilizzare qualsiasi utilità o applicazione client MariaDB standard per connetterti all'istanza. Nella stringa di connessione, specifica l'indirizzo Domain Name System (DNS) dell'endpoint dell'istanza database come il parametro host. Puoi inoltre specificare il numero di porta dell'endpoint dell'istanza database come parametro porta.

Puoi stabilire la connessione ad un'istanza database Amazon RDS for MariaDB usando strumenti come il client della riga di comando MySQL. Per ulteriori informazioni sull'utilizzo del client della riga di comando MySQL, consulta Client della riga di comando mysql nella documentazione di MariaDB. Un'applicazione basata su GUI che puoi utilizzare per la connessione è Heidi. Per ulteriori informazioni, consulta la pagina relativa al download di HeidiSQL. Per informazioni sull'installazione di MySQL (compreso il client della riga di comando MySQL), consulta Installazione e aggiornamento di MySQL.

La maggior parte delle distribuzioni Linux include il client MariaDB invece del client Oracle MySQL. Per installare il client della linea di comando MySQL su Amazon Linux 2023, esegui il comando seguente:

```
sudo dnf install mariadb105
```

Per installare il client della linea di comando MySQL su Amazon Linux 2, esegui il comando seguente:

```
sudo yum install mariadb
```

Per installare il client a riga di comando MySQL sulla maggior parte delle distribuzioni Linux basate su DEB, esegui il comando seguente:

```
apt-get install mariadb-client
```

Per controllare la versione del client a riga di comando MySQL, esegui il comando seguente.

```
mysql --version
```

Per leggere la documentazione MySQL per la versione corrente del client, esegui il comando seguente:

```
man mysql
```

Per connettersi a un'istanza database dall'esterno di un cloud privato virtuale (VPC) basato su Amazon VPC, l'istanza database deve essere accessibile pubblicamente. Inoltre, l'accesso deve essere concesso utilizzando le regole in ingresso del gruppo di sicurezza dell'istanza database e devono essere soddisfatti altri requisiti. Per ulteriori informazioni, consulta Impossibile connettersi all'istanza database di Amazon RDS.

Puoi utilizzare la crittografia SSL nelle connessioni a un'istanza database MariaDB. Per informazioni, consulta Utilizzo di SSL/TLS con un'istanza database MariaDB.

Argomenti

- Ricerca delle informazioni di connessione per un'istanza database MariaDB
- Connessione dal client a riga di comando MySQL (non crittografato)
- Connessione a RDS per MariaDB con il driver JDBC Amazon Web Services ()AWS
- Connessione a RDS per MariaDB con il driver Python Amazon Web Services ()AWS
- Risoluzione dei problemi relativi alle connessioni all'istanza database MariaDB

Ricerca delle informazioni di connessione per un'istanza database MariaDB

Le informazioni di connessione per un'istanza database includono l'endpoint, la porta e un utente di database valido, ad esempio l'utente master. Si supponga, ad esempio, che un valore endpoint sia `mydb.123456789012.us-east-1.rds.amazonaws.com`. In questo caso, il valore della porta è `3306` e l'utente del database è `admin`. Date queste informazioni, è possibile specificare i seguenti valori in una stringa di connessione:

- Per host, nome host o nome DNS, specifica `mydb.123456789012.us-east-1.rds.amazonaws.com`.
- Per la porta, specific `3306`.
- Per l'utente, specifica `admin`.

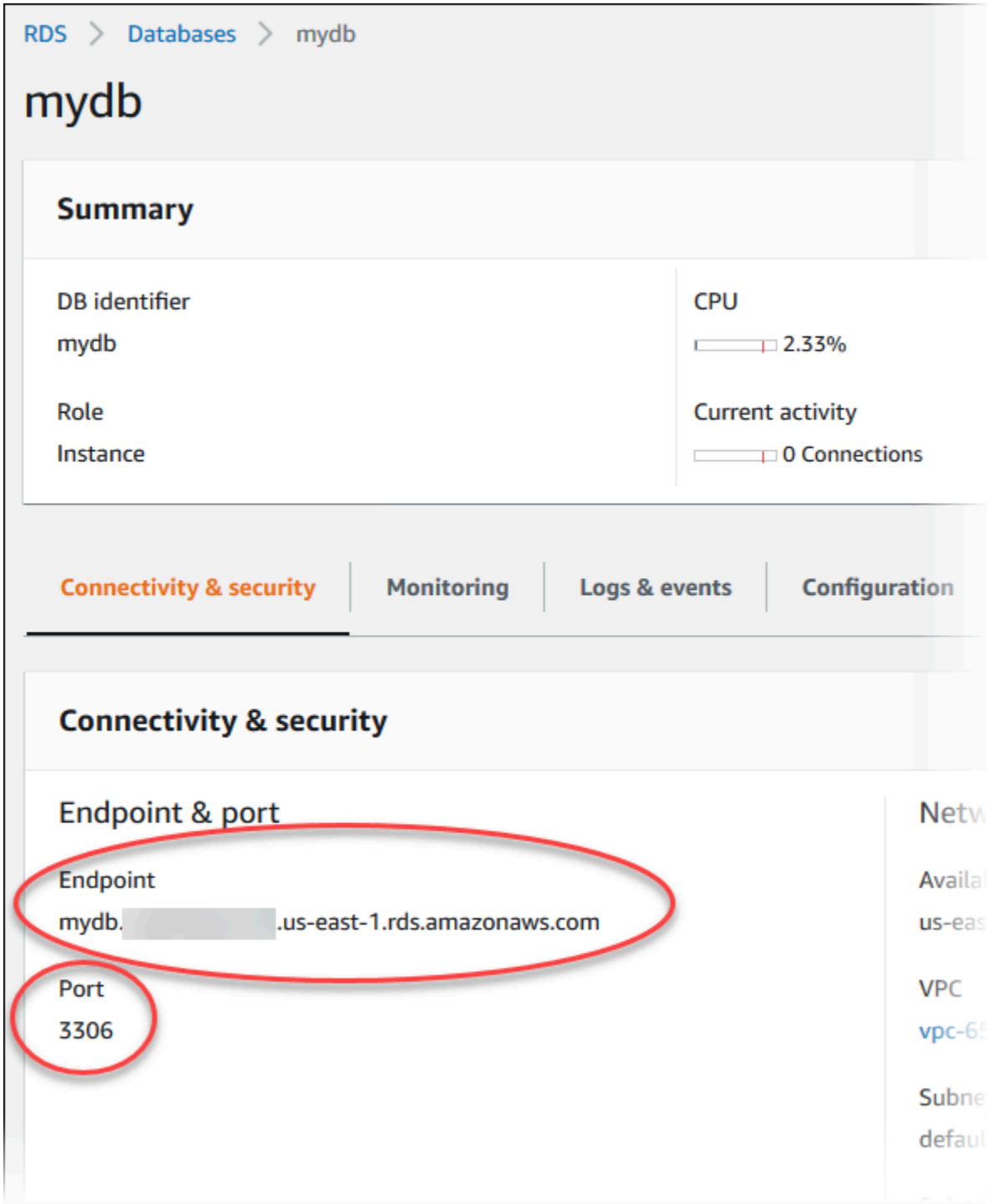
Per connettersi a un'istanza database, utilizzare qualsiasi client per un motore di database MariaDB. Ad esempio, è possibile utilizzare il client a riga di comando MySQL o MySQL Workbench.

Per trovare le informazioni di connessione per un'istanza DB, puoi utilizzare il describe-db-instancescomando AWS Management Console, the AWS Command Line Interface (AWS CLI) o l'operazione DescribedBInstances API di Amazon RDS per elencarne i dettagli.

Console

Per trovare le informazioni di connessione per un'istanza DB nel AWS Management Console

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di spostamento scegliere Database per visualizzare un elenco delle istanze database.
3. Scegliere il nome dell'istanza database MariaDB per visualizzarne i dettagli.
4. Nella scheda Connectivity & security (Connettività e sicurezza), copiare l'endpoint. Annotare anche il numero di porta. L'endpoint e il numero di porta sono necessari per la connessione all'istanza database.



5. Se è necessario trovare il nome utente master, scegliere la scheda Configurazione e visualizzare il valore del nome utente principale .

AWS CLI

Per trovare le informazioni di connessione per un'istanza di MariaDB DB utilizzando AWS CLI il, chiamate il comando. describe-db-instances Nella chiamata, eseguire una query per l'ID istanza database, l'endpoint, la porta e il nome utente master.

PerLinux, omacOS: Unix

```
aws rds describe-db-instances \
  --filters "Name=engine,Values=mariadb" \
  --query "*[].[DBInstanceIdentifier,Endpoint.Address,Endpoint.Port,MasterUsername]"
```

Per Windows:

```
aws rds describe-db-instances ^
  --filters "Name=engine,Values=mariadb" ^
  --query "*[].[DBInstanceIdentifier,Endpoint.Address,Endpoint.Port,MasterUsername]"
```

L'output visualizzato dovrebbe essere simile al seguente.

```
[
    [
        "mydb1",
        "mydb1.123456789012.us-east-1.rds.amazonaws.com",
        3306,
        "admin"
    ],
    [
        "mydb2",
        "mydb2.123456789012.us-east-1.rds.amazonaws.com",
        3306,
        "admin"
    ]
]
```

API RDS

Per trovare le informazioni di connessione per un'istanza database utilizzando l'API Amazon RDS, richiamare l'operazione DescribeDBInstances. Nell'output, individuare i valori per l'indirizzo dell'endpoint, la porta dell'endpoint e il nome utente master.

Connessione dal client a riga di comando MySQL (non crittografato)

> **Important**
>
> Utilizzare una connessione MySQL non crittografata solo quando il client e il server sono nello stesso VPC e la rete è attendibile. Per ulteriori informazioni sull'uso di connessioni crittografate, consulta Connessione dal client a riga di comando MySQL con SSL/TLS (crittografato).

Per connetterti a un'istanza database utilizzando il client della riga di comando MySQL, immetti il seguente comando al prompt dei comandi su un computer client. In questo modo esegui la connessione a un database su un'istanza database MariaDB. Sostituisci il nome DNS (endpoint) per l'istanza database con *<endpoint>* e il nome utente master utilizzato con *<mymasteruser>*. Devi fornire la password master utilizzata quando viene richiesta una password.

```
mysql -h <endpoint> -P 3306 -u <mymasteruser> -p
```

Dopo avere inserito la password per l'utente, l'output dovrebbe essere analogo a quanto mostrato di seguito.

```
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 31
Server version: 10.6.10-MariaDB-log Source distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]>
```

Connessione a RDS per MariaDB con il driver JDBC Amazon Web Services ()AWS

Il driver JDBC di Amazon Web Services (AWS) è progettato come wrapper JDBC avanzato. Questo wrapper è complementare e amplia le funzionalità di un driver JDBC esistente. Il driver è compatibile direttamente con il driver MySQL Connector/J della community e il driver MariadB Connector/J della community.

Per installare il driver AWS JDBC, aggiungi il file.jar del driver AWS JDBC (che si trova nell'applicazione) e mantieni i riferimenti al rispettivo driver della community. `CLASSPATH` Aggiorna il rispettivo prefisso dell'URL di connessione come segue:

- `jdbc:mysql://` Da a `jdbc:aws-wrapper:mysql://`
- `jdbc:mariadb://` Da a `jdbc:aws-wrapper:mariadb://`

Per ulteriori informazioni sul driver AWS JDBC e istruzioni complete per il suo utilizzo, consulta l'archivio dei driver JDBC di Amazon Web Services (AWS). GitHub

Connessione a RDS per MariaDB con il driver Python Amazon Web Services ()AWS

Il driver Python di Amazon Web Services (AWS) è progettato come wrapper Python avanzato. Questo wrapper è complementare ed estende le funzionalità del driver open source Psycopg. Il AWS Python Driver supporta le versioni Python 3.8 e successive. È possibile installare il `aws-advanced-python-wrapper` pacchetto utilizzando il `pip` comando, insieme ai pacchetti open source. `psycopg`

Per ulteriori informazioni sul driver AWS Python e istruzioni complete per il suo utilizzo, consulta il repository Amazon Web Services ()AWS Python Driver. GitHub

Risoluzione dei problemi relativi alle connessioni all'istanza database MariaDB

Di seguito sono indicate due cause comuni degli errori di connessione a una nuova istanza database:

- L'istanza database è stata creata utilizzando un gruppo di sicurezza che non autorizza le connessioni dal dispositivo o dall'istanza Amazon EC2 in cui è in esecuzione l'applicazione o l'utilità MariaDB. L'istanza database deve disporre di un gruppo di sicurezza VPC che autorizzi le connessioni. Per ulteriori informazioni, consulta VPC di Amazon VPC e Amazon RDS.

  Puoi aggiungere o modificare una regola in entrata nel gruppo di sicurezza: per Source (Origine), scegli My IP (Il mio IP). Questo consente l'accesso all'istanza database dall'indirizzo IP rilevato nel browser.
- L'istanza database è stata creata utilizzando la porta predefinita 3306 e nell'azienda vi sono regole del firewall che bloccano le connessioni a tale porta dai dispositivi nella rete aziendale. Per correggere l'errore, ricrea l'istanza con una porta diversa.

Per ulteriori informazioni sui problemi di connessione, consulta Impossibile connettersi all'istanza database di Amazon RDS.

Protezione delle connessioni di istanze database MariaDB

Puoi gestire la sicurezza delle istanze database MariaDB.

Argomenti

Sicurezza di MariaDB in Amazon RDS

La sicurezza delle istanze database MariaDB è gestita su tre livelli:

- AWS Identity and Access Management controlla chi è in grado di eseguire le operazioni di gestione Amazon RDS nelle istanze database. Quando esegui la connessione ad AWS usando le credenziali IAM, il tuo account IAM deve disporre di policy IAM per la concessione delle autorizzazioni richieste per eseguire le operazioni di gestione di Amazon RDS. Per ulteriori informazioni, consultare Gestione accessi e identità per Amazon RDS.
- Quando crei un'istanza database, utilizzi un gruppo di sicurezza VPC per controllare i dispositivi e le istanze Amazon EC2 che possono aprire le connessioni all'endpoint e alla porta dell'istanza database. Queste connessioni possono essere stabilite tramite Secure Socket Layer (SSL) e Transport Layer Security (TLS). Le regole del firewall aziendale possono inoltre determinare se i dispositivi in esecuzione nell'azienda possono aprire connessioni all'istanza database.
- Dopo la creazione di una connessione a un'istanza database MariaDB, l'autenticazione del login e le autorizzazioni sono applicate come in un'istanza autonoma di MariaDB. I comandi come `CREATE USER`, `RENAME USER`, `GRANT`, `REVOKE` e `SET PASSWORD` funzionano esattamente come nei database autonomi, come avviene per la modifica diretta delle tabelle dello schema del database.

Quando crei un'istanza database Amazon RDS, l'utente master ha i seguenti privilegi predefiniti:

- `alter`
- `alter routine`
- `create`
- `create routine`

- `create temporary tables`
- `create user`
- `create view`
- `delete`
- `drop`
- `event`
- `execute`
- `grant option`
- `index`
- `insert`
- `lock tables`
- `process`
- `references`
- `reload`

  Questo privilegio è limitato sulle istanze database di MariaDB. Non concede l'accesso alle operazioni FLUSH LOGS o FLUSH TABLES WITH READ LOCK.
- `replication client`
- `replication slave`
- `select`
- `show databases`
- `show view`
- `trigger`
- `update`

Per ulteriori informazioni su questi privilegi, consulta User Account Management nella documentazione di MariaDB.

> **Note**
>
> Sebbene sia possibile eliminare l'utente master in un'istanza database, consigliamo di non farlo. Per ricreare l'utente master, utilizza l'API `ModifyDBInstance` o lo `modify-db-instance` AWS CLIe specifica una nuova password utente master con il parametro

appropriato. Se l'utente master non è presente nell'istanza, viene creato con la password specificata.

Per fornire servizi di gestione per ogni istanza database, viene creato l'utente `rdsadmin` al momento della creazione dell'istanza database. I tentativi di rimuovere, rinominare, cambiare la password o modificare i privilegi dell'account `rdsadmin` produrranno errori.

Per consentire la gestione dell'istanza database, i comandi standard `kill` e `kill_query` sono stati limitati. I comandi Amazon RDS `mysql.rds_kill`, `mysql.rds_kill_query` e `mysql.rds_kill_query_id`, vengono forniti per l'uso in MariaDB e anche in MySQL, per permettere di terminare le query o le sessioni utente nelle istanze database.

Crittografia delle connessioni client alle istanze database MariaDB con SSL/TLS

Secure Sockets Layer (SSL) è un protocollo standard del settore utilizzato per proteggere connessioni di rete tra client e server. Dopo SSL versione 3.0, il nome è stato modificato in Transport Layer Security (TLS). Amazon RDS supporta la crittografia SSL/TLS per le istanze database MariaDB. Mediante SSL/TLS, puoi crittografare una connessione tra il client dell'applicazione e l'istanza database MariaDB. Il supporto SSL/TLS è disponibile in tutti. Regioni AWS

Argomenti

- Utilizzo di SSL/TLS con un'istanza database MariaDB
- Richiesta di SSL/TLS per tutte le connessioni a un'istanza database MariaDB
- Connessione dal client a riga di comando MySQL con SSL/TLS (crittografato)

Utilizzo di SSL/TLS con un'istanza database MariaDB

Amazon RDS crea un certificato SSL/TLS e installa il certificato nell'istanza database quando Amazon RDS effettua il provisioning dell'istanza. Questi certificati sono firmati da un'autorità di certificazione. Il certificato SSL/TLS include l'endpoint dell'istanza database come nome comune (CN) per il certificato SSL/TLS per la protezione contro attacchi di spoofing.

Un certificato SSL/TLS creato da Amazon RDS è l'entità root attendibile e funziona nella maggior parte dei casi, ma potrebbe non funzionare se l'applicazione non accetta catene di certificati. Se l'applicazione non accetta le catene di certificati, potrebbe essere necessario utilizzare un certificato

intermedio per la connessione alla Regione AWS. Ad esempio, è necessario utilizzare un certificato intermedio per connettersi alle regioni utilizzando SSL/TLS. AWS GovCloud (US)

Per ulteriori informazioni sul download dei certificati, consultare . Per ulteriori informazioni sull'uso di SSL/TLS con MySQL, consulta Aggiornamento delle applicazioni per la connessione a istanze database MariaDB mediante i nuovi certificati SSL/TLS.

Amazon RDS for MariaDB supporta le versioni Transport Layer Security (TLS) 1.3, 1.2, 1.1 e 1.0. Il supporto TLS dipende dalla versione secondaria di MariadB. La tabella seguente mostra il supporto TLS per le versioni minori di MariadB.

Versione TLS	MariaDB 10.11	MariaDB 10.6	MariaDB 10.5	MariaDB 10.4
TLS 1.3	Tutte le versioni minori	Tutte le versioni secondarie	Tutte le versioni secondarie	Tutte le versioni secondarie
TLS 1.2	Tutte le versioni secondarie	Tutte le versioni secondarie	Tutte le versioni secondarie	Tutte le versioni secondarie
TLS 1.1	10.11.6 e versioni precedenti	10.6.16 e versioni precedenti	10.5.23 e versioni precedenti	10.4.32 e versioni precedenti
TLS 1.0	10.11.6 e versioni precedenti	10.6.16 e versioni precedenti	10.5.23 e versioni precedenti	10.4.32 e versioni precedenti

Puoi richiedere le connessioni SSL/TLS per account utente specifici. Ad esempio, in base alla versione di MariaDB, puoi usare una delle seguenti istruzioni per richiedere connessioni SSL/TLS sull'account utente `encrypted_user`.

Utilizza la seguente istruzione.

```
ALTER USER 'encrypted_user'@'%' REQUIRE SSL;
```

Per ulteriori informazioni sulle connessioni SSL/TLS con MariaDB, consulta la pagina relativa alla protezione delle connessioni per client e servernella documentazione di MariaDB.

Richiesta di SSL/TLS per tutte le connessioni a un'istanza database MariaDB

Utilizza il parametro `require_secure_transport` per richiedere che tutte le connessioni utente all'istanza database MariaDB utilizzino SSL/TLS. Per impostazione predefinita, il parametro `require_secure_transport` è impostato su `OFF`. Puoi impostare il parametro `require_secure_transport` su `ON` per richiedere la crittografia SSL/TLS per le connessioni all'istanza database.

> **Note**
>
> Il parametro `require_secure_transport` è supportato solo per MariaDB versione 10.5 e successive.

Puoi impostare il valore del parametro `require_secure_transport` aggiornando il gruppo di parametri database per l'istanza database. Non è necessario riavviare l'istanza database affinché la modifica abbia effetto.

Quando il parametro `require_secure_transport` è impostato su `ON` per un'istanza database, un client di database può connettersi a essa se è in grado di stabilire una connessione crittografata. In caso contrario, viene restituito al client un messaggio di errore simile al seguente:

```
ERROR 1045 (28000): Access denied for user 'USER'@'localhost' (using password: YES |
 NO)
```

Per informazioni sull'estensione dei parametri consulta Modifica di parametri in un gruppo di parametri del database.

Per ulteriori informazioni sul parametro `require_secure_transport`, consulta la documentazione di MariaDB.

Connessione dal client a riga di comando MySQL con SSL/TLS (crittografato)

I parametri del programma client `mysql` sono leggermente diversi se si utilizza la versione MySQL 5.7, la versione MySQL 8.0 o la versione MariaDB.

Per scoprire quale versione è disponibile, esegui il comando `mysql` con l'opzione `--version`. Nell'esempio seguente, nell'output viene mostrato che il programma client proviene da MariaDB.

```
$ mysql --version
```

```
mysql  Ver 15.1 Distrib 10.5.15-MariaDB, for osx10.15 (x86_64) using readline 5.1
```

La maggior parte delle distribuzioni Linux, come Amazon Linux, CentOS, SUSE e Debian, hanno sostituito MySQL con MariaDB e la versione `mysql` presente proviene da MariaDB.

Per eseguire la connessione all'istanza database utilizzando SSL/TLS, segui questi passaggi:

Per eseguire la connessione a un'istanza database con SSL/TLS utilizzando il client a riga di comando MySQL

1. Scarica un certificato root che funzioni per tutti. Regioni AWS

   Per ulteriori informazioni sul download dei certificati, consultare .

2. Per stabilire la connessione a un'istanza database con la crittografia SSL/TLS, utilizza il client a riga di comando MySQL Per il parametro `-h`, sostituisci il nome DNS (endpoint) per l'istanza database. Per il parametro `--ssl-ca`, sostituisci il nome file del certificato SSL/TLS. Per il parametro `-P`, sostituisci la porta per l'istanza database. Per il parametro `-u`, sostituisci il nome utente di un utente di database valido, ad esempio l'utente master. Immetti la password dell'utente master quando richiesto.

   L'esempio seguente mostra come avviare il client utilizzando il parametro `--ssl-ca` con il client MariaDB.

   ```
   mysql -h mysql-instance1.123456789012.us-east-1.rds.amazonaws.com --ssl-ca=global-bundle.pem --ssl -P 3306 -u myadmin -p
   ```

   Per richiedere alla connessione SSL/TLS di verificare l'endpoint dell'istanza database confrontandolo con l'endpoint nel certificato SSL/TLS, immetti il seguente comando:

   ```
   mysql -h mysql-instance1.123456789012.us-east-1.rds.amazonaws.com --ssl-ca=global-bundle.pem --ssl-verify-server-cert -P 3306 -u myadmin -p
   ```

   L'esempio seguente mostra come avviare il client utilizzando il parametro `--ssl-ca` per il client MySQL 5.7 o versioni successive.

   ```
   mysql -h mysql-instance1.123456789012.us-east-1.rds.amazonaws.com --ssl-ca=global-bundle.pem --ssl-mode=REQUIRED -P 3306 -u myadmin -p
   ```

3. Immetti la password dell'utente master quando richiesto.

Verrà visualizzato un output simile al seguente.

```
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 31
Server version: 10.6.10-MariaDB-log Source distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]>
```

Aggiornamento delle applicazioni per la connessione a istanze database MariaDB mediante i nuovi certificati SSL/TLS

A partire dal 13 gennaio 2023, Amazon RDS ha pubblicato nuovi certificati dell'autorità di certificazione (CA) per la connessione alle istanze database RDS utilizzando Secure Socket Layer o Transport Layer Security (SSL/TLS). Di seguito sono disponibili le informazioni sull'aggiornamento delle applicazioni per utilizzare i nuovi certificati.

Questo argomento aiuta a determinare se le applicazioni richiedono la verifica dei certificati per la connessione delle istanze database.

> **Note**
>
> Alcune applicazioni sono configurate per connettersi a MariaDB sono se possono correttamente verificare il certificato sul server. Per queste applicazioni, è necessario aggiornare gli archivi di trust delle applicazioni client per includere i nuovi certificati CA.
> Puoi specificare le seguenti modalità SSL: `disabled`, `preferred` e `required`. Quando si utilizza la modalità `preferred` SSL e il certificato CA non esiste o non è aggiornato, la connessione non utilizza SSL e continua a connettersi correttamente.
> Consigliamo di evitare la modalità `preferred`. In modalità `preferred`, se la connessione rileva un certificato non valido, interrompe l'utilizzo della crittografia e procede in modo non crittografato.

Dopo aver aggiornato i certificati CA negli archivi di trust delle applicazioni client, puoi ruotare i certificati nelle istanze database. Consigliamo vivamente di testare queste procedure in un ambiente di sviluppo o di gestione temporanea prima di implementarle negli ambienti di produzione.

Per ulteriori informazioni sulla rotazione dei certificati, consulta Rotazione del certificato SSL/TLS. Per ulteriori informazioni sul download, consulta . Per informazioni sull'utilizzo di SSL/TLS con le istanze database MariaDB, consulta Utilizzo di SSL/TLS con un'istanza database MariaDB.

Argomenti

- Determinare se un client richiede la verifica del certificato per la connessione
- Aggiornare l'archivio di trust delle applicazioni
- Codice Java di esempio per stabilire connessioni SSL

Determinare se un client richiede la verifica del certificato per la connessione

Puoi verificare se i client JDBC e MySQL richiedono la verifica del certificato per la connessione.

JDBC

L'esempio seguente con MySQL Connector/J 8.0 mostra un modo per verificare le proprietà della connessione JDBC di un'applicazione per determinare se le connessioni riuscite richiedono un certificato valido. Per ulteriori informazioni su tutte le opzioni di connessione JDBC per MySQL, consulta l'argomento relativo alle proprietà di configurazione nella documentazione di MySQL.

Quando utilizzi MySQL Connector/J 8.0, la connessione SSL richiede la verifica del certificato CA del server se nelle proprietà di connessione `sslMode` è impostato su `VERIFY_CA` o `VERIFY_IDENTITY`, come nell'esempio seguente.

```
Properties properties = new Properties();
properties.setProperty("sslMode", "VERIFY_IDENTITY");
properties.put("user", DB_USER);
properties.put("password", DB_PASSWORD);
```

> **Note**
>
> Se si utilizza MySQL Java Connector v5.1.38 o versione successiva o MySQL Java Connector v8.0.9 o versione successiva per connettersi ai database, anche se non sono state configurate esplicitamente le applicazioni per l'utilizzo di SSL/TLS durante la connessione ai database, questi driver client utilizzano automaticamente SSL/TLS. Inoltre, quando utilizzano SSL/TLS, eseguono la verifica parziale del certificato e non riescono a connettersi se il certificato del server di database è scaduto.

Specifica una password diversa dal prompt mostrato qui come best practice per la sicurezza.

MySQL

I seguenti esempi con il client MySQL mostrano due modi per verificare la connessione MySQL di uno script per determinare se le connessioni riuscite richiedono un certificato valido. Per ulteriori informazioni su tutte le opzioni di connessione con il client MySQL, consulta l'argomento relativo alla configurazione lato client delle connessioni crittografate nella documentazione di MySQL.

Quando utilizzi il client MySQL 5.7 o MySQL 8.0, la connessione SSL richiede la verifica del certificato CA del server se per l'opzione `--ssl-mode` viene specificato `VERIFY_CA` o `VERIFY_IDENTITY`, come nell'esempio seguente.

```
mysql -h mysql-database.rds.amazonaws.com -uadmin -ppassword --ssl-ca=/tmp/ssl-cert.pem
 --ssl-mode=VERIFY_CA
```

Quando utilizzi il client MySQL 5.6, la connessione SSL richiede la verifica del certificato CA del server se viene specificata l'opzione `--ssl-verify-server-cert`, come nell'esempio seguente.

```
mysql -h mysql-database.rds.amazonaws.com -uadmin -ppassword --ssl-ca=/tmp/ssl-cert.pem
 --ssl-verify-server-cert
```

Aggiornare l'archivio di trust delle applicazioni

Per informazioni relative all'archivio di trust per le applicazioni MySQL, consultare l'argomento relativo all'utilizzo di TLS/SSL con MariaDB Connector/J nella documentazione di MariaDB.

Per ulteriori informazioni sul download del certificato root, consulta .

Per gli script di esempio che importano i certificati, consulta Script di esempio per l'importazione di certificati nel tuo archivio di trust.

> **Note**
>
> Quando aggiorni l'archivio di trust puoi conservare i certificati meno recenti oltre ad aggiungere i nuovi certificati.

Se utilizzi il driver JDBC MariaDB Connector/J in un'applicazione, imposta le seguenti proprietà nell'applicazione.

```
System.setProperty("javax.net.ssl.trustStore", certs);
System.setProperty("javax.net.ssl.trustStorePassword", "password");
```

Quando avvii l'applicazione, imposta le seguenti proprietà.

```
java -Djavax.net.ssl.trustStore=/path_to_truststore/MyTruststore.jks -
Djavax.net.ssl.trustStorePassword=my_truststore_password com.companyName.MyApplication
```

> **Note**
>
> Specifica password diverse dalle istruzioni mostrate qui come best practice di sicurezza.

Codice Java di esempio per stabilire connessioni SSL

L'esempio di codice seguente mostra come impostare la connessione SSL utilizzando JDBC.

```
private static final String DB_USER = "admin";

        private static final String DB_USER = "user name";
        private static final String DB_PASSWORD = "password";
        // This key store has only the prod root ca.
        private static final String KEY_STORE_FILE_PATH = "file-path-to-keystore";
        private static final String KEY_STORE_PASS = "keystore-password";
```

```
    public static void main(String[] args) throws Exception {
        Class.forName("org.mariadb.jdbc.Driver");

        System.setProperty("javax.net.ssl.trustStore", KEY_STORE_FILE_PATH);
        System.setProperty("javax.net.ssl.trustStorePassword", KEY_STORE_PASS);

        Properties properties = new Properties();
        properties.put("user", DB_USER);
        properties.put("password", DB_PASSWORD);


        Connection connection = DriverManager.getConnection("jdbc:mysql://ssl-mariadb-
public.cni62e2e7kwh.us-east-1.rds.amazonaws.com:3306?useSSL=true",properties);
        Statement stmt=connection.createStatement();

        ResultSet rs=stmt.executeQuery("SELECT 1 from dual");

        return;
    }
```

Important

Dopo aver stabilito che le connessioni al database utilizzano SSL/TLS e aver aggiornato l'archivio attendibile dell'applicazione, è possibile aggiornare il database per utilizzare i certificati 2048-g1. rds-ca-rsa Per istruzioni, consulta la fase 3 in Aggiornamento del certificato CA modificando l'istanza o il cluster di database.

Specifica una password diversa dal prompt mostrato qui come best practice per la sicurezza.

Prestazioni delle query migliorate per RDS per MariaDB con Amazon RDS Optimized Reads

Puoi ottenere un'elaborazione delle query per RDS per MariaDB più rapida con Amazon RDS Optimized Reads. Un'istanza database RDS per MariaDB che utilizza RDS Optimized Reads può ottenere un'elaborazione delle query fino a due volte più veloce rispetto a un'istanza database che non lo utilizza.

Argomenti

Panoramica di RDS Optimized Reads

Quando si utilizza un'istanza database RDS per MariaDB con RDS Optimized Reads attivato, l'istanza database ottiene prestazioni di query più rapide tramite l'uso di un archivio dell'istanza. Un archivio istanze fornisce uno storage temporaneo di livello per l'istanza database. L'archiviazione è basata su unità di memoria a stato solido (SSD) NVMe (Non-Volatile Memory Express) fisicamente collegata al server host. Questa archiviazione è ottimizzata per bassa latenza, prestazioni I/O casuali elevate e velocità di trasmissione effettiva di lettura sequenziale elevata.

RDS Optimized Reads è attivato per impostazione predefinita quando un'istanza database utilizza una classe di istanza database con un archivio dell'istanza, ad esempio db.m5d o db.m6gd. Con RDS Optimized Reads, alcuni oggetti temporanei vengono archiviati nell'archivio dell'istanza. Questi oggetti temporanei includono file temporanei interni, tabelle temporanee interne su disco, file di mappe in memoria e file di cache di log binario (binlog). Per ulteriori informazioni sull'archivio dell'istanza, consulta Instance store Amazon EC2 nella Guida per l'utente di Amazon Elastic Compute Cloud per istanze Linux.

I carichi di lavoro che generano gli oggetti temporanei in MariaDB per l'elaborazione delle query possono sfruttare l'archivio dell'istanza per elaborare più rapidamente le query. Questo tipo di

carico di lavoro include query che coinvolgono ordinamenti, aggregazioni di hash, join a carico elevato, espressioni di tabella comuni (CTE) e query su colonne non indicizzate. Questi volumi dell'archivio dell'istanza forniscono operazioni IOPS e prestazioni più elevate, indipendentemente dalle configurazioni utilizzate per l'archivio persistente di Amazon EBS. Poiché RDS Optimized Reads trasferisce le operazioni sugli oggetti temporanei all'archivio dell'istanza, le operazioni di input/output al secondo (IOPS) o la velocità di trasmissione effettiva dell'archivio persistente (Amazon EBS) possono ora essere utilizzate per le operazioni su oggetti persistenti. Queste includono le normali operazioni di lettura e scrittura dei file di dati e le operazioni del motore in background, come lo svuotamento e l'unione di inserimenti di buffer.

> **Note**
>
> Gli snapshot RDS manuali e automatici contengono solo i file del motore per gli oggetti persistenti. Gli oggetti temporanei creati nell'archivio dell'istanza non sono inclusi negli snapshot RDS.

Casi d'uso per RDS Optimized Reads

Se hai carichi di lavoro che si basano pesantemente sugli oggetti temporanei, come tabelle o file interni, per l'esecuzione delle query, puoi trarre vantaggio dall'attivazione di RDS Optimized Reads. I seguenti casi d'uso sono candidati per RDS Optimized Reads:

- Applicazioni che eseguono query analitiche con espressioni di tabella comuni (CTE) complesse, tabelle derivate e operazioni di raggruppamento
- Repliche di lettura che generano un intenso traffico di lettura con query non ottimizzate
- Applicazioni che eseguono query di report on demand o dinamiche che includono operazioni complesse, ad esempio query con le clausole `GROUP BY` e `ORDER BY`
- Carichi di lavoro che utilizzano tabelle temporanee interne per l'elaborazione delle query

  È possibile monitorare la variabile di stato del motore `created_tmp_disk_tables` per determinare il numero di tabelle temporanee basate su disco create nell'istanza database.
- Applicazioni che creano tabelle temporanee di grandi dimensioni, direttamente o tramite procedure, per archiviare risultati intermedi
- Query di database che eseguono il raggruppamento o l'ordinamento di colonne non indicizzate

Best practice per RDS Optimized Reads

Usa le seguenti best practice per RDS Optimized Reads:

- Aggiungi la logica dei tentativi per le query di sola lettura, nel caso in cui non riescano perché l'archivio dell'istanza è completo durante l'esecuzione.
- Monitora lo spazio di archiviazione disponibile sull'instance store con la CloudWatch metrica`FreeLocalStorage`. Se l'archivio dell'istanza sta raggiungendo il limite a causa del carico di lavoro dell'istanza database, modifica l'istanza database in modo da utilizzare una classe di istanza database più grande.
- Se l'istanza database ha la memoria sufficiente ma raggiunge comunque il limite di archiviazione dell'archivio dell'istanza, aumenta il valore `binlog_cache_size` per mantenere in memoria le voci binlog specifiche della sessione. Questa configurazione impedisce di scrivere le voci binlog in file di cache binlog temporanei memorizzati su disco.

  Il parametro `binlog_cache_size` è specifico della sessione. É possibile modificare il valore per ogni nuova sessione. L'impostazione di questo parametro può aumentare l'utilizzo della memoria dell'istanza database durante i picchi di carico di lavoro. Pertanto, è consigliabile aumentare il valore del parametro in base al modello di carico di lavoro dell'applicazione e alla memoria disponibile nell'istanza database.
- Usa il valore predefinito `MIXED` per `binlog_format`. A seconda della dimensione delle transazioni, l'impostazione `binlog_format` su `ROW` può comportare la creazione di file di cache binlog di grandi dimensioni nell'archivio dell'istanza.
- Evita di apportare modifiche in blocco in una singola transazione. Questi tipi di transazioni possono generare file di cache binlog di grandi dimensioni nell'archivio dell'istanza e possono causare problemi quando l'archivio dell'istanza è pieno. Prendi in considerazione la suddivisione delle scritture in transazioni più piccole per ridurre al minimo l'uso dello spazio di archiviazione per i file di cache binlog.

Utilizzo di RDS Optimized Reads

L'istanza database utilizza automaticamente la funzionalità Letture ottimizzate per Amazon RDS quando in un'implementazione di istanza database single-AZ o multi-AZ, effettui il provisioning di un'istanza database RDS per MariaDB con una delle seguenti classi di istanza database.

Per attivare RDS Optimized Reads, procedi in uno dei seguenti modi:

- Crea un'istanza database RDS per MariaDB utilizzando una di queste classi di istanza database. Per ulteriori informazioni, consulta Creazione di un'istanza database Amazon RDS.
- Modifica un'istanza database RDS per MariaDB esistente per utilizzare una di queste classi di istanza database. Per ulteriori informazioni, consulta Modifica di un'istanza database Amazon RDS.

RDS Optimized Reads è disponibile Regioni AWS ovunque siano supportate una o più classi di istanze DB con storage SSD NVMe locale. Per informazioni sulle classi di istanza database, consulta the section called “Classi di istanze database”.

La disponibilità delle classi di istanze DB è diversa per. Regioni AWS Per determinare se una classe di istanza DB è supportata in una determinata istanza Regione AWS, consultathe section called “Determinazione del supporto delle classi di istanze DB in Regioni AWS”.

Se non desideri utilizzare RDS Optimized Reads, modifica l'istanza database in modo che non utilizzi una classe di istanza database che supporti la funzionalità.

Monitoraggio delle istanze database che utilizzano RDS Optimized Reads

È possibile monitorare le istanze DB che utilizzano RDS Optimized Reads con le seguenti metriche: CloudWatch

- `FreeLocalStorage`
- `ReadIOPSLocalStorage`
- `ReadLatencyLocalStorage`
- `ReadThroughputLocalStorage`
- `WriteIOPSLocalStorage`
- `WriteLatencyLocalStorage`
- `WriteThroughputLocalStorage`

Queste metriche forniscono dati sullo spazio di archiviazione dell'archivio dell'istanza, sulle operazioni IOPS e sulla velocità di trasmissione effettiva disponibili. Per ulteriori informazioni su questi parametri, consulta Parametri a CloudWatch livello di istanza Amazon per Amazon RDS.

Limitazioni per RDS Optimized Reads

Le seguenti limitazioni si applicano a RDS Optimized Reads:

- RDS Optimized Reads è supportato nelle seguenti versioni di RDS per MariaDB:
  - 10.11.4 e versioni successive alla 10.11
  - 10.6.7 e versioni successive alla 10.6
  - 10.5.16 e versioni successive alla 10.5
  - 10.4.25 e versioni successive alla 10.4

  Per ulteriori informazioni sulle versioni di RDS per MariaDB, consulta Versioni di MariaDB in Amazon RDS.
- Non è possibile modificare la posizione degli oggetti temporanei nell'archivio persistente (Amazon EBS) nelle classi di istanza database che supportano RDS Optimized Reads.
- Quando i log binari sono abilitati su un'istanza database, la dimensione massima della transazione è limitata alla dimensione dell'archivio dell'istanza. In MariaDB, qualsiasi sessione che richiede più spazio di archiviazione rispetto al valore `binlog_cache_size` scrive le modifiche della transazione nei file di cache binlog temporanei, che vengono creati nell'archivio dell'istanza.
- Le transazioni possono non riuscire quando l'archivio dell'istanza è pieno.

Prestazioni di scrittura migliorate con Scritture ottimizzate per Amazon RDS per MariaDB

Puoi migliorare le prestazioni delle transazioni di scrittura con Scritture ottimizzate per RDS per MariaDB. Quando il database RDS per MariaDB utilizza Scritture ottimizzate per Amazon RDS, può raggiungere una velocità di trasmissione effettiva delle transazioni di scrittura fino a due volte superiore.

Argomenti

Panoramica di RDS Optimized Writes

Quando attivi Scritture ottimizzate per RDS, i database RDS per MariaDB scrivono solo una volta, quando trasferiscono i dati nell'archiviazione durevole senza la necessità del buffer di doppia scrittura. I database continuano a fornire le protezioni delle proprietà ACID per le transazioni di database affidabili, insieme alle prestazioni migliorate.

I database relazionali, come MariaDB, forniscono le proprietà ACID di atomicità, consistenza, isolamento e durabilità per le transazioni di database affidabili. Per fornire queste proprietà, MariaDB utilizza un'area di archiviazione di dati chiamata buffer di doppia scrittura che impedisce gli errori di scrittura parziale della pagina. Questi errori si verificano nel caso di un guasto hardware mentre il database sta aggiornando una pagina, ad esempio in caso di interruzione dell'alimentazione. Un database MariaDB può rilevare le scritture parziali della pagina e recuperarle con una copia della pagina nel buffer di doppia scrittura. Sebbene questa tecnica fornisca protezione, comporta anche operazioni di scrittura aggiuntive. Per ulteriori informazioni sul buffer di doppia scrittura MariaDB, consulta l'argomento relativo al buffer di doppia scrittura nella documentazione di MySQL.

Quando attivi Scritture ottimizzate per Amazon RDS, i database RDS per MariaDB scrivono una sola volta, quando trasferiscono i dati nell'archiviazione durevole senza usare il buffer di doppia scrittura. Scritture ottimizzate per Amazon RDS è utile se esegui carichi di lavoro intensivi in scrittura sui database RDS per MariaDB. Esempi di database con carichi di lavoro intensivi in scrittura includono quelli che supportano pagamenti digitali, trading finanziario e applicazioni di gioco.

Questi database vengono eseguiti in classi di istanza database che utilizzano AWS Nitro System. Grazie alla configurazione hardware di questi sistemi, il database può scrivere pagine da 16 KiB direttamente su file di dati in modo affidabile e durevole in un solo passaggio. AWS Nitro System permette di usare RDS Optimized Writes.

Puoi impostare il nuovo parametro di database `rds.optimized_writes` per controllare la funzionalità Scritture ottimizzate per Amazon RDS per i database RDS per MariaDB. Accedi a questo parametro nei gruppi di parametri database RDS per MariaDB aventi le seguenti versioni:

- 10.11.4 e versioni successive alla 10.11
- 10.6.10 e versioni successive alla 10.6

Imposta il parametro su uno dei seguenti valori:

- `AUTO` - Attiva RDS Optimized Writes se la funzionalità è supportata dal database. In caso contrario, disattiva RDS Optimized Writes. Questa è l'impostazione di default.
- `OFF` - Disattiva RDS Optimized Writes anche la funzionalità è supportata dal database.

Se esegui la migrazione di un database RDS per MariaDB configurato per utilizzare Scritture ottimizzate per Amazon RDS in una classe di istanza database che non supporta la funzionalità, RDS disattiva automaticamente Scritture ottimizzate per Amazon RDS per il database.

Quando la funzionalità Scritture ottimizzate per Amazon RDS è disattivata, il database utilizza il buffer di doppia scrittura MariaDB.

Per determinare se un database RDS per MySQL utilizza Scritture ottimizzate per Amazon RDS, osserva il valore corrente del parametro `innodb_doublewrite` per il database. Se il database utilizza RDS Optimized Writes, questo parametro è impostato su `FALSE` (`0`).

Utilizzo di RDS Optimized Writes

È possibile attivare Scritture ottimizzate per Amazon RDS quando si crea un database RDS per MariaDBcon la console RDS, la AWS CLI o l'API RDS. La funzionalità RDS Optimized Writes viene attivata automaticamente quando si verificano entrambe le seguenti condizioni durante la creazione del database:

- Si specificano una versione del motore di database e una classe di istanza database che supportano RDS Optimized Writes.

- La funzionalità Scritture ottimizzate per Amazon RDS è supportata nelle seguenti versioni di RDS per MariaDB:
  - 10.11.4 e versioni successive alla 10.11
  - 10.6.10 e versioni successive alla 10.6

  Per ulteriori informazioni sulle versioni di RDS per MariaDB, consulta Versioni di MariaDB in Amazon RDS.
- La funzionalità Scritture ottimizzate per Amazon RDS è supportata per i database RDS per MariaDB che utilizzano le seguenti classi di istanza database:
  - db.m7g
  - db.m6g
  - db.m6gd
  - db.m6i
  - db.m5
  - db.m5d
  - db.r7g
  - db.r6g
  - db.r6gd
  - db.r6i
  - db.r5
  - db.r5b
  - db.r5d
  - db.x2idn
  - db.x2iedn

  Per informazioni sulle classi di istanza database, consulta the section called “Classi di istanze database”.

  La disponibilità della classe di istanze DB è diversa nelle varie Regioni AWS. Per determinare se una classe di istanza DB è supportata in una Regione AWS specifica, consulta the section called “Determinazione del supporto delle classi di istanze DB in Regioni AWS”.
- Nel gruppo di parametri associato al database, il parametro `rds.optimized_writes` è impostato su AUTO. Nei gruppi di parametri predefiniti, questo parametro è sempre impostato su AUTO.

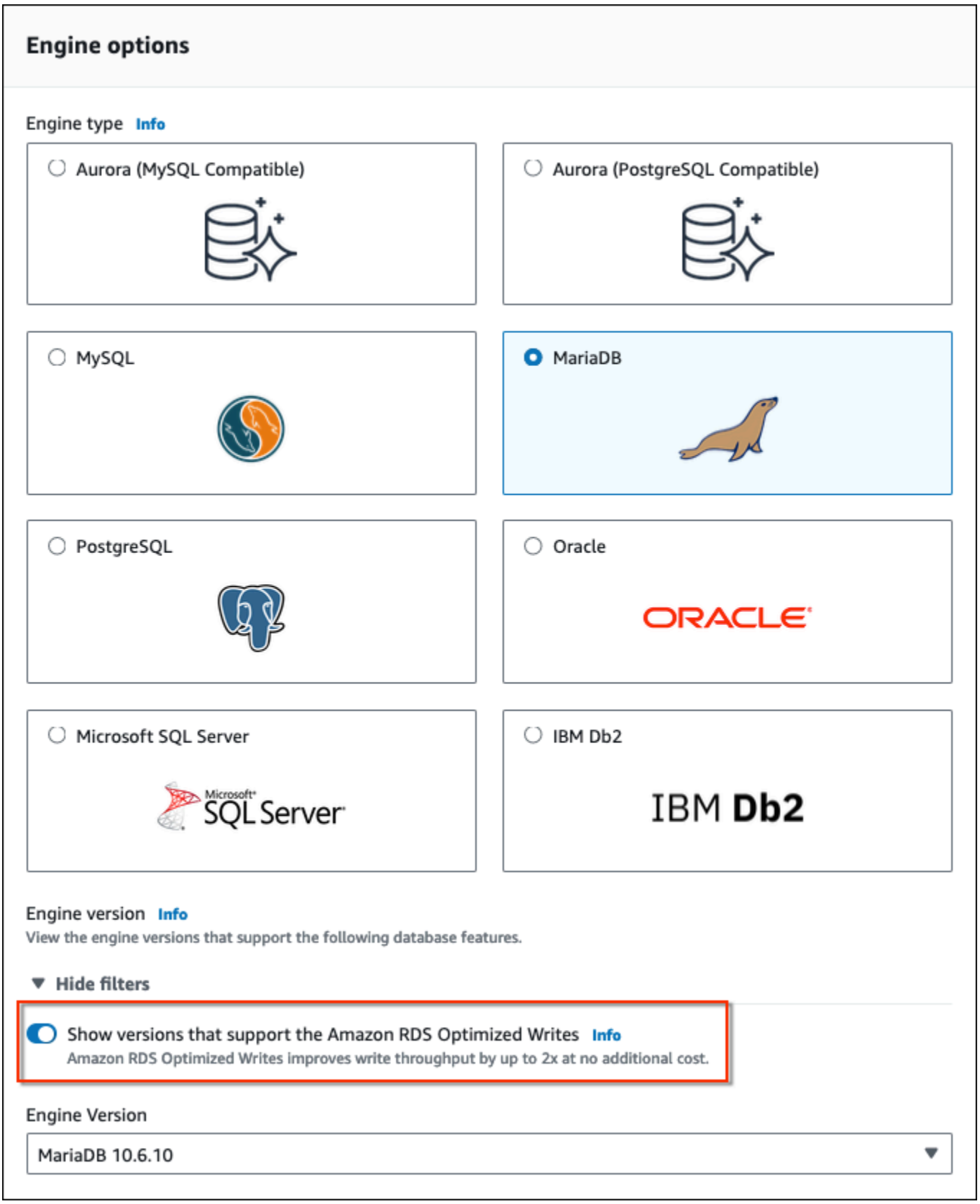
Se vuoi utilizzare una versione del motore di database e una classe di istanza database che supportino Scritture ottimizzate per Amazon RDS, senza usare questa funzionalità, specifica un gruppo di parametri personalizzato durante la creazione del database. In questo gruppo di parametri, imposta il parametro `rds.optimized_writes` su `OFF`. Se si desidera che il database utilizzi RDS Optimized Writes in un secondo momento, è possibile impostare il parametro su `AUTO` per attivarlo. Per informazioni sull'utilizzo dei gruppi di parametri personalizzati e sull'impostazione dei parametri, consulta Utilizzo di gruppi di parametri.

Per informazioni sulla creazione di un'istanza database, consulta Creazione di un'istanza database Amazon RDS.

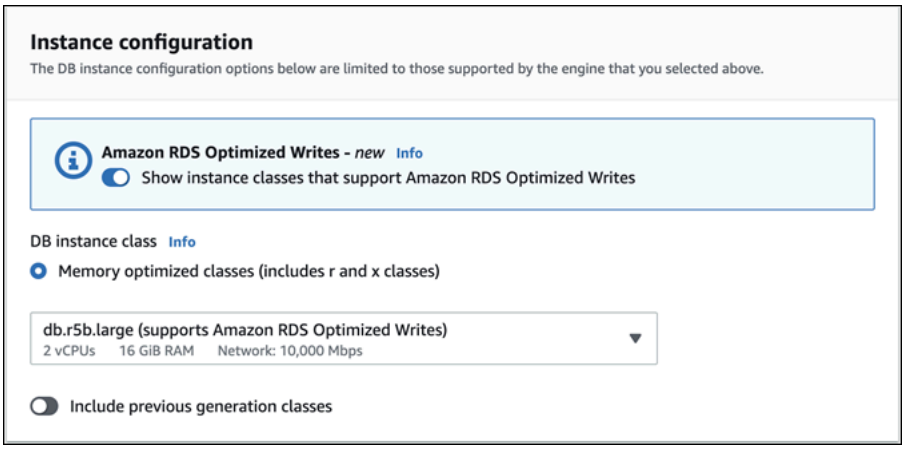
Console

Quando usi la console RDS per creare un database RDS per MariaDB, puoi filtrare le versioni del motore di database e le classi di istanza database che supportano Scritture ottimizzate per Amazon RDS. Dopo aver attivato i filtri, puoi scegliere tra le versioni del motore di database e le classi di istanza database disponibili.

Per scegliere una versione del motore di database che supporti Scritture ottimizzate per Amazon RDS, filtra le versioni del motore di database RDS per MariaDB che supportano la funzionalità in Versione motore, quindi scegli una versione.


Engine options
Engine type Info
Aurora (MySQL Compatible)
Aurora (PostgreSQL Compatible)
MySQL
MariaDB
PostgreSQL
Oracle
ORACLE
Microsoft SQL Server
Microsoft SQL Server
IBM Db2
IBM Db2
Engine version Info
View the engine versions that support the following database features.
Hide filters
Show versions that support the Amazon RDS Optimized Writes Info
Amazon RDS Optimized Writes improves write throughput by up to 2x at no additional cost.
Engine Version
MariaDB 10.6.10

Nella sezione Instance configuration (Configurazione dell'istanza), filtra le classi di istanza database che supportano RDS Optimized Writes, quindi scegli una classe di istanza database.



Dopo aver effettuato queste selezioni, puoi scegliere altre impostazioni che soddisfano i tuoi requisiti e completare la creazione del database RDS per MariaDB con la console.

AWS CLI

Per creare un'istanza DB utilizzando ilAWS CLI, utilizzare il create-db-instancecomando. Assicurati che i valori `--engine-version` e `--db-instance-class` supportino RDS Optimized Writes. Inoltre, assicurati che il gruppo di parametri associato all'istanza database abbia il parametro `rds.optimized_writes` impostato su AUTO. Questo esempio associa il gruppo di parametri predefinito all'istanza database.

Example Creazione di un'istanza database che utilizza RDS Optimized Writes

Per LinuxmacOS, oUnix:

```
aws rds create-db-instance \
    --db-instance-identifier mydbinstance \
    --engine mariadb \
    --engine-version 10.6.10 \
    --db-instance-class db.r5b.large \
    --manage-master-user-password \
    --master-username admin \
    --allocated-storage 200
```

Per Windows:

```
aws rds create-db-instance ^
```

```
    --db-instance-identifier mydbinstance ^
    --engine mariadb ^
    --engine-version 10.6.10 ^
    --db-instance-class db.r5b.large ^
    --manage-master-user-password ^
    --master-username admin ^
    --allocated-storage 200
```

API RDS

È possibile creare un'istanza database utilizzando l'operazione CreateDBInstance. Quando utilizzi questa operazione, assicurati che i valori `EngineVersion` e `DBInstanceClass` supportino RDS Optimized Writes. Inoltre, assicurati che il gruppo di parametri associato all'istanza database abbia il parametro `rds.optimized_writes` impostato su `AUTO`.

Abilitazione delle scritture ottimizzate per RDS in un database esistente

Per modificare un database RDS per MariaDB esistente per attivare Scritture ottimizzate per RDS, il database deve essere stato creato con una versione del motore di database e una classe di istanza database supportate. Inoltre, il database deve essere stato creato dopo il rilascio di Scritture ottimizzate per RDS del 7 marzo 2023, poiché la configurazione del file system sottostante richiesta è incompatibile con quella dei database creati prima del rilascio. Se queste condizioni sono soddisfatte, è possibile attivare Scritture ottimizzate per RDS impostando il parametro `rds.optimized_writes` su `AUTO`.

Se il database non è stato creato con una versione del motore, una classe di istanza o una configurazione del file system supportate, puoi utilizzare le implementazioni blu/verde di RDS per migrare a una configurazione supportata. Durante la creazione dell'implementazione blu/verde, esegui le seguenti operazioni:

- Seleziona Abilita le scritture ottimizzate sul database verde, quindi specifica una versione del motore e una classe di istanza database che supportano e scritture ottimizzate RDS. Per l'elenco delle versioni di motore e delle classi di istanza supportate, consulta the section called “Utilizzo con un nuovo database”.
- In Archiviazione scegli Aggiorna la configurazione del file system di archiviazione. Questa opzione aggiorna il database a una configurazione del file system sottostante compatibile.

Se quando crei l'implementazione blu/verde il parametro `rds.optimized_writes` è impostato su `AUTO`, Scritture ottimizzate per RDS viene abilitato automaticamente nell'ambiente verde. Quindi puoi

eseguire lo switchover all'implementazione blu/verde, che rende l'ambiente verde il nuovo ambiente di produzione.

Per ulteriori informazioni, consulta the section called “Creazione di un'implementazione blu/verde”.

Limitazioni per RDS Optimized Writes

Quando si ripristina un database RDS per MariaDB da uno snapshot, è possibile attivare Scritture ottimizzate per Amazon RDS per il database solo se si verificano tutte le seguenti condizioni:

- Lo snapshot è stato creato da un database che supporta RDS Optimized Writes.
- Lo snapshot è stato creato da un database creato dopo il rilascio della funzionalità Scritture ottimizzate per Amazon RDS.
- Lo snapshot è stato ripristinato in un database che supporta RDS Optimized Writes.
- Il database ripristinato è associato a un gruppo di parametri con il parametro `rds.optimized_writes` impostato su `AUTO`.

Aggiornamento del motore di database MariaDB

Quando Amazon RDS supporta una nuova versione di un motore del database, puoi effettuare l'aggiornamento delle istanze database alla nuova versione. Sono disponibili due tipi di aggiornamenti per le istanze database MariaDB: per la versione principale e per la versione secondaria.

Gli aggiornamenti di versione principali possono contenere modifiche al database non compatibili con le versioni precedenti delle applicazioni esistenti. Ne risulta che è necessario eseguire manualmente gli aggiornamenti della versione principale per le proprie istanze database. Puoi avviare manualmente un aggiornamento principale a una versione modificando l'istanza. Tuttavia, prima di eseguire un aggiornamento della versione principale, si consiglia di seguire le istruzioni presenti in Aggiornamenti di versione principale per MariaDB.

Al contrario, gli aggiornamento secondari a una versione includono solo modifiche compatibili con le versioni precedenti delle applicazioni esistenti. Puoi avviare un aggiornamento a una versione secondaria manualmente modificando la tua istanza database. In alternativa, è possibile abilitare l'opzione Auto minor version upgrade (Aggiornamenti automatico della versione secondaria) durante la creazione o la modifica di un'istanza database. Ciò significa che l'istanza database viene automaticamente aggiornata dopo che Amazon RDS testa e approva la nuova versione. Per informazioni sull'esecuzione di un aggiornamento, consulta Aggiornamento della versione del motore di un'istanza database.

Se la tua istanza database MariaDB sta utilizzando repliche di lettura, dovrai aggiornare tutte le repliche di lettura prima di aggiornare l'istanza di origine. Se la tua istanza database è in un'implementazione Multi-AZ, le repliche principali e le repliche standby vengono entrambe aggiornate. L'istanza database potrebbe non essere disponibile fino al completamento dell'aggiornamento.

Per ulteriori informazioni sulle versioni di MariaDB supportate e sulla gestione delle versioni, consulta Versioni di MariaDB in Amazon RDS.

Per gli aggiornamenti del motore di database si verificano tempi di inattività. La durata dell'interruzione varia in base alla dimensione dell'istanza database.

> **Tip**
>
> È possibile ridurre al minimo i tempi di inattività necessari per l'aggiornamento dell'istanza database utilizzando un'implementazione blu/verde. Per ulteriori informazioni, consulta Utilizzo delle implementazioni blu/verde Amazon RDS per gli aggiornamenti del database.

Argomenti

Panoramica dell'aggiornamento

Quando si utilizza AWS Management Console per aggiornare un'istanza DB, mostra gli obiettivi di aggiornamento validi per l'istanza DB. È inoltre possibile utilizzare il AWS CLI comando seguente per identificare gli obiettivi di aggiornamento validi per un'istanza DB:

Per LinuxmacOS, oUnix:

```
aws rds describe-db-engine-versions \
  --engine mariadb \
  --engine-version version-number \
  --query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" --
output text
```

Per Windows:

```
aws rds describe-db-engine-versions ^
  --engine mariadb ^
  --engine-version version-number ^
  --query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" --
output text
```

Ad esempio, per identificare gli obiettivi di aggiornamento validi per un'istanza DB MariaDB versione 10.5.17, esegui il seguente comando: AWS CLI

PerLinux, o: macOS Unix

```
aws rds describe-db-engine-versions \
```

```
  --engine mariadb \
  --engine-version 10.5.17 \
  --query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" --
output text
```

Per Windows:

```
aws rds describe-db-engine-versions ^
  --engine mariadb ^
  --engine-version 10.5.17 ^
  --query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" --
output text
```

Durante il processo di aggiornamento, Amazon RDS acquisisce due o più istantanee DB. Amazon RDS richiede fino a due istantanee dell'istanza database prima di apportare modifiche all'aggiornamento. Se l'aggiornamento non funziona per i database, puoi ripristinare una di queste istantanee per creare un'istanza database che esegue la versione precedente. Amazon RDS acquisisce un'altra istantanea dell'istanza database al termine dell'aggiornamento. Amazon RDS acquisisce queste istantanee indipendentemente dal fatto che AWS Backup gestisca o meno i backup per l'istanza DB.

> **Note**
>
> Amazon RDS acquisisce gli snapshot DB solo se hai impostato il periodo di retention dei backup per l'istanza database su un valore maggiore di 0. Per cambiare il periodo di retention dei backup, consulta Modifica di un'istanza database Amazon RDS.

Al termine dell'aggiornamento, non puoi ripristinare la versione precedente del motore di database. Se desideri tornare alla versione precedente, ripristina il primo snapshot DB acquisito per creare una nuova istanza database.

Puoi controllare quando eseguire l'aggiornamento dell'istanza database a una nuova versione supportata da Amazon RDS. Questo livello di controllo ti consente di mantenere la compatibilità con versioni di database specifiche e testare le nuove versioni con l'applicazione prima di distribuirle in produzione. Puoi aggiornare le versioni quando più appropriato in base alla tua pianificazione.

Se la tua istanza database utilizza una replica di lettura, devi aggiornare tutte le repliche di lettura prima di aggiornare l'istanza di origine.

Se l'istanza database è in un'implementazione Multi-AZ, vengono aggiornate sia l'istanza database principale che quella di standby. Le istanze database principali e standby vengono aggiornate contemporaneamente e si verificherà un'interruzione fino al completamento dell'aggiornamento. Il tempo di interruzione necessario varia in base a motore di database, versione del motore e dimensione dell'istanza database.

Numeri di versione di MariaDB

La sequenza di numerazione delle versioni per il motore di database RDS per MariadB è nella forma di major.minor.patch.yyyymmdd o major.minor.patch, ad esempio 10.11.5.R2.20231201 o 10.4.30. Il formato utilizzato dipende dalla versione del motore MariadB.

importante

Il numero di versione principale è sia il numero intero che la prima parte frazionaria del numero di versione, ad esempio 10.11. Un aggiornamento della versione principale incrementa la parte principale del numero di versione. Ad esempio, un aggiornamento da 10.5.20 a 10.6.12 è un aggiornamento della versione principale, dove 10.5 e 10.6 sono i numeri di versione principali.

minore

Il numero di versione secondario è la terza parte del numero di versione, ad esempio la 5 nella versione 10.11.5.

patch

La patch è la quarta parte del numero di versione, ad esempio la R2 in 10.11.5.R2. Una versione della patch di RDS include importanti correzioni di bug aggiunte a una versione secondaria dopo il rilascio.

YYYYMMGD

La data è la quinta parte del numero di versione, ad esempio 20231201 in 10.11.5.R2.20231201. Una versione con data RDS è una patch di sicurezza che include importanti correzioni di sicurezza aggiunte a una versione secondaria dopo il suo rilascio. Non include correzioni che potrebbero modificare il comportamento di un motore.

Versione principale	Versione secondaria	Schema di denominazione
10.11	≥ 5	Le nuove istanze DB utilizzano Major.min or.patch.YYMMDD, ad esempio 10.11.5.R 2.20231201. Le istanze DB esistenti potrebbero utilizzare major.minor.patch, ad esempio 10.11.5.R2, fino al prossimo aggiornamento della versione principale o secondaria.
	< 5	Le istanze DB esistenti utilizzano major.min or.patch, ad esempio 10.11.4.R2.
10.6	≥ 14	Le nuove istanze DB utilizzano Major.min or.patch.YYMMDD, ad esempio 10.6.14.R 2.20231201. Le istanze DB esistenti potrebbero utilizzare major.minor.patch, ad esempio 10.6.14.R2, fino al prossimo aggiornamento della versione principale o secondaria.
	< 14	Le istanze DB esistenti utilizzano major.min or.patch, ad esempio 10.6.13.R2.
10.5	≥ 21	Le nuove istanze DB utilizzano Major.min or.patch.YYMMDD, ad esempio 10.5.21.R 2.20231201. Le istanze DB esistenti potrebbero utilizzare major.minor.patch, ad esempio 10.5.21.R2, fino al prossimo aggiornamento della versione principale o secondaria.
	< 21	Le istanze DB esistenti utilizzano major.min or.patch, ad esempio 10.5.20.R2.

Versione principale	Versione secondaria	Schema di denominazione
10.4	≥ 30	Le nuove istanze DB utilizzano Major.min or.patch.YYMMDD, ad esempio 10.4.30.R 2.20231201. Le istanze DB esistenti potrebbero utilizzare major.minor.patch, ad esempio 10.4.30.R2, fino al prossimo aggiornamento della versione principale o secondaria.
	< 30	Le istanze DB esistenti utilizzano major.min or.patch, ad esempio 10.4.29.R2.

Numero di versione RDS

I numeri di versione RDS utilizzano lo schema di denominazione o lo schema di denominazione. *major*.*minor*.*patch* *major*.*minor*.*patch*.*YYYYMMDD* Una versione della patch di RDS include importanti correzioni di bug aggiunte a una versione secondaria dopo il rilascio. Una versione con data RDS (*YYMMDD*) è una patch di sicurezza. Una patch di sicurezza non include correzioni che potrebbero modificare il comportamento del motore.

Per identificare il numero di versione Amazon RDS del tuo database, è prima necessario creare l'estensione `rds_tools` utilizzando il seguente comando:

```
CREATE EXTENSION rds_tools;
```

Puoi scoprire il numero di versione RDS del tuo database RDS per MariaDB con la seguente query SQL:

```
mysql> select mysql.rds_version();
```

Ad esempio, l'interrogazione di un database RDS per MariaDB 10.6.14 restituisce il seguente output:

```
+---------------------+
| mysql.rds_version() |
+---------------------+
| 10.6.14.R2.20231201  |
```

```
+---------------------+
1 row in set (0.01 sec)
```

Aggiornamenti di versione principale per MariaDB

Gli aggiornamenti di versione principali possono contenere modifiche al database non compatibili con le versioni precedenti delle applicazioni esistenti. Ne risulta che Amazon RDS non applica aggiornamenti automatici alla versione principale. È necessario modificare manualmente l'istanza database. Ti raccomandiamo di eseguire un test approfondito di qualsiasi aggiornamento prima di applicarlo alle istanze di produzione.

Amazon RDS supporta i seguenti aggiornamenti in loco per le versioni principali del motore di database MariaDB:

- Qualsiasi versione MariaDB fino a MariaDB 10.11
- Qualsiasi versione MariaDB a MariaDB 10.6
- Da MariaDB 10.4 a MariaDB 10.5
- Da MariaDB 10.3 a MariaDB 10.4

Per eseguire un aggiornamento della versione principale a una versione di MariaDB inferiore alla 10.6, è necessario eseguire l'aggiornamento a ogni versione principale in ordine. Ad esempio, per eseguire l'aggiornamento dalla versione 10.3 alla versione 10.5, esegui l'aggiornamento nel seguente ordine: da 10.3 a 10.4 e quindi da 10.4 a 10.5.

Se usi un gruppo di parametri personalizzato ed esegui l'aggiornamento di una versione principale, devi specificare un gruppo di parametri predefinito per la nuova versione del motore di database oppure creare un gruppo di parametri personalizzato per la nuova versione del motore di database. L'associazione del nuovo gruppo di parametri all'istanza database richiede il riavvio del database avviato dal cliente al termine dell'aggiornamento. Lo stato del gruppo di parametri dell'istanza riporterà `pending-reboot` se è necessario riavviare l'istanza per applicare le modifiche del gruppo di parametri. È possibile visualizzare lo stato del gruppo di parametri dell'istanza nella AWS Management Console oppure utilizzando una chiamata "describe" ("descrivi") come `describe-db-instances`.

Aggiornamento di un'istanza database MariaDB

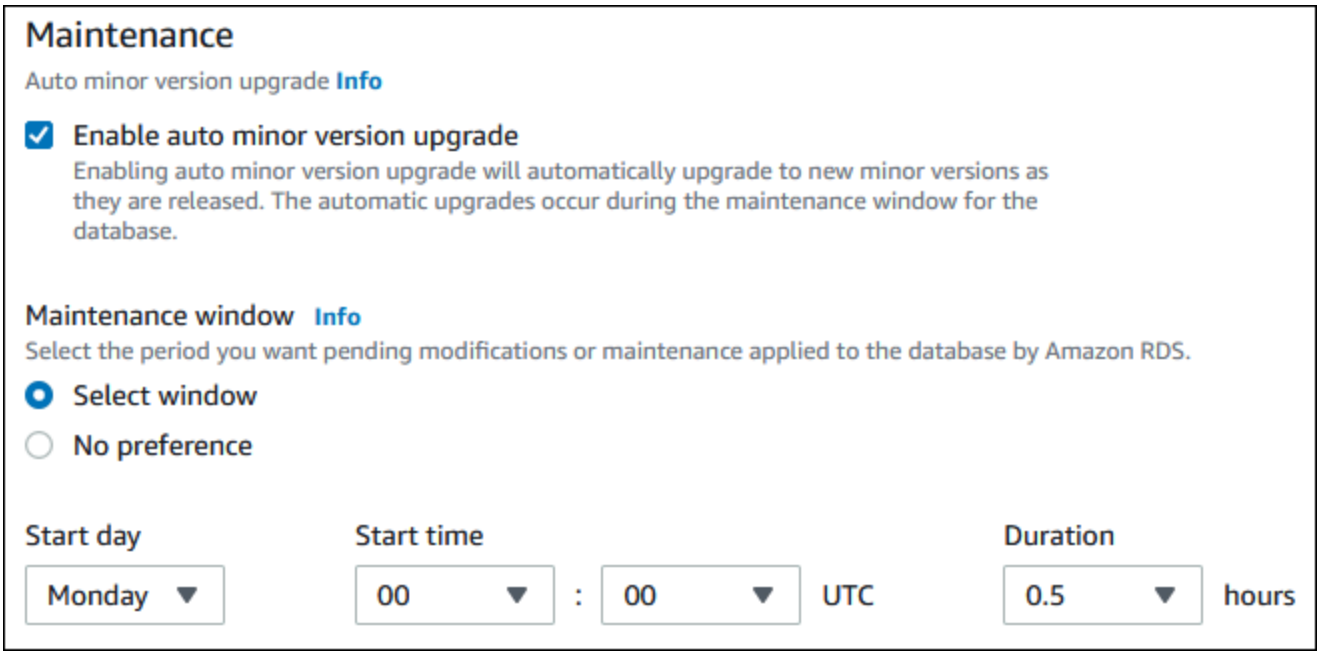
Per informazioni sull'aggiornamento manuale o automatico di un'istanza database MariaDB, consulta Aggiornamento della versione del motore di un'istanza database.

Aggiornamenti a versioni secondarie automatiche per MariaDB

Se specifichi le seguenti impostazioni durante la creazione o la modifica di un'istanza database, puoi decidere aggiornare automaticamente l'istanza database.

- L'impostazione di aggiornamento automatico della versione secondaria deve essere attivata.
- L'impostazione del periodo di conservazione del backup deve essere maggiore di 0.

In, queste impostazioni si trovano in AWS Management Console Configurazione aggiuntiva. L'immagine che segue mostra l'impostazione Auto Minor Version Upgrade (Aggiornamento automatico versione secondaria).



Per ulteriori informazioni su queste impostazioni, consultare Impostazioni per istanze database.

Per alcune versioni principali di RDS per MariaDB, in Regioni AWS alcune, una versione secondaria viene designata da RDS come versione di aggiornamento automatico. Una volta che una versione secondaria è stata testata e approvata da Amazon RDS, l'aggiornamento della versione secondaria avviene automaticamente nel corso della finestra di manutenzione. RDS non imposta mai automaticamente le nuove release secondarie come versione di aggiornamento automatico. Prima che RDS indichi una versione di aggiornamento automatico più recente, vengono considerati diversi livelli di valutazione, quali:

- Problemi di sicurezza noti

- Bug nella versione della community di MariaDB
- Stabilità generale del parco istanze da quando la versione secondaria è stata rilasciata

> **Note**
>
> Il supporto per l'utilizzo delle versioni TLS 1.0 e 1.1 è stato rimosso a partire da specifiche versioni secondarie di MariaDB. Per informazioni sulle versioni secondarie di MariadB supportate, consulta. the section called “Supporto SSL/TLS”

È possibile utilizzare il seguente AWS CLI comando per determinare l'attuale versione di destinazione dell'aggiornamento secondario automatico per una versione secondaria di MariaDB specificata in una specifica. Regione AWS

PerLinux, omacOS: Unix

```
aws rds describe-db-engine-versions \
--engine mariadb \
--engine-version minor-version \
--region region \
--query "DBEngineVersions[*].ValidUpgradeTarget[*].
{AutoUpgrade:AutoUpgrade,EngineVersion:EngineVersion}" \
--output text
```

Per Windows:

```
aws rds describe-db-engine-versions ^
--engine mariadb ^
--engine-version minor-version ^
--region region ^
--query "DBEngineVersions[*].ValidUpgradeTarget[*].
{AutoUpgrade:AutoUpgrade,EngineVersion:EngineVersion}" ^
--output text
```

Ad esempio, il AWS CLI comando seguente determina l'obiettivo di aggiornamento secondario automatico per la versione secondaria di MariaDB 10.5.16 negli Stati Uniti orientali (Ohio) (us-east-2). Regione AWS

Per, oUnix: Linux macOS

```
aws rds describe-db-engine-versions \
--engine mariadb \
--engine-version 10.5.16 \
--region us-east-2 \
--query "DBEngineVersions[*].ValidUpgradeTarget[*].
{AutoUpgrade:AutoUpgrade,EngineVersion:EngineVersion}" \
--output table
```

Per Windows:

```
aws rds describe-db-engine-versions ^
--engine mariadb ^
--engine-version 10.5.16 ^
--region us-east-2 ^
--query "DBEngineVersions[*].ValidUpgradeTarget[*].
{AutoUpgrade:AutoUpgrade,EngineVersion:EngineVersion}" ^
--output table
```

L'output è simile a quello riportato di seguito.

```
---------------------------------
|    DescribeDBEngineVersions    |
+--------------+-----------------+
|  AutoUpgrade |  EngineVersion  |
+--------------+-----------------+
|  True        |  10.5.17        |
|  False       |  10.5.18        |
|  False       |  10.5.19        |
|  False       |  10.6.5         |
|  False       |  10.6.7         |
|  False       |  10.6.8         |
|  False       |  10.6.10        |
|  False       |  10.6.11        |
|  False       |  10.6.12        |
+--------------+-----------------+
```

In questo esempio, il valore `AutoUpgrade` è `True` per MariaDB versione 10.5.17. Quindi, la destinazione dell'aggiornamento secondario automatico è MariaDB versione 10.5.17, che è evidenziata nell'output.

Un'istanza database MariaDB viene aggiornata automaticamente durante la finestra di manutenzione se vengono soddisfatti i seguenti criteri:

- L'impostazione di aggiornamento automatico della versione secondaria deve essere attivata.
- L'impostazione del periodo di conservazione del backup deve essere maggiore di 0.
- L'istanza database esegue una versione motore database minore rispetto a una versione minore automatica dell'aggiornamento corrente.

Per ulteriori informazioni, consulta Aggiornamento automatico della versione secondaria del motore.

Utilizzo di una replica di lettura per ridurre i tempi di inattività durante l'aggiornamento di un database MariaDB

Nella maggior parte dei casi, un'implementazione blu/verde è l'opzione migliore per ridurre i tempi di inattività durante l'aggiornamento di un'istanza database MariaDB. Per ulteriori informazioni, consulta Utilizzo delle implementazioni blu/verde Amazon RDS per gli aggiornamenti del database.

Se non è possibile utilizzare un'implementazione blu/verde e l'istanza database MariaDB è attualmente in uso con un'applicazione di produzione, puoi seguire la seguente procedura per aggiornare la versione del database per l'istanza database. Questa procedura permette di ridurre i tempi di indisponibilità dell'applicazione.

Utilizzando una replica di lettura, è possibile eseguire la maggior parte dei passaggi di manutenzione in anticipo e ridurre al minimo le modifiche necessarie durante l'interruzione effettiva. Con questa tecnica, è possibile testare e preparare la nuova istanza database senza apportare alcuna modifica all'istanza database esistente.

La seguente procedura mostra un esempio di aggiornamento da MariaDB versione 10.5 a MariaDB versione 10.6. Puoi utilizzare la stessa procedura generale per gli aggiornamenti ad altre versioni principali.

Per aggiornare un database MariaDB con un'istanza database in uso

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Crea una replica di lettura dell'istanza database MariaDB 10.5. Questo processo crea una copia aggiornabile del database. Potrebbero esistere già presenti altre repliche di lettura dell'istanza database.
    a. Nella console, seleziona Database e quindi l'istanza database da aggiornare.
    b. Per Actions (Operazioni), scegliere Create read replica (Crea replica di lettura).

c. Fornisci un valore per DB instance identifier (Identificatore di istanza database) per la replica di lettura e assicurati che DB instance class (Classe di istanza database) e altre impostazioni corrispondano all'istanza database MariaDB 10.5.

d. Scegliere Create read replica (Crea replica di lettura).

3. (Facoltativo) Una volta creata la replica di lettura e il campo Stato riporta Disponibile, converti la replica di lettura in una implementazione Multi-AZ e abilita i backup.

   Per impostazione predefinita, una replica di lettura viene creata come implementazione single-AZ con backup disabilitati. Poiché la replica di lettura diventerà in definitiva l'istanza database di produzione, è opportuno configurare un'implementazione multi-AZ e abilitare i backup in questo momento.

   a. Nella console, seleziona Database, quindi seleziona la replica di lettura appena creata.

   b. Scegliere Modify (Modifica).

   c. Per Implementazione Multi-AZ, seleziona Crea istanza di standby.

   d. In Backup Retention Period (Periodo di conservazione dei backup), seleziona un valore positivo diverso da zero, ad esempio 3 giorni, quindi scegli Continue (Continua).

   e. In Scheduling of Modifications (Pianificazione delle modifiche), scegliere Apply immediately (Applica immediatamente).

   f. Scegliere Modify DB Instance (Modifica istanza database).

4. Quando il campo Status (Stato) della replica di lettura riporta Available (Disponibile), aggiorna la replica di lettura a MariaDB 10.6.

   a. Nella console, seleziona Database, quindi seleziona la replica di lettura appena creata.

   b. Scegliere Modify (Modifica).

   c. In DB engine version (Versione motore database) scegli la versione MariaDB 10.6 da aggiornare e quindi seleziona Continue (Continua).

   d. In Scheduling of Modifications (Pianificazione delle modifiche), scegliere Apply immediately (Applica immediatamente).

   e. Scegliere Modify DB Instance (Modifica istanza database) per avviare l'aggiornamento.

5. Quando l'aggiornamento è completo e lo stato mostra Available, verifica che la replica di lettura aggiornata sia up-to-date con l'istanza database di MariaDB 10.5 di origine. Per verificare, connettiti alla replica di lettura ed esegui il comando `SHOW REPLICA STATUS`. Se il `Seconds_Behind_Master` campo è, allora la replica è`0`. up-to-date

> **Note**
>
> Le versioni precedenti di MariaDB utilizzavano `SHOW SLAVE STATUS` anziché `SHOW REPLICA STATUS`. Se si utilizza una versione di MariaDB precedente alla 10.6, usa `SHOW SLAVE STATUS`.

6. (Facoltativo) Crea una replica di lettura della replica di lettura.

   Se desideri che l'istanza database disponga di una replica di lettura dopo che è stata promossa a un'istanza database autonoma, puoi crearla in questo momento.

   a. Nella console, seleziona Database, quindi scegli la replica di lettura appena aggiornata.
   b. Per Actions (Operazioni), scegliere Create read replica (Crea replica di lettura).
   c. Fornisci un valore per DB instance identifier (Identificatore di istanza database) per la replica di lettura e assicurati che DB instance class (Classe di istanza database) e altre impostazioni corrispondano all'istanza database MariaDB 10.5.
   d. Scegliere Create read replica (Crea replica di lettura).

7. (Facoltativo) Configura un gruppo di parametri database personalizzato per la replica di lettura.

   Se desideri che l'istanza database utilizzi un gruppo di parametri personalizzato dopo che è stato promossa a un'istanza database autonoma, puoi creare il gruppo e associarlo alla replica di lettura.

   a. Crea un gruppo di parametri database personalizzato per MariaDB 10.6. Per istruzioni, consulta Creazione di un gruppo di parametri del database.
   b. Modifica i parametri che desideri modificare nel gruppo di parametri database appena creato. Per istruzioni, consulta Modifica di parametri in un gruppo di parametri del database.
   c. Nella console seleziona Database, quindi scegli la replica di lettura.
   d. Scegliere Modify (Modifica).
   e. Per DB parameter group (Gruppo di parametri database), seleziona il gruppo di parametri database MariaDB 10.6 appena creato, quindi scegli Continue (Continua).
   f. In Scheduling of Modifications (Pianificazione delle modifiche), scegliere Apply immediately (Applica immediatamente).
   g. Scegliere Modify DB Instance (Modifica istanza database) per avviare l'aggiornamento.

8. Promuovi la replica di lettura MariaDB 10.6 a un'istanza database autonoma.

> **Important**
>
> Quando promuovi la replica di lettura MariaDB 10.6 a un'istanza database autonoma, la replica non sarà più dell'istanza database MariaDB 10.5. Ti consigliamo di promuovere la replica di lettura MariaDB 10.6 durante una finestra di manutenzione quando l'istanza database MariaDB 10.5 di origine è in modalità di sola lettura e tutte le operazioni di scrittura sono sospese. Al termine dell'operazione, è possibile indirizzare le operazioni di scrittura all'istanza database MariaDB 10.6 aggiornata per evitare la perdita di qualsiasi informazione.
>
> Inoltre, prima di promuovere la replica di lettura MariaDB 10.6, ti consigliamo di eseguire tutte le operazioni DDL (Data Definition Language) necessarie sulla replica di lettura MariaDB 10.6. Un esempio di tale operazione è la creazione degli indici. Questo approccio consente di evitare qualsiasi effetto negativo sulle prestazioni della replica di lettura MariaDB 10.6 dopo la promozione. Per promuovere una replica di lettura, utilizzare la procedura seguente.

a. Nella console, seleziona Database, quindi scegli la replica di lettura appena aggiornata.
b. In Actions (Operazioni), selezionare Promote (Promuovi).
c. Scegliere Yes (Sì) per abilitare backup automatizzati per l'istanza della replica di lettura. Per ulteriori informazioni, consulta Introduzione ai backup.
d. Scegli Continue (Continua).
e. Selezionare Promote read replica (Promuovi replica di lettura).

9. Ora si dispone di una versione aggiornata del database MariaDB. A questo punto, puoi indirizzare le applicazioni alla nuova istanza database MariaDB 10.6.

Importazione di dati in un'istanza database MariaDB

Puoi utilizzare diverse tecniche per importare i dati in un'istanza database RDS for MariaDB. L'approccio migliore dipende dall'origine e dalla quantità dei dati e dal fatto che l'importazione venga eseguita in modo occasionale o continuo. Se stai migrando un'applicazione insieme a tutti i suoi dati, dovrai valutare per quanto tempo il sistema può rimanere inattivo.

La tabella di seguito riporta le varie tecniche per importare i dati in un'istanza database RDS for MariaDB.

Origine	Quantità di dati	Una volta o continua	Tempo di inattività delle applicazioni	Tecnica	Ulteriori informazioni
Istanza database MariaDB esistente	Qualsiasi	Una volta o continua	Minima	Creare una replica di lettura per la replica continua. Promuovere la replica di lettura per la creazione una tantum di una nuova istanza database.	Uso delle repliche di lettura dell'istanza database
Database MariaDB o MySQL esistente	Small	Una volta	Medio	Copiare i dati direttamente nell'istanza database MySQL utilizzando un'utilità a riga di comando.	Importazione dei dati da un database MariaDB o MySQL a un'istanza

Origine	Quantità di dati	Una volta o continua	Tempo di inattività delle applicazioni	Tecnica	Ulteriori informazioni
					database MariaDB o MySQL
Dati non salvati in un database esistente	Medium	Una volta	Medio	Crea file flat e importali utilizzando istruzioni `LOAD DATA LOCAL INFILE` MySQL.	Importazione dei dati da qualsiasi origine a un'istanza database MariaDB o MySQL

Origine	Quantità di dati	Una volta o continua	Tempo di inattività delle applicazioni	Tecnica	Ulteriori informazioni
Database MariaDB o MySQL esistente in locale o su Amazon EC2	Qualsiasi	Continua	Minima	Configurare la replica utilizzando un database MariaDB o MySQL esistente come origine della replica. Puoi configurare la replica in un'istanza database MariaDB tramite identificatori globali di transazione (GTID) di MariaDB se l'istanza esterna è MariaDB versione 10.0.24 o successiva o tramite le coordinate del log binario per le istanze MySQL o MariaDB nelle versioni precedenti alla 10.0.24. I GTID in MariaDB vengono implementati in modo diverso da quelli in MySQL, che non sono supportati da Amazon RDS.	Configurazione della replica della posizione del file di log binario con un'istanza di origine esterna. Importazione dei dati in un'istanza database MariaDB o MySQL di Amazon RDS, riducendo i tempi

Origine	Quantità di dati	Una volta o continua	Tempo di inattività delle applicazioni	Tecnica	Ulteriori informazioni
					di inattività
Qualsiasi database esistente	Qualsiasi	Una volta o continua	Minima	AWS Database Migration Service Utilizzatelo per migrare il database con tempi di inattività minimi e, per molti motori di database di database, continuare la replica continua.	Cos'è AWS Database Migration Service e Utilizzo di un database compatibile con MySQL come destinazione per AWS DMS nella Guida per l'utente di AWS Database Migration Service

> **Note**
>
> Il database di sistema mysql contiene le informazioni di autenticazione e autorizzazione necessarie per accedere all'istanza database e ai dati. L'eliminazione, la modifica, la ridenominazione o il troncamento di tabelle, dati o altro contenuto del database mysql nell'istanza database può causare errori e rendere inaccessibili dati e istanza database. In tal caso, l'istanza DB può essere ripristinata da un'istantanea utilizzando i comandi AWS CLI `restore-db-instance-from-db-snapshot`o ripristinata utilizzando i comandi. `restore-db-instance-to-point-in-time`

Importazione dei dati da un database MariaDB o MySQL a un'istanza database MariaDB o MySQL

In alternativa, puoi importare i dati da un database MariaDB o MySQL esistente a un'istanza database MySQL o MariaDB. A questo scopo, copia il database con mysqldump e reindirizzalo direttamente nell'istanza database MariaDB o MySQL. L'utility a riga di comando `mysqldump` viene spesso usata per creare backup e trasferire dati da un server MariaDB o MySQL a un altro. ed è inclusa nel software del client MySQL e MariaDB.

> **Note**
>
> Se stai importando o esportando grandi quantità di dati con un'istanza DB MySQL, è più affidabile e veloce spostare i dati da e verso Amazon RDS utilizzando file di backup e Amazon S3. `xtrabackup` Per ulteriori informazioni, consulta Ripristino di un backup in un'istanza database MySQL.

Un tipico comando `mysqldump` per spostare dati da un database esterno a un'istanza database Amazon RDS è simile al seguente.

```
mysqldump -u local_user \
    --databases database_name \
    --single-transaction \
    --compress \
    --order-by-primary  \
    -plocal_password | mysql -u RDS_user \
        --port=port_number \
```

```
        --host=host_name \
        -pRDS_password
```

> **Important**
>
> Assicurati di non lasciare spazi tra l'opzione `-p` e la password immessa.
> Specifica credenziali diverse dai prompt mostrati qui come best practice per la sicurezza.

Assicurati di essere a conoscenza dei seguenti suggerimenti e considerazioni:

- Escludi gli schemi seguenti dal file dump: `sys`, `performance_schema` e `information_schema`. Per impostazione predefinita, l'utility `mysqldump` esclude questi schemi.
- Se devi migrare utenti e privilegi, prendi in considerazione l'utilizzo di uno strumento che genera il linguaggio di controllo dei dati (DCL) per ricrearli, come l'utilità. pt-show-grants
- L'utente che esegue l'importazione deve avere accesso all'istanza database. Per ulteriori informazioni, consulta Controllo dell'accesso con i gruppi di sicurezza.

I parametri utilizzati sono i seguenti:

- `-u` *`local_user`* – Specifica un nome utente. La prima volta che usi questo parametro, devi specificare il nome di un account utente nel database MariaDB o MySQL locale, identificato dal parmetro `--databases`.
- `--databases` *`database_name`* – Specifica il nome del database nell'istanza database MariaDB o MySQL locale che vuoi importare in Amazon RDS.
- `--single-transaction` – Verifica che tutti i dati caricati dal database locale siano coerenti a un singolo punto temporale. Nel caso in cui vi siano altri processi che modificano i dati mentre `mysqldump` li legge, l'uso di questo parametro aiuta a preservare l'integrità dei dati.
- `--compress` – Riduce il consumo della larghezza di banda di rete comprimendo i dati dal database locale prima di inviarli ad Amazon RDS.
- `--order-by-primary` – Riduce il tempo di caricamento ordinando i dati di ogni tabella in base alla chiave primaria.
- `-p`*`local_password`* – Specifica una password. La prima volta che usi questo parametro, devi specificare la password per l'account utente identificato dal primo parametro `-u`.

- `-u` *`RDS_user`* – Specifica un nome utente. La seconda volta che usi questo parametro, devi specificare il nome di un account utente nel database predefinito per l'istanza database MariaDB o MySQL identificata dal parametro `--host`.
- `--port` *`port_number`* – Specifica la porta per l'istanza database MariaDB o MySQL. Il valore predefinito è 3306, ma può essere modificato al momento della creazione dell'istanza.
- `--host` *`host_name`* – Specifica il nome del sistema dei nomi di dominio (DNS) dall'endpoint dell'istanza database Amazon RDS, ad esempio `myinstance.123456789012.us-east-1.rds.amazonaws.com`. Puoi trovare il valore dell'endpoint è disponibile nei dettagli dell'istanza, nella console di gestione Amazon RDS.
- `-p`*`RDS_password`* – Specifica una password. La seconda volta che usi questo parametro, devi specificare la password per l'account utente identificato dal secondo parametro `-u`.

Eventuali procedure, trigger, funzioni o eventi devono essere creati manualmente nel database Amazon RDS. Se il database da copiare dovesse contenere questi tipi di oggetti, dovrai escluderli al momento di eseguire `mysqldump`. Per farlo, includi i seguenti parametri obbligatori con il tuo comando `mysqldump`: `--routines=0 --triggers=0 --events=0`.

Nell'esempio seguente viene copiato il database di esempio `world` sull'host locale in un'istanza database MySQL.

PerLinux, o: macOS Unix

```
sudo mysqldump -u localuser \
    --databases world \
    --single-transaction \
    --compress \
    --order-by-primary  \
    --routines=0 \
    --triggers=0 \
    --events=0 \
    -plocalpassword | mysql -u rdsuser \
        --port=3306 \
        --host=myinstance.123456789012.us-east-1.rds.amazonaws.com \
        -prdspassword
```

Per Windows, esegui il comando seguente in un prompt dei comandi che viene aperto facendo clic con il pulsante destro del mouse su Command Prompt (Prompt dei comandi) del menu dei programmi di Windows e selezionando Run as administrator (Esegui come amministratore):

```
mysqldump -u localuser ^
    --databases world ^
    --single-transaction ^
    --compress ^
    --order-by-primary  ^
    --routines=0 ^
    --triggers=0 ^
    --events=0 ^
    -plocalpassword | mysql -u rdsuser ^
        --port=3306 ^
        --host=myinstance.123456789012.us-east-1.rds.amazonaws.com ^
        -prdspassword
```

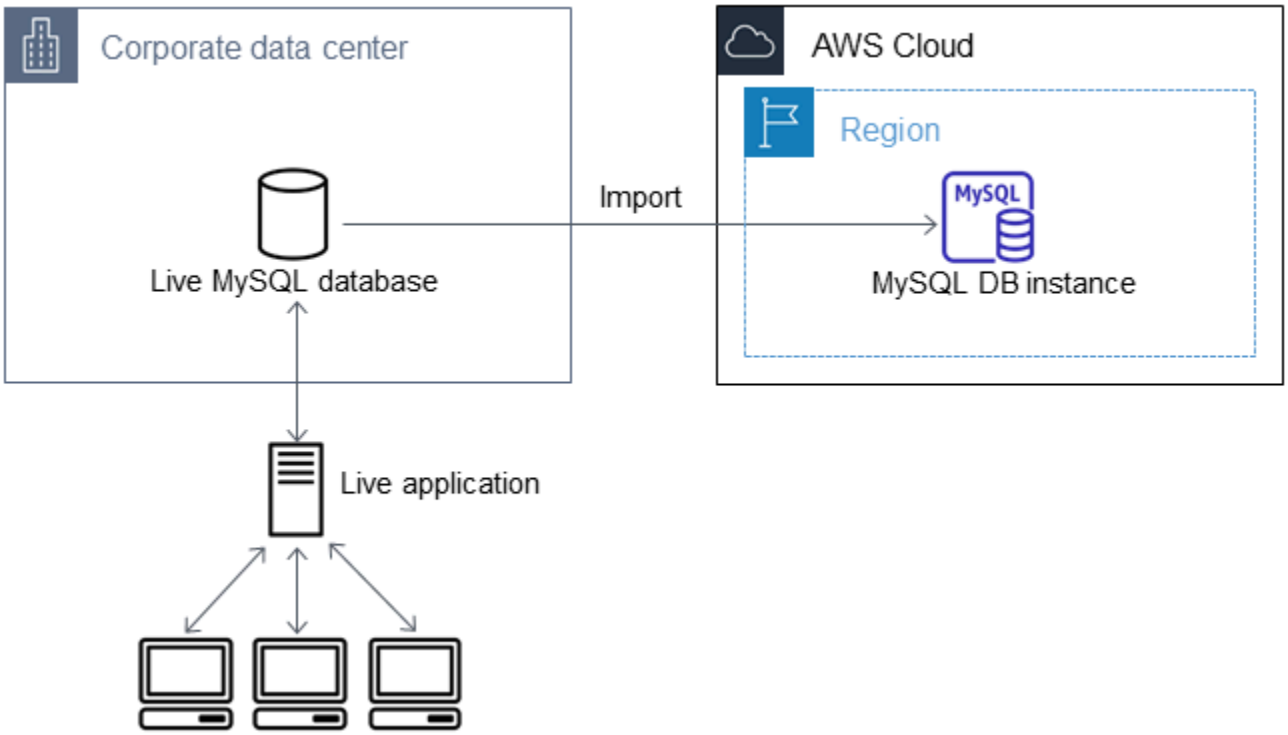
> **Note**
>
> Specifica credenziali diverse dai prompt mostrati qui come best practice per la sicurezza.

Importazione dei dati in un'istanza database MariaDB o MySQL di Amazon RDS, riducendo i tempi di inattività

In alcuni casi, potrebbe essere necessaria l'importazione dei dati da un database MariaDB o MySQL esterno che supporti un'applicazione live per un'istanza database MariaDB o MySQL oppure per un cluster database multi-AZ MySQL. Usa la seguente procedura per ridurre l'impatto sulla disponibilità delle applicazioni. Questa procedura può risultare utile anche quando utilizzi un database di dimensioni particolarmente elevate. Utilizzando questa procedura, è possibile ridurre il costo dell'importazione riducendo la quantità di dati trasmessi attraverso la rete AWS.

Con questa procedura trasferisci una copia dei dati del database in un'istanza Amazon EC2 e li importi in un nuovo database Amazon RDS. Utilizza quindi la replica per portare il database Amazon RDS up-to-date con la tua istanza esterna attiva, prima di reindirizzare l'applicazione al database Amazon RDS. La replica MariaDB viene configurata in base agli identificatori globali di transazione (GTID) se l'istanza esterna è MariaDB 10.0.24 o versioni successive e l'istanza di destinazione è RDS per MariaDB. In alternativa, è possibile configurare la replica in base alle coordinate del log binario. Si consiglia la replica basata su GTID se supportata dal database esterno perché è un metodo più affidabile. Per ulteriori informazioni, consulta ID globali di transazione nella documentazione di MariaDB.

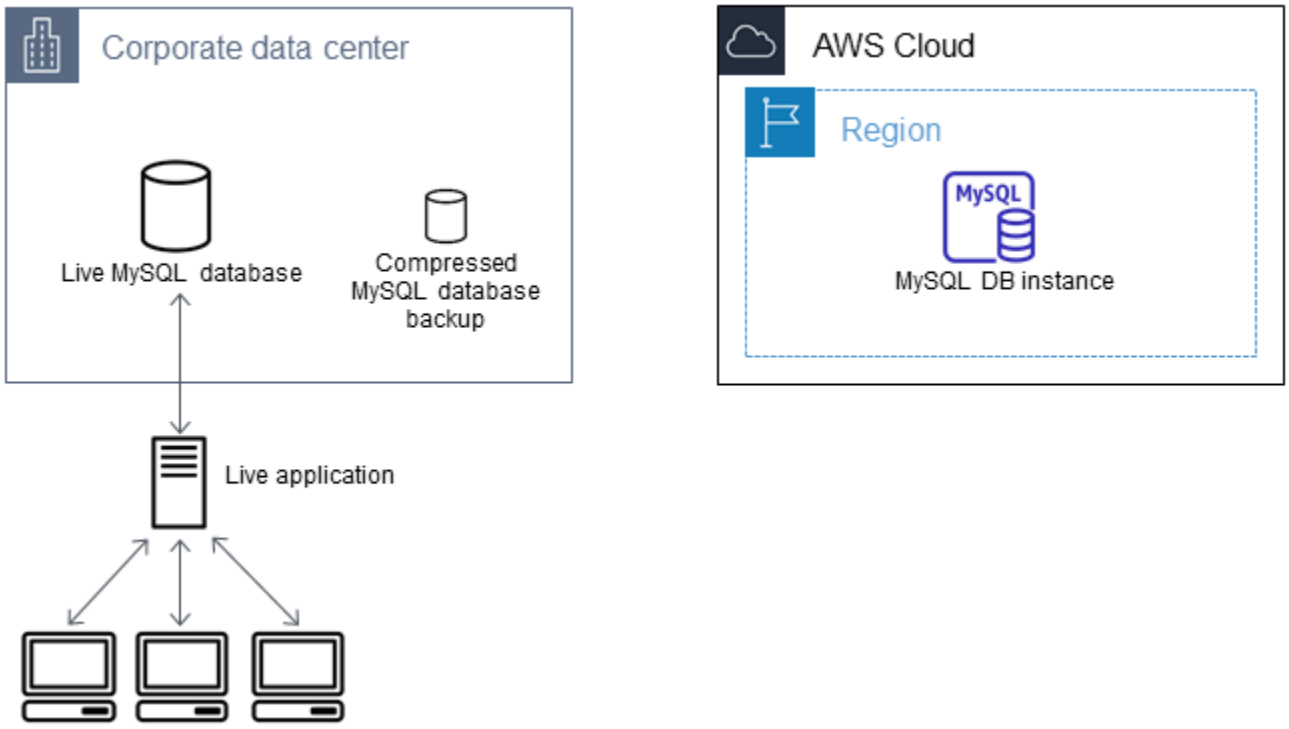
> **Note**
>
> Per importare i dati in un'istanza database MySQL e lo scenario supporta questo approccio, si consiglia di spostare dati da e verso Amazon RDS usando i file di backup e Amazon S3. Per ulteriori informazioni, consulta Ripristino di un backup in un'istanza database MySQL.



> **Note**
>
> Non è consigliabile utilizzare questa procedura per i database MySQL di origine con versioni di MySQL precedenti alla 5.5 a causa dei potenziali problemi di replica. Per ulteriori informazioni, consulta Compatibilità delle repliche fra le versioni di MySQL nella documentazione di MySQL.

Creazione di una copia del database esistente

La prima fase per eseguire la migrazione di grandi quantità di dati in un database RDS per MariaDB o RDS per MySQL riducendo al minimo i tempi di inattività consiste nella creazione di una copia dei dati di origine.



Puoi utilizzare l'utility mysqldump per creare un backup del database in formato SQL o come testo delimitato. Consigliamo di eseguire un test con ciascun formato, fuori dall'ambiente di produzione, per capire quale metodo consente di ridurre maggiormente il tempo di esecuzione di mysqldump.

Consigliamo anche di soppesare le prestazioni di mysqldump e i vantaggi offerti dal caricamento con il formato a testo delimitato. Un backup eseguito con testo delimitato crea un file di testo separato da tabulazioni per ciascuna tabella eliminata. Per ridurre il tempo di importazione del database, puoi caricare questi file in parallelo con il comando `LOAD DATA LOCAL INFILE`. Per ulteriori informazioni sul formato di mysqldump più adatto per il caricamento dei dati, consulta Utilizzo di mysqldump per i backup nella documentazione di MySQL.

Prima di iniziare l'operazione di backup, devi impostare le opzioni di replica nel database MariaDB o MySQL da copiare in Amazon RDS. Le opzioni di replic includono l'attivazione del log binario e l'impostazione di un ID server univoco. L'impostazione di tali opzioni porta il server ad avviare la registrazione delle transazioni del database e lo prepara per diventare l'istanza di replica di origine in una fase successiva del processo.

> **Note**
>
> Utilizzare l'opzione `--single-transaction` con mysqldump perché esegue il dump di uno stato coerente del database. Per garantire un file di dump valido, non eseguire istruzioni DDL

(Data Definition Language) durante l'esecuzione di mysqldump. È possibile pianificare una finestra di manutenzione per queste operazioni.
Escludi gli schemi seguenti dal file dump: `sys`, `performance_schema` e `information_schema`. Per impostazione predefinita, l'utility mysqldump esclude questi schemi.
Per migrare utenti e privilegi, prendi in considerazione l'utilizzo di uno strumento che genera il linguaggio di controllo dei dati (DCL) per ricrearli, come l'utilità. pt-show-grants

Per impostare le opzioni di autenticazione

1. Modificare il file `my.cnf` (posto in genere sotto `/etc`).

   ```
   sudo vi /etc/my.cnf
   ```

   Aggiungere le opzioni `log_bin` e `server_id` alla sezione `[mysqld]`. L'opzione `log_bin` fornisce un identificatore di nome file per i file di log binari. L'opzione `server_id` fornisce un identificatore univoco per il server in relazioni master-replica.

   L'esempio seguente mostra la sezione `[mysqld]` aggiornata di un file `my.cnf`.

   ```
   [mysqld]
   log-bin=mysql-bin
   server-id=1
   ```

   Per ulteriori informazioni, consulta la documentazione di MySQL.
2. Per la replica con un cluster database multi-AZ, imposta `ENFORCE_GTID_CONSISTENCY` e il parametro `GTID_MODE` su `ON`.

   ```
   mysql> SET @@GLOBAL.ENFORCE_GTID_CONSISTENCY = ON;
   ```

   ```
   mysql> SET @@GLOBAL.GTID_MODE = ON;
   ```

   Queste impostazioni non sono necessarie per la replica con un'istanza database.
3. Riavvia il servizio `mysql`.

   ```
   sudo service mysqld restart
   ```

Per creare una copia di backup del database esistente

1. Creare un backup dei dati con l'utility mysqldump, specificando SQL o testo delimitato.

   Specificare `--master-data=2` per creare un file di backup che possa essere utilizzato per avviare la replica fra i server. Per ulteriori informazioni, consultare la documentazione di mysqldump.

   Per migliorare le prestazioni e garantire l'integrità dei dati, utilizzare le opzioni `--order-by-primary` e `--single-transaction` di mysqldump.

   Per non includere il database del sistema MySQL nel backup, non utilizzare l'opzione `--all-databases` con mysqldump. Per ulteriori informazioni, consultare Creating a Data Snapshot Using mysqldump nella documentazione di MySQL.

   Se necessario, utilizzare `chmod` per avere la certezza che la directory in cui viene creato il file di backup sia scrivibile.

   > **Important**
   >
   > In Windows, eseguire la finestra di comando come amministratore.

   - Per produrre un output SQL, utilizzare il comando seguente.

     PerLinux, o: macOS Unix

     ```
     sudo mysqldump \
         --databases database_name \
         --master-data=2  \
         --single-transaction \
         --order-by-primary \
         -r backup.sql \
         -u local_user \
         -p password
     ```

     > **Note**
     >
     > Specifica credenziali diverse dai prompt mostrati qui come best practice per la sicurezza.

Per Windows:

```
mysqldump ^
    --databases database_name ^
    --master-data=2  ^
    --single-transaction ^
    --order-by-primary ^
    -r backup.sql ^
    -u local_user ^
    -p password
```

> **Note**
>
> Specifica credenziali diverse dai prompt mostrati qui come best practice per la sicurezza.

- Per produrre un output in testo delimitato, utilizzare il comando seguente.

  Per LinuxmacOS, oUnix:

```
sudo mysqldump \
    --tab=target_directory \
    --fields-terminated-by ',' \
    --fields-enclosed-by '"' \
    --lines-terminated-by 0x0d0a \
    database_name \
    --master-data=2 \
    --single-transaction \
    --order-by-primary \
    -p password
```

  Per Windows:

```
mysqldump ^
    --tab=target_directory ^
    --fields-terminated-by "," ^
    --fields-enclosed-by """ ^
    --lines-terminated-by 0x0d0a ^
    database_name ^
    --master-data=2 ^
    --single-transaction ^
```

```
    --order-by-primary ^
    -p password
```

> **Note**
>
> Specifica credenziali diverse dai prompt mostrati qui come best practice per la sicurezza. Eventuali procedure, trigger, funzioni o eventi devono essere creati manualmente nel database Amazon RDS. Se il database da copiare dovesse contenere questi tipi di oggetti, dovrai escluderli al momento di eseguire mysqldump. A tale scopo, includi i seguenti argomenti con il comando mysqldump: `--routines=0 --triggers=0 --events=0`.

Quando si utilizza il formato con testo delimitato, il commento `CHANGE MASTER TO` viene restituito all'esecuzione di mysqldump. Tale commento contiene il nome e la posizione del file log principale. Se l'istanza esterna è diversa da MariaDB versione 10.0.24 o successiva, annotare i valori per `MASTER_LOG_FILE` e `MASTER_LOG_POS`. Questi valori sono necessari durante l'impostazione della replica.

```
-- Position to start replication or point-in-time recovery from
--
-- CHANGE MASTER TO MASTER_LOG_FILE='mysql-bin-changelog.000031',
 MASTER_LOG_POS=107;
```

Se si utilizza il formato SQL, è possibile ottenere il nome e la posizione del file log principale nel commento `CHANGE MASTER TO` nel file di backup. Se l'istanza esterna è MariaDB, versione 10.0.24 o successiva, si può ottenere il GTID nella fase successiva.

2. Se l'istanza esterna è MariaDB, versione 10.0.24 o successiva, si utilizza la replica basata su GTID. Eseguire `SHOW MASTER STATUS` nell'istanza MariaDB esterna per ottenere il nome e la posizione del file di log binario e convertirlo in un GTID utilizzando `BINLOG_GTID_POS` nell'istanza MariaDB esterna.

   ```
   SELECT BINLOG_GTID_POS('binary log file name', binary log file position);
   ```

   Annotare il GTID restituito, perché sarà necessario per configurare la replica.

3. Comprimi i dati copiati per ridurre la quantità di risorse di rete necessarie per copiare i dati nell'istanza database Amazon RDS. Annota la dimensione del file di backup. Questa informazione

è necessaria per determinare le dimensioni dell'istanza Amazon EC2 da creare. Al termine, comprimere il file di backup con GZIP o un'altra utility simile.

- Per comprimere l'output SQL, utilizzare il comando seguente.

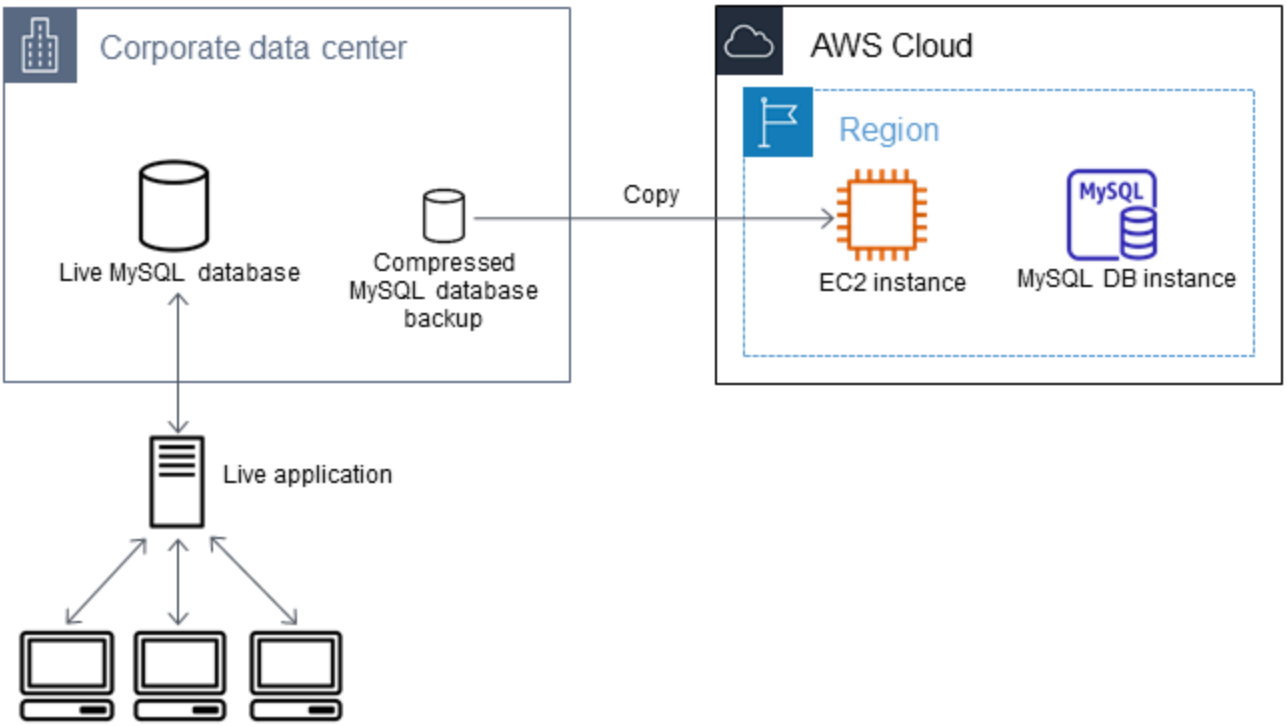
```
gzip backup.sql
```

- Per comprimere un output in testo delimitato, utilizzare il comando seguente.

```
tar -zcvf backup.tar.gz target_directory
```

Creazione di un'istanza Amazon EC2 e copia del database compresso

La copia del file di backup del database compresso in un'istanza Amazon EC2 richiede una quantità di risorse di rete inferiore rispetto alla copia diretta di dati non compressi da un'istanza database a un'altra. Una volta che i dati sono presenti in Amazon EC2, puoi copiarli direttamente nell'istanza database MariaDB o MySQL. Per risparmiare sul costo delle risorse di rete, l'istanza Amazon EC2 deve trovarsi nella stessa AWS regione dell'istanza Amazon RDS DB. La presenza dell'istanza Amazon EC2 nella stessa AWS regione del database Amazon RDS riduce anche la latenza di rete durante l'importazione.



Per creare un'istanza Amazon EC2 e copiare i dati

1. Nel luogo in Regione AWS cui prevedi di creare il database RDS, crea un cloud privato virtuale (VPC), un gruppo di sicurezza VPC e una sottorete VPC. Verificare che le regole in entrata del gruppo di sicurezza VPC consentano agli indirizzi IP necessari per l'applicazione di connettersi ad AWS. Puoi specificare un intervallo di indirizzi IP (ad esempio `203.0.113.0/24`) oppure un altro gruppo di sicurezza VPC. È possibile utilizzare la Console di gestione Amazon VPC per creare e gestire VPC, sottoreti e gruppi di sicurezza. Per ulteriori informazioni, consultare le nozioni di base su Amazon VPC nella Guida alle operazioni di base di Amazon Virtual Private Cloud.
2. Apri la console di gestione Amazon EC2 e scegli la AWS regione in cui contenere sia l'istanza Amazon EC2 che il database Amazon RDS. Avviare un'istanza di Amazon EC2 utilizzando il VPC, la sottorete e il gruppo di sicurezza creati nella fase 1. Assicurarsi di selezionare un tipo di istanza con spazio di storage sufficiente per il file di backup del database decompresso. Per informazioni sulle istanze Amazon EC2, consulta l'argomento Nozioni di base sulle istanze Amazon EC2 Linux nella Guida per l'utente di Amazon Elastic Compute Cloud per Linux.
3. Per connetterti al database Amazon RDS dall'istanza Amazon EC2, modifica il gruppo di sicurezza VPC. Aggiungere una regola in entrata specificando l'indirizzo IP privato e l'istanza EC2. L'indirizzo IP privato è indicato nella scheda Details (Dettagli) del riquadro Instance (Istanza) della finestra della console EC2. Per modificare il gruppo di sicurezza VPC e aggiungere una regola in entrata, selezionare Security Groups (Gruppi di sicurezza) nel riquadro di navigazione della console di EC2, selezionare il gruppo di sicurezza e aggiungere una regola in entrata per MySQL/Aurora specificando l'indirizzo IP privato dell'istanza EC2. Per ulteriori informazioni sull'aggiunta di una regola in entrata a un gruppo di sicurezza VPC, consulta Aggiunta ed eliminazione delle regole nella Guida per l'utente di Amazon VPC.
4. Copiare il file compresso con il backup del database dal sistema locale all'istanza Amazon EC2. Se necessario, utilizzare `chmod` per ottenere l'autorizzazione di scrittura per la directory di destinazione dell'istanza Amazon EC2. Il file può essere copiato con `scp` oppure con un client Secure Shell (SSH). Di seguito è riportato un esempio.

```
scp -r -i key pair.pem backup.sql.gz ec2-user@EC2 DNS:/target_directory/backup.sql.gz
```

> **Important**
>
> Assicurarsi di copiare i dati sensibili utilizzando un protocollo di trasferimento di rete sicuro.

5. Eseguire la connessione all'istanza Amazon EC2 e installare gli ultimi aggiornamenti e gli strumenti del clien MySQL mediante i seguenti comandi.

```
sudo yum update -y
sudo yum install mysql -y
```

Per ulteriori informazioni, consultare la pagina relativa alla connessione all'istanza nella Guida per l'utente di Amazon Elastic Compute Cloud per Linux.

> **Important**
>
> Questo esempio installa il client MySQL su una Amazon Machine Image (AMI) per una distribuzione Amazon Linux. Non si applica all'installazione del client MySQL su una distribuzione diversa, come Ubuntu o Red Hat Enterprise Linux. Per informazioni sull'installazione di MySQL, visita la pagina Installazione e aggiornamento di MySQL nella documentazione MySQL.

6. Durante la connessione all'istanza Amazon EC2 decomprimere il file di backup del database. Di seguito vengono mostrati gli esempi.
   - Per decomprimere l'output SQL, utilizzare il comando seguente.

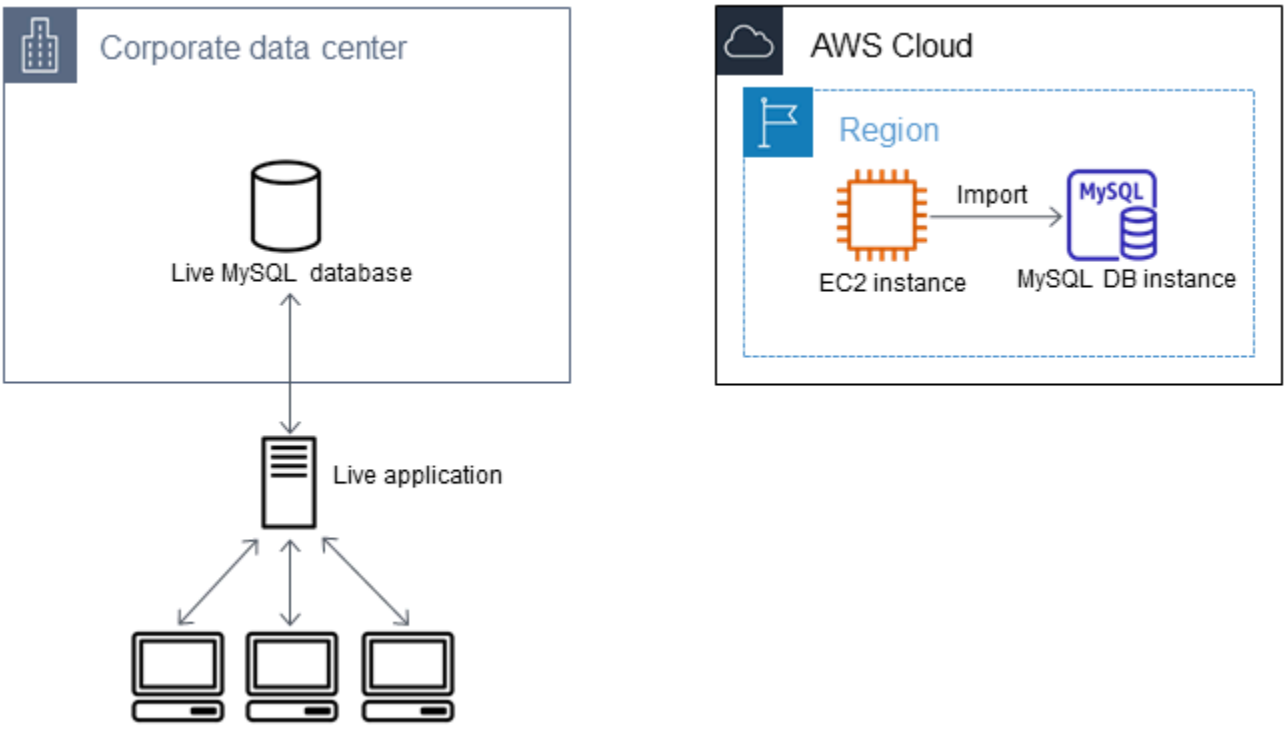
```
gzip backup.sql.gz -d
```

   - Per decomprimere un output in testo delimitato, utilizzare il comando seguente.

```
tar xzvf backup.tar.gz
```

Creazione di un database MySQL o MariaDB e importazione dei dati dall'istanza Amazon EC2

Creando un'istanza DB MariaDB, un'istanza DB MySQL o un cluster DB MySQL Multi-AZ nella AWS stessa regione dell'istanza Amazon EC2, puoi importare il file di backup del database da EC2 più velocemente che su Internet.



Per creare un database MariaDB o MySQL e importare i dati

1. Determina la classe di istanza database e la quantità di spazio di archiviazione necessaria per supportare il carico di lavoro previsto per il database Amazon RDS. Come parte di questo processo, è necessario valutare la quantità di spazio richiesta e la capacità di elaborazione per le procedure di caricamento dati. Valuta anche l'occorrente per gestire il carico di lavoro della produzione. È possibile produrre una stima sulla base delle dimensioni e delle risorse del database MariaDB o MySQL di origine. Per ulteriori informazioni, consulta Classi di istanze database.
2. Crea un'istanza DB o un cluster DB Multi-AZ nella AWS regione che contiene la tua istanza Amazon EC2.

   Per creare un cluster database multi-AZ MySQL, segui le istruzioni riportate in Creazione di un cluster di database Multi-AZ.

   Per creare un'istanza database MariaDB o MySQL, segui le istruzioni riportate in Creazione di un'istanza database Amazon RDS e attieniti alle seguenti linee guida:

   - Specifica una versione del motore di database compatibile con l'istanza database di origine, come indicato di seguito:
     - Se l'istanza di origine è MySQL 5.5.x, l'istanza database Amazon RDS deve essere MySQL.

- Se l'istanza di origine è MySQL 5.6.x o 5.7.x, l'istanza database Amazon RDS deve essere MySQL o MariaDB.
- Se l'istanza di origine è MySQL 8.0.x, l'istanza database di Amazon RDS deve essere MySQL 8.0.x.
- Se l'istanza di origine è MariaDB 5.5 o versione successiva, l'istanza database Amazon RDS deve essere MariaDB.

- Specifica lo stesso cloud privato virtuale (VPC) e lo stesso gruppo di sicurezza VPC dell'istanza Amazon EC2. Questo approccio garantisce che l'istanza Amazon EC2 e l'istanza Amazon RDS siano visibili una all'altra in rete. Assicurati che l'istanza database sia accessibile pubblicamente. L'istanza database deve essere pubblicamente accessibile per impostare la replica con il database di origine descritto successivamente in questo argomento.
- Non configurare più zone di disponibilità, retention dei backup o repliche di lettura fino a quando non è stato importato il backup del database. Al termine dell'importazione, puoi configurare le varie zone di disponibilità e la conservazione dei backup per l'istanza di produzione.

3. Esamina le opzioni di configurazione predefinite per il database Amazon RDS. Se il gruppo di parametri predefinito per il database non include le opzioni di configurazione desiderate, cercane uno che le contenga oppure crea un gruppo di parametri nuovo. Per ulteriori informazioni sulla creazione di gruppi di parametri, consulta Utilizzo di gruppi di parametri.
4. Connettiti al nuovo database Amazon RDS come utente master. Creare gli utenti necessari per supportare gli amministratori, le applicazioni e i servizi che devono accedere all'istanza. Il nome host per il database Amazon RDS corrisponde al valore dell'endpoint per l'istanza, senza includere il numero di porta. Un esempio è `mysampledb.123456789012.us-west-2.rds.amazonaws.com`. Il valore dell'endpoint è disponibile nei dettagli del database nella Console di gestione Amazon RDS.
5. Eseguire la connessione all'istanza di Amazon EC2. Per ulteriori informazioni, consultare la pagina relativa alla connessione all'istanza nella Guida per l'utente di Amazon Elastic Compute Cloud per Linux.
6. Connettiti al database Amazon RDS come host remoto dall'istanza Amazon EC2 usando il comando `mysql`. Di seguito è riportato un esempio.

```
mysql -h host_name -P 3306 -u db_master_user -p
```

Il nome host corrisponde all'endpoint del database Amazon RDS.

7. Al prompt `mysql` eseguire il comando `source` e passare al comando il nome del file dump del database per caricare i dati nell'istanza database Amazon RDS.

   - Per il formato SQL, utilizzare il comando seguente.

     ```
     mysql> source backup.sql;
     ```

   - Per il formato con testo delimitato, crea innanzitutto il database, se non usi il database predefinito creato al momento dell'impostazione del database Amazon RDS.

     ```
     mysql> create database database_name;
     mysql> use database_name;
     ```

     Quindi creare le tabelle.

     ```
     mysql> source table1.sql
     mysql> source table2.sql
     etc...
     ```

     Infine, importare i dati.

     ```
     mysql> LOAD DATA LOCAL INFILE 'table1.txt' INTO TABLE table1 FIELDS TERMINATED BY
      ',' ENCLOSED BY '"' LINES TERMINATED BY '0x0d0a';
     mysql> LOAD DATA LOCAL INFILE 'table2.txt' INTO TABLE table2 FIELDS TERMINATED BY
      ',' ENCLOSED BY '"' LINES TERMINATED BY '0x0d0a';
     etc...
     ```

     Per migliorare le prestazioni, puoi eseguire queste operazioni in parallelo da più connessioni, in modo che tutte le tabelle vengano create e caricate contemporaneamente.

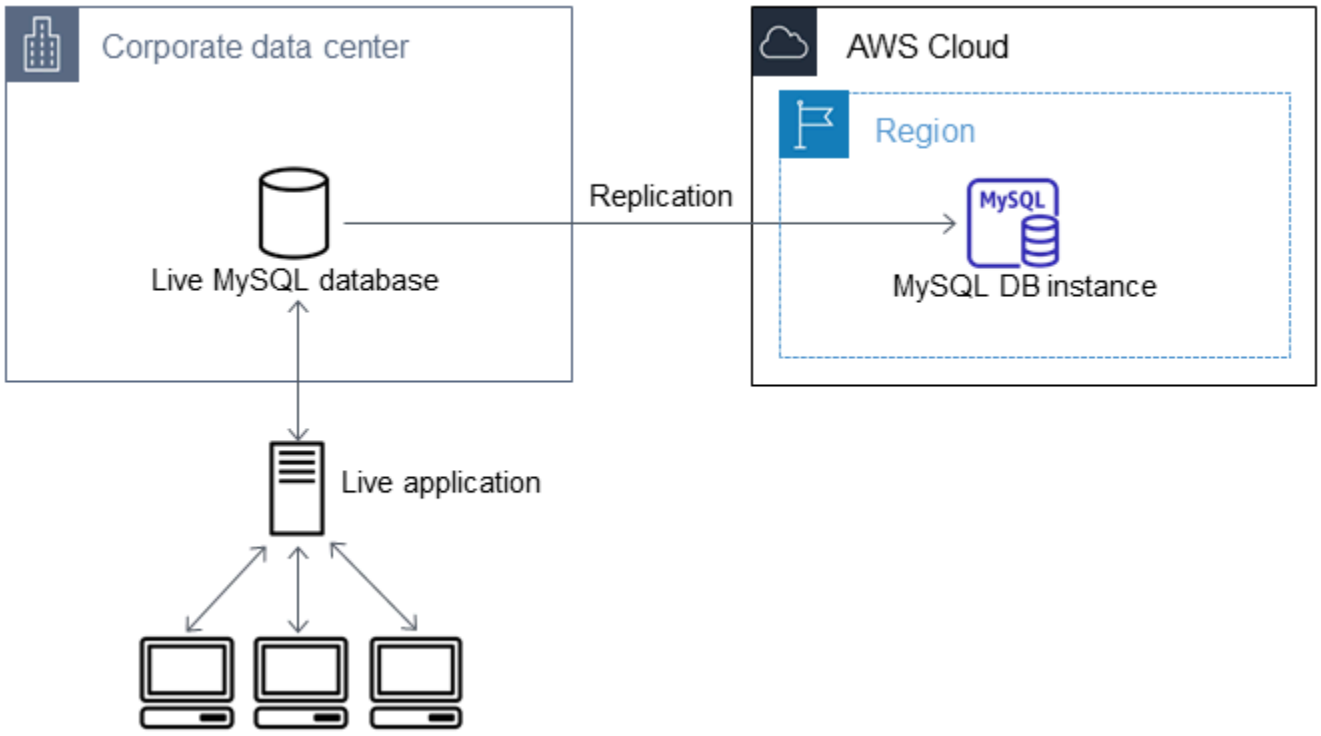
     > **Note**
     >
     > Se hai utilizzato opzioni di formattazione dei dati con mysqldump quando hai inizialmente scaricato la tabella, assicurati di utilizzare le stesse opzioni per garantire una corretta interpretazione del contenuto del file di `LOAD DATA LOCAL INFILE` dati.

8. Eseguite una semplice `SELECT` query su una o due tabelle del database importato per verificare che l'importazione sia avvenuta correttamente.

Se non hai più bisogno dell'istanza Amazon EC2 utilizzata in questa procedura, interrompi l'istanza EC2 per ridurre l'utilizzo delle risorse. AWS Per terminare un'istanza EC2, consulta Cessazione di un'istanza nella Guida per l'utente di Amazon EC2.

Replica tra il database esterno e un nuovo database Amazon RDS

È probabile che il database di origine sia stato aggiornato durante la copia e il trasferimento dei dati nel database MariaDB o MySQL. Pertanto, puoi utilizzare la replica per portare il database copiato con il database up-to-date di origine.



Le autorizzazioni necessarie per avviare la replica in un database Amazon RDS sono limitate e non disponibili per l'utente master Amazon RDS. Per questo motivo, assicurati di usare il comando mysql.rds_set_external_master o mysql.rds_set_external_master_gtid di Amazon RDS per configurare la replica e il comando mysql.rds_start_replication per avviare la replica tra il database attivo e il database Amazon RDS.

Per avviare la replica

In precedenza, hai attivato il log binario e impostato un ID server univoco per il database di origine. Ora puoi impostare il database Amazon RDS come replica, utilizzando il database live come istanza di replica di origine.

1. Nella Console di gestione Amazon RDS aggiungi l'indirizzo IP del server che ospita il database di origine al gruppo di sicurezza VPC per il database Amazon RDS. Per ulteriori informazioni sulla modifica di un gruppo di sicurezza VPC, consulta Gruppi di sicurezza per il VPC nella Guida per l'utente di Amazon Virtual Private Cloud.

   Potrebbe essere necessario configurare anche la rete locale per consentire le connessioni dall'indirizzo IP del database Amazon RDS, in modo da poter comunicare con l'istanza di origine. Per individuare l'indirizzo IP del database Amazon RDS, utilizza il comando `host`.

   ```
   host rds_db_endpoint
   ```

   Il nome host corrisponde al nome DNS dell'endpoint del database Amazon RDS, ad esempio `myinstance.123456789012.us-east-1.rds.amazonaws.com`. Puoi trovare il valore dell'endpoint è disponibile nei dettagli dell'istanza, nella console di gestione Amazon RDS.

2. Utilizzando il client scelto, eseguire la connessione all'istanza di origine e creare un utente da utilizzare per la replica. Questo account viene utilizzato unicamente per la replica e deve essere limitato al dominio personale per aumentare la sicurezza. Di seguito è riportato un esempio.

   MySQL 5.5, 5.6 e 5.7

   ```
   CREATE USER 'repl_user'@'mydomain.com' IDENTIFIED BY 'password';
   ```

   MySQL 8.0

   ```
   CREATE USER 'repl_user'@'mydomain.com' IDENTIFIED WITH mysql_native_password BY
    'password';
   ```

   > **Note**
   >
   > Specifica credenziali diverse dai prompt mostrati qui come best practice per la sicurezza.

3. Per l'istanza di origine, concedere i privilegi `REPLICATION CLIENT` e `REPLICATION SLAVE` all'utente di replica. Per concedere ad esempio i privilegi `REPLICATION CLIENT` e `REPLICATION SLAVE` su tutti i database per l'utente "`repl_user`" del proprio dominio, eseguire questo comando.

   MySQL 5.5, 5.6 e 5.7

```
GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user'@'mydomain.com'
 IDENTIFIED BY 'password';
```

MySQL 8.0

```
GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user'@'mydomain.com';
```

> **Note**
>
> Specifica credenziali diverse dai prompt mostrati qui come best practice per la sicurezza.

4. Se per creare il file di backup è stato usato il formato SQL e l'istanza esterna non è MariaDB 10.0.24 o superiore, controllare il contenuto del file.

   ```
   cat backup.sql
   ```

   Il file include un commento `CHANGE MASTER TO` che contiene il nome e la posizione del file di log principale. Il commento si trova nel file di backup, se è stata utilizzata l'opzione `--master-data` con mysqldump. Prendere nota dei valori per `MASTER_LOG_FILE` e `MASTER_LOG_POS`.

   ```
   --
   -- Position to start replication or point-in-time recovery from
   --

   -- CHANGE MASTER TO MASTER_LOG_FILE='mysql-bin-changelog.000031', MASTER_LOG_POS=107;
   ```

   Se per creare il file di backup è stato usato il formato con testo delimitato e l'istanza esterna non è MariaDB 10.0.24 o superiore, si dovrebbe già disporre delle coordinate del log binario dalla fase 1 della procedura descritta nella sezione “Per creare una copia di backup del database esistente” di questo articolo.

   Se l'istanza esterna è MariaDB 10.0.24 o superiore, si dovrebbe già disporre del GTID da cui avviare la replica dalla fase 2 della procedura descritta nella sezione “Per creare una copia di backup del database esistente” di questo articolo.

5. Definisci il database Amazon RDS come replica. Se l'istanza esterna non è MariaDB 10.0.24 o versioni successive, connettiti al database Amazon RDS come utente master e identifica il database di origine come istanza di replica di origine usando il comando

mysql.rds_set_external_master. Se si dispone di un file di backup in formato SQL, utilizzare il nome e la posizione del file log principale, recuperati nella fase precedente. Se invece è stato usato il formato con testo delimitato, utilizzare il nome e la posizione determinati al momento di creare i file di backup. Di seguito è riportato un esempio.

```
CALL mysql.rds_set_external_master ('myserver.mydomain.com', 3306,
    'repl_user', 'password', 'mysql-bin-changelog.000031', 107, 0);
```

> **Note**
>
> Specifica credenziali diverse dai prompt mostrati qui come best practice per la sicurezza.

Se l'istanza esterna è MariaDB 10.0.24 o versioni successive, connettiti al database Amazon RDS come utente master e identifica il database di origine come istanza di replica di origine usando il comando mysql.rds_set_external_master_gtid. Utilizzare il GTID determinato nel passaggio 2 della procedura descritta nella sezione “Per creare una copia di backup del database esistente” di questo articolo. Di seguito è riportato un esempio.

```
CALL mysql.rds_set_external_master_gtid ('source_server_ip_address', 3306,
 'ReplicationUser', 'password', 'GTID', 0);
```

`source_server_ip_address` è l' indirizzo IP dell'istanza di replica di origine. Al momento, gli indirizzi DNS privati di EC2 non sono supportati.

> **Note**
>
> Specifica credenziali diverse dai prompt mostrati qui come best practice per la sicurezza.

6. Nel database Amazon RDS esegui il comando mysql.rds_start_replication per avviare la replica.

```
CALL mysql.rds_start_replication;
```

7. Sul database Amazon RDS, esegui il comando SHOW REPLICA STATUS per determinare quando la replica è up-to-date con l'istanza di replica di origine. I risultati del comando `SHOW REPLICA STATUS` includono il campo `Seconds_Behind_Master`. Quando il

`Seconds_Behind_Master` campo restituisce 0, la replica si trova up-to-date con l'istanza di replica di origine.

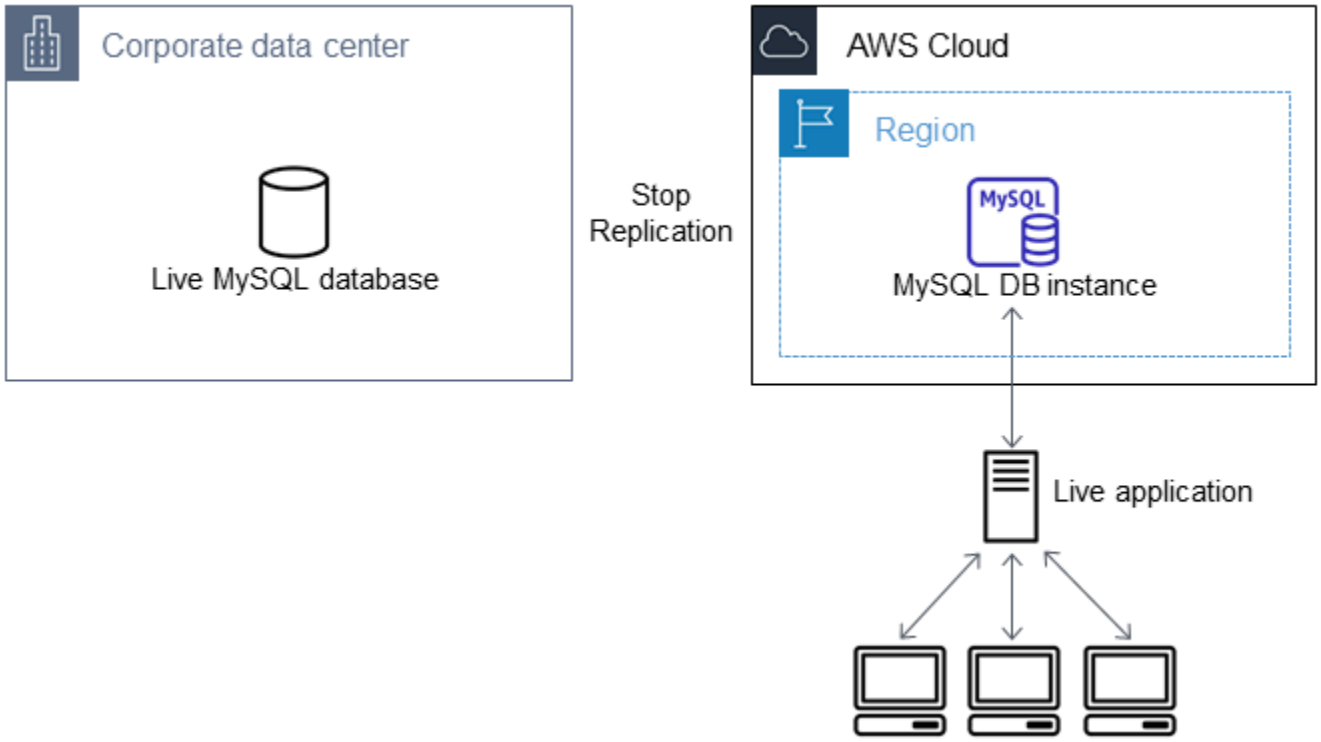
> **Note**
>
> Versioni precedenti di MySQL utilizzate `SHOW SLAVE STATUS` al posto di `SHOW REPLICA STATUS`. Se si utilizza una versione MySQL prima della 8.0.23, utilizzare `SHOW SLAVE STATUS`.

Per un'istanza database MariaDB 10.5, 10.6 o 10.11, esegui la procedura mysql.rds_replica_status anziché il comando MySQL.

8. Una volta installato il database Amazon RDS up-to-date, attiva i backup automatici in modo da poter ripristinare il database, se necessario. È possibile attivare o modificare i backup automatici per il database Amazon RDS tramite la Console di gestione Amazon RDS. Per ulteriori informazioni, consulta Introduzione ai backup.

Reindirizzamento di un'applicazione attiva nell'istanza di Amazon RDS

Dopo che il up-to-date database MariaDB o MySQL è con l'istanza di replica di origine, ora puoi aggiornare la tua applicazione live per utilizzare l'istanza Amazon RDS.



Per reindirizzare l'applicazione live al database MariaDB o MySQL e arrestare la replica

1. Per aggiungere il gruppo di sicurezza VPC per il database Amazon RDS, immetti l'indirizzo IP del server che ospita l'applicazione. Per ulteriori informazioni sulla modifica di un gruppo di sicurezza VPC, consulta Gruppi di sicurezza per il VPC nella Guida per l'utente di Amazon Virtual Private Cloud.
2. Verifica che il `Seconds_Behind_Master` campo nei risultati del comando SHOW REPLICA STATUS sia 0, il che indica che la replica è con l'istanza di replica di origine. up-to-date

```
SHOW REPLICA STATUS;
```

> **Note**
>
> Versioni precedenti di MySQL utilizzate `SHOW SLAVE STATUS` al posto di `SHOW REPLICA STATUS`. Se si utilizza una versione MySQL prima della 8.0.23, utilizzare `SHOW SLAVE STATUS`.

Per un'istanza database MariaDB 10.5, 10.6 o 10.11, esegui la procedura mysql.rds_replica_status anziché il comando MySQL.

3. Chiudere tutte le connessioni all'origine quando le loro transazioni sono complete.
4. Aggiorna l'applicazione per usare il database Amazon RDS. In genere, l'aggiornamento prevede la modifica delle impostazioni di connessione per identificare il nome host e la porta del database Amazon RDS, l'account utente e la password per eseguire la connessione e il database da utilizzare.
5. Effettua la connessione all'istanza database.

   Per un cluster database multi-AZ, connettiti all'istanza database di scrittura.
6. Interrompere la replica per l'istanza Amazon RDS tramite il comando mysql.rds_stop_replication.

```
CALL mysql.rds_stop_replication;
```

7. Esegui il comando mysql.rds_reset_external_master nel database Amazon RDS per reimpostare la configurazione della replica in modo che l'istanza non venga più identificata come replica.

```
CALL mysql.rds_reset_external_master;
```

8. Attivare le caratteristiche aggiuntive di Amazon RDS, quali il supporto Multi-AZ e le repliche di lettura. Per ulteriori informazioni, consultare Configurazione e gestione di un'implementazione multi-AZ e Uso delle repliche di lettura dell'istanza database.

Importazione dei dati da qualsiasi origine a un'istanza database MariaDB o MySQL

Consigliamo di creare snapshot DB dell'istanza database Amazon RDS di destinazione prima e dopo il caricamento dei dati. Le snapshot DB di Amazon RDS sono backup completi della tua istanza database che puoi utilizzare per ripristinare la tua istanza database in uno stato noto. Quando avvii una snapshot DB, le operazioni I/O dell'istanza database vengono temporaneamente sospese per il backup.

Creando una snapshot DB immediatamente prima di caricare i dati ti consente di ripristinare il database allo stato precedente il caricamento, se fosse necessario. Una snapshot DB effettuata immediatamente dopo il caricamento evita la necessità di caricare nuovamente i dati in caso di problemi e può essere utilizzata per inizializzare nuove istanze database.

Nell'elenco seguente è indicata la procedura da eseguire. Ciascun passaggio della procedura è descritto in modo dettagliato di seguito.

1. Creazione di file flat contenenti i dati da caricare.
2. Arresto delle applicazioni che accedono all'istanza database di destinazione.
3. Creazione di una snapshot DB.
4. Valuta se disattivare i backup automatici di Amazon RDS.
5. Carica i dati.
6. Riattivazione dei backup automatici.

Fase 1: Creazione di file flat contenenti i dati da caricare

Per salvare i dati da caricare, utilizza un formato comune, come ad esempio valori separati da virgola (CSV). Ciascuna tabella deve possedere il proprio file. Non è possibile combinare i dati di più tabelle nello stesso file. Devi fornire a ciascun file lo stesso nome della tabella corrispondente. Il file può avere qualsiasi estensione. Ad esempio, se il nome della tabella è `sales`, il nome del file potrebbe essere `sales.csv` o `sales.txt`, ma non `sales_01.csv`.

Quando possibile, ordina i dati in base alla chiave primaria della tabella da caricare. In questo modo i tempi di caricamento risultano significativamente più rapidi e si riduce il consumo di spazio su disco.

La velocità e l'efficienza di questa procedura dipende dalla capacità di mantenere contenute le dimensioni dei file. Se le dimensioni di un qualsiasi file (non compresso) superano 1 GiB, suddividilo in più file da caricare separatamente.

Nei sistemi di tipo Unix (incluso Linux), puoi utilizzare il comando `split`. Ad esempio, il comando seguente divide il file `sales.csv` in vari file con dimensioni inferiori a 1 GiB. Le divisioni vengono effettuate solo sulle interruzioni di riga (-C 1024m). I nuovi file sono denominati `sales.part_00`, `sales.part_01` e così via.

```
split -C 1024m -d sales.csv sales.part_
```

Utility simili sono disponibili anche per altri sistemi operativi.

Fase 2: Arresto delle applicazioni che accedono all'istanza database di destinazione

Prima di avviare il caricamento di grandi quantità di dati, interrompie le attività di tutte le applicazioni che accedono all'istanza database in cui intendi eseguire il caricamento. Questa operazione è particolarmente consigliata se le altre sessioni modificano le tabelle caricate o quelle di riferimento. In questo modo, puoi ridurre i rischi di violazione dei vincoli e ottimizzare le prestazioni durante

il caricamento. Inoltre, diventa possibile ripristinare l'istanza database al punto immediatamente precedente il caricamento, senza perdere le modifiche apportate dai processi che non sono coinvolti nell'operazione di caricamento.

Ovviamente, ci sono casi in cui l'esecuzione di questa operazione risulta impossibile o poco pratica. Se puoi evitare che alcune applicazioni accedano all'istanza database prima del caricamento, prendi tutte le misure necessarie per assicurare la disponibilità e l'integrità dei dati. Tali misure dipendono in larga parte dal tipo specifico di utilizzo e dai requisiti del sito.

Fase 3: Creazione di una snapshot DB

Se desideri caricare i dati in una nuova istanza database priva di dati, puoi ignorare questa parte. In caso contrario, la creazione di uno snapshot DB dell'istanza database ti consente di ripristinare l'istanza database nel punto immediatamente precedente al caricamento, se fosse necessario. Come spiegato in precedenza, quando avvii uno snapshot DB, le operazioni I/O dell'istanza database vengono sospese per alcuni minuti, mentre ha luogo il backup.

L'esempio seguente utilizza il AWS CLI `create-db-snapshot` comando per creare uno snapshot DB dell'`AcmeRDS`istanza e assegnare all'istantanea del DB l'identificatore. `"preload"`

PerLinux, o: macOS Unix

```
aws rds create-db-snapshot \
    --db-instance-identifier AcmeRDS \
    --db-snapshot-identifier preload
```

Per Windows:

```
aws rds create-db-snapshot ^
    --db-instance-identifier AcmeRDS ^
    --db-snapshot-identifier preload
```

Puoi utilizzare anche la funzione di ripristino da snapshot DB per creare istanze database di prova in cui eseguire test o annullare modifiche apportate durante il caricamento.

Ricorda che il ripristino di un database da una snapshot DB crea una nuova istanza database che, come tutte le istanze database, possiede un identificatore e un endpoint univoci. Per ripristinare l'istanza database senza modificare l'endpoint, devi innanzitutto eliminare l'istanza database, in modo da poter riutilizzare l'endpoint.

Ad esempio, per creare un'istanza database in cui eseguire test di vario tipo, devi assegnare all'istanza database il proprio identificatore. Nell'esempio, l'identificatore è `AcmeRDS-2`". L'esempio si connette all'istanza database utilizzando l'endpoint associato a `AcmeRDS-2`.

Per LinuxmacOS, oUnix:

```
aws rds restore-db-instance-from-db-snapshot \
    --db-instance-identifier AcmeRDS-2 \
    --db-snapshot-identifier preload
```

Per Windows:

```
aws rds restore-db-instance-from-db-snapshot ^
    --db-instance-identifier AcmeRDS-2 ^
    --db-snapshot-identifier preload
```

Per riutilizzare l'endpoint esistente, innanzitutto elimina l'istanza database e fornisci al database ripristinato lo stesso identificatore.

Per LinuxmacOS, oUnix:

```
aws rds delete-db-instance \
    --db-instance-identifier AcmeRDS \
    --final-db-snapshot-identifier AcmeRDS-Final

aws rds restore-db-instance-from-db-snapshot \
    --db-instance-identifier AcmeRDS \
    --db-snapshot-identifier preload
```

Per Windows:

```
aws rds delete-db-instance ^
    --db-instance-identifier AcmeRDS ^
    --final-db-snapshot-identifier AcmeRDS-Final

aws rds restore-db-instance-from-db-snapshot ^
    --db-instance-identifier AcmeRDS ^
    --db-snapshot-identifier preload
```

L'esempio precedente crea uno snapshot DB finale dell'istanza database prima di eliminarla. Questo passaggio è facoltativo, ma è consigliato.

Fase 4: Eventuale disattivazione dei backup automatici di Amazon RDS

> **Warning**
>
> Non disattivare i backup automatici se è necessario eseguire il point-in-time ripristino.

La disattivazione dei backup automatici cancella tutti i backup esistenti, quindi il point-in-time ripristino non è possibile dopo la disattivazione dei backup automatici. La disattivazione dei backup automatici serve a ottimizzare le prestazioni, ma non è indispensabile per il caricamento dei dati. Gli snapshot DB manuali non sono influenzati dalla disattivazione dei backup automatici. Tutti gli snapshot DB esistenti rimangono disponibili per il ripristino.

La disattivazione dei backup automatici velocizza il tempo di caricamento di circa il 25% e riduce la quantità di spazio richiesto. Se devi caricare dati in una nuova istanza database che non contiene altri dati, la disattivazione dei backup rappresenta un'ottima soluzione per velocizzare il caricamento ed evitare di occupare troppo spazio con i backup. Tuttavia, in alcuni casi, potresti pianificare di caricare i dati in un'istanza database che contiene già altri dati. In tal caso, soppesate i vantaggi della disattivazione dei backup rispetto all'impatto della perdita della capacità di esecuzione. point-in-time-recovery

Per impostazione predefinita, i backup sono attivati per le istanze database (con un periodo di conservazione di un giorno). Per disabilitare i backup automatici, imposta il periodo di conservazione del backup su zero. Dopo il caricamento potrai riattivare i backup automatici impostando il periodo di conservazione su un valore diverso da zero. Per attivare o disattivare i backup, Amazon RDS chiude l'istanza database e la riavvia in modo da attivare o disattivare i log MariaDB o MySQL.

Utilizzate il AWS CLI `modify-db-instance` comando per impostare la conservazione dei backup su zero e applicare immediatamente la modifica. Per impostare il periodo di retention su zero è necessario riavviare l'istanza database, quindi prima di continuare dovrai attendere il completamento del riavvio.

Per LinuxmacOS, oUnix:

```
aws rds modify-db-instance \
    --db-instance-identifier AcmeRDS \
    --apply-immediately \
    --backup-retention-period 0
```

Per Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier AcmeRDS ^
    --apply-immediately ^
    --backup-retention-period 0
```

Puoi controllare lo stato della tua istanza DB con il AWS CLI `describe-db-instances` comando. Nell'esempio seguente viene visualizzato lo stato dell'istanza database dell'istanza database `AcmeRDS`.

```
aws rds describe-db-instances --db-instance-identifier AcmeRDS --query "*[].
{DBInstanceStatus:DBInstanceStatus}"
```

Quando lo stato dell'istanza DB è `available`, si è pronti per procedere.

Fase 5: Caricamento dei dati

Usa l'istruzione `LOAD DATA LOCAL INFILE` MySQL per leggere le righe dai tuoi file flat nelle tabelle del database.

L'esempio seguente mostra come caricare i dati da un file denominato `sales.txt` in una tabella denominata `Sales` nel database.

```
mysql> LOAD DATA LOCAL INFILE 'sales.txt' INTO TABLE Sales FIELDS TERMINATED BY ' '
 ENCLOSED BY '' ESCAPED BY '\\';
Query OK, 1 row affected (0.01 sec)
Records: 1  Deleted: 0  Skipped: 0  Warnings: 0
```

Per ulteriori informazioni sulla `LOAD DATA` dichiarazione, consulta la documentazione di MySQL.

Fase 6: Riattivazione dei backup automatici di Amazon RDS

Al termine del caricamento, riattiva i backup automatici di Amazon RDS reimpostando il tempo di conservazione del backup sul valore originale. Come indicato in precedenza, Amazon RDS riavvia l'istanza database, interrompendo brevemente le attività.

L'esempio seguente utilizza il AWS CLI `modify-db-instance` comando per attivare i backup automatici per l'istanza `AcmeRDS` DB e impostare il periodo di conservazione su un giorno.

Per LinuxmacOS, oUnix:

```
aws rds modify-db-instance \
    --db-instance-identifier AcmeRDS \
    --backup-retention-period 1 \
    --apply-immediately
```

Per Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier AcmeRDS ^
    --backup-retention-period 1 ^
    --apply-immediately
```

Uso della replica MariaDB in Amazon RDS

Generalmente, per configurare la replica tra le istanze database di Amazon RDS si utilizzano repliche di lettura. Per informazioni generali sulle repliche di lettura, consulta Uso delle repliche di lettura dell'istanza database. Per informazioni specifiche sull'uso di repliche di lettura in Amazon RDS for MariaDB, consulta Uso di repliche di lettura MariaDB.

Puoi anche configurare la replica in base alle coordinate del log binario per un'istanza database MariaDB. Per le istanze MariaDB, puoi configurare la replica anche in base a ID transazione globali (GTID), per una maggiore sicurezza contro gli arresti anomali. Per ulteriori informazioni, consultare Configurazione della replica basata su GTID con un'istanza di origine esterna.

Quelle elencate di seguito sono altre opzioni di replica disponibili con RDS for MariaDB:

- Puoi impostare la replica fra un'istanza database RDS for MariaDB e MySQL oppure MariaDB, che è esterna ad Amazon RDS. Per ulteriori informazioni sulla configurazione della replica con un'origine esterna, consulta Configurazione della replica della posizione del file di log binario con un'istanza di origine esterna..
- Puoi configurare la replica per l'importazione di database da un'istanza MySQL o MariaDB esterna ad Amazon RDS o per l'esportazione di database a tali istanze. Per ulteriori informazioni, consulta Importazione dei dati in un'istanza database MariaDB o MySQL di Amazon RDS, riducendo i tempi di inattività e Esportazione di dati da un'istanza database MySQL tramite la replica.

Per qualsiasi opzione di replica, puoi utilizzare la replica basata su riga, basata su istruzioni o quella mista. La replica basata su riga replica solamente le righe modificate che risultano da un'istruzione SQL. La replica basata su istruzioni replica l'intera istruzione SQL. La replica mista utilizza la replica basata su istruzione quando possibile, ma passa alla replica basata su riga quando vengono eseguite le istruzioni SQL che non sono sicure per la replica basata su istruzione. Nella maggior parte dei casi, si consiglia l'utilizzo della replica mista. Il formato di log binario dell'istanza database determina se la replica è basata su riga, su istruzione o è mista. Per informazioni sull'impostazione del formato di log binario, consulta Formato di registrazione binario.

Argomenti

Uso di repliche di lettura MariaDB

Questa sezione contiene informazioni specifiche sull'utilizzo delle repliche di lettura su Amazon RDS for MariaDB. Per informazioni generali sulle repliche di lettura e istruzioni su come usarle, consulta Uso delle repliche di lettura dell'istanza database.

Argomenti

Configurazione delle repliche di lettura con MariaDB

Prima di poter usare un'istanza database MariaDB come origine della replica, devi attivare i backup automatici nell'istanza database di origine impostando il periodo di retention dei backup su un valore diverso da 0. Questo requisito si applica anche a una replica di lettura che rappresenta l'istanza database di origine per un'altra replica di lettura.

È possibile creare fino a 15 repliche di lettura da un'istanza database nella stessa regione. Per un efficace funzionamento della replica, ogni replica di lettura deve avere la stessa quantità di risorse di calcolo e di storage dell'istanza database di origine. Se si dimensiona l'istanza database di origine, si devono dimensionare anche le repliche di lettura.

RDS per MariaDB supporta le repliche di lettura a cascata. Per informazioni su come configurare le repliche di lettura a cascata, consulta Utilizzo di repliche di lettura a cascata con RDS per MariaDB.

Puoi eseguire più operazioni di creazione ed eliminazione di repliche di lettura simultanee che fanno riferimento alla stessa istanza database di origine. Quando esegui queste operazioni, rimani entro il limite delle 15 repliche di lettura per ogni istanza di origine.

Configurazione dei filtri di replica con MariaDB

Puoi utilizzare i filtri di replica per specificare quali database e tabelle vengono replicati con una replica di lettura. I filtri di replica possono includere database e tabelle nella replica o escluderli dalla replica.

Di seguito sono riportati alcuni casi d'uso per i filtri di replica:

- Per ridurre le dimensioni di una replica di lettura. Con il filtro di replica è possibile escludere i database e le tabelle che non sono necessari nella replica di lettura.
- Per escludere database e tabelle dalle repliche di lettura per motivi di sicurezza.
- Per replicare database e tabelle diversi per casi d'uso specifici in repliche di lettura diverse. Ad esempio, è possibile utilizzare repliche di lettura specifiche per l'analisi o la condivisione.
- Per un'istanza database che dispone di repliche di lettura in diverse Regioni AWS, per replicare database o tabelle diversi in diverse Regioni AWS.

> **Note**
>
> Puoi utilizzare i filtri di replica anche per specificare i database e le tabelle che vengono replicati con un'istanza database MariaDB primaria configurata come replica in una topologia di replica in ingresso. Per ulteriori informazioni su questa configurazione, consulta Configurazione della replica della posizione del file di log binario con un'istanza di origine esterna..

Argomenti

- Impostazione dei parametri di filtro della replica Amazon RDS for MariaDB
- Limitazioni di filtro delle repliche per RDS for MariaDB
- Esempi di filtri di replica per RDS for MariaDB
- Visualizzazione dei filtri di replica per una replica di lettura

Impostazione dei parametri di filtro della replica Amazon RDS for MariaDB

Per configurare i filtri di replica, impostare i seguenti parametri di filtro replica sulla replica di lettura:

- `replicate-do-db` – Replicare le modifiche ai database specificati. Quando si imposta questo parametro per una replica di lettura, vengono replicati solo i database specificati nel parametro.
- `replicate-ignore-db` – Non replicare le modifiche ai database specificati. Quando il parametro `replicate-do-db` è impostato per una replica di lettura, questo parametro non viene valutato.
- `replicate-do-table` – Replicare le modifiche alle tabelle specificate. Quando si imposta questo parametro per una replica di lettura, vengono replicate solo le tabelle specificate nel parametro. Inoltre, quando viene impostato il parametro `replicate-do-db` o `replicate-ignore-db`, assicurarsi di includere il database che include le tabelle specificate nella replica con la replica di lettura.
- `replicate-ignore-table` – Non replicare le modifiche alle tabelle specificate. Quando il parametro `replicate-do-table` è impostato per una replica di lettura, questo parametro non viene valutato.
- `replicate-wild-do-table` – Replicare le tabelle in base ai modelli di nome del database e della tabella specificati. I caratteri jolly `%` e `_` sono supportati. Quando è impostato il parametro `replicate-do-db` o `replicate-ignore-db`, assicurarsi di includere il database che include le tabelle specificate nella replica con la replica di lettura.
- `replicate-wild-ignore-table` – Non replicare le tabelle in base ai modelli di nomi di database e tabella specificati. I caratteri jolly `%` e `_` sono supportati. Quando è impostato il parametro `replicate-do-table` o `replicate-wild-do-table` per una replica di lettura, questo parametro non viene valutato.

I parametri vengono valutati nell'ordine in cui sono elencati. Per maggiori informazioni sul funzionamento di questi parametri, consulta la documentazione di MariaDB.

Per impostazione predefinita, ognuno di questi parametri ha un valore vuoto. In ogni replica di lettura, è possibile utilizzare questi parametri per impostare, modificare ed eliminare i filtri di replica. Quando viene impostato uno di questi parametri, è necessario separare ogni filtro dagli altri con una virgola.

È possibile utilizzare i caratteri jolly `%` e `_` nei parametri `replicate-wild-do-table` e `replicate-wild-ignore-table`. Il carattere jolly `%` corrisponde a un numero qualsiasi di caratteri e il carattere jolly `_` corrisponde a un solo carattere.

Il formato di registrazione binaria dell'istanza database di origine è importante per la replica perché determina il record delle modifiche ai dati. L'impostazione del parametro `binlog_format` determina se la replica è basata su righe o basata su dichiarazione. Per ulteriori informazioni, consulta Formato di registrazione binario.

> **Note**
>
> Tutte le istruzioni DDL (Data Definition Language) vengono replicate come istruzioni, indipendentemente dall'impostazione `binlog_format` dell'istanza database di origine.

Limitazioni di filtro delle repliche per RDS for MariaDB

Le seguenti limitazioni si applicano al filtro di replica per RDS for MariaDB:

- Ogni parametro di filtro della replica ha un limite di 2.000 caratteri.
- Le virgole non sono supportate nei filtri di replica.
- Le opzioni MariaDB `binlog_do_db` e `binlog_ignore_db` per il filtro dei log binari non sono supportate.
- Il filtro delle repliche non supporta le transazioni XA.

  Per ulteriori informazioni, consulta Restrizioni sulle transazioni XA nella documentazione di MySQL.
- Il filtro di replica non è supportato per RDS for MariaDB 10.2.

Esempi di filtri di replica per RDS for MariaDB

Per configurare il filtro di replica per una replica di lettura, modificare i parametri di filtro replica nel gruppo di parametri associato alla replica di lettura.

> **Note**
>
> Non è consentito modificare un gruppo di parametri predefinito. Se la replica di lettura usa un gruppo di parametri predefinito, creare un nuovo gruppo di parametri e associarlo alla replica di lettura. Per ulteriori informazioni sui gruppi di parametri database, consulta Utilizzo di gruppi di parametri.

È possibile impostare parametri in un gruppo di parametri utilizzando la AWS Management Console, la AWS CLI o l'API RDS. Per informazioni sull'estensione dei parametri consulta Modifica di parametri in un gruppo di parametri del database. Quando si impostano parametri in un gruppo di parametri, tutte le istanze DB associate al gruppo di parametri utilizzano le impostazioni dei parametri. Se si impostano i parametri di filtro della replica in un gruppo di parametri, assicurarsi che il gruppo di

parametri sia associato solo alle repliche di lettura. Lasciare vuoti i parametri di filtro di replica per le istanze database di origine.

Negli esempi seguenti vengono impostati i parametri utilizzando AWS CLI. In questi esempi si imposta `ApplyMethod` su `immediate` in modo che le modifiche ai parametri avvengano immediatamente dopo il completamento del comando della CLI. Se si desidera applicare una modifica in sospeso dopo il riavvio della replica di lettura, impostare `ApplyMethod` su `pending-reboot`.

Gli esempi seguenti impostano i filtri di replica:

- Including databases in replication
- Including tables in replication
- Including tables in replication with wildcard characters
- Escaping wildcard characters in names
- Excluding databases from replication
- Excluding tables from replication
- Excluding tables from replication using wildcard characters

Example Inclusione dei database nella replica

Nell'esempio seguente sono inclusi i database `mydb1` e `mydb2` nella replica. Quando si imposta `replicate-do-db` per una replica di lettura, vengono replicati solo i database specificati nel parametro.

Per LinuxmacOS, oUnix:

```
aws rds modify-db-parameter-group \
  --db-parameter-group-name myparametergroup \
  --parameters "[{"ParameterName": "replicate-do-db", "ParameterValue": "mydb1,mydb2",
 "ApplyMethod":"immediate"}]"
```

Per Windows:

```
aws rds modify-db-parameter-group ^
  --db-parameter-group-name myparametergroup ^
```

```
  --parameters "[{"ParameterName": "replicate-do-db", "ParameterValue": "mydb1,mydb2",
 "ApplyMethod":"immediate"}]"
```

Example Inclusione delle tabelle nella replica

Nell'esempio seguente sono incluse le tabelle `table1` e `table2` nel database `mydb1` nella replica.

Per LinuxmacOS, oUnix:

```
aws rds modify-db-parameter-group \
  --db-parameter-group-name myparametergroup \
  --parameters "[{"ParameterName": "replicate-do-table", "ParameterValue":
 "mydb1.table1,mydb1.table2", "ApplyMethod":"immediate"}]"
```

Per Windows:

```
aws rds modify-db-parameter-group ^
  --db-parameter-group-name myparametergroup ^
  --parameters "[{"ParameterName": "replicate-do-table", "ParameterValue":
 "mydb1.table1,mydb1.table2", "ApplyMethod":"immediate"}]"
```

Example Inclusione di tabelle nella replica utilizzando caratteri jolly

Nell'esempio seguente sono incluse tabelle con nomi che iniziano con `orders` e `returns` nel database `mydb` nella replica.

Per LinuxmacOS, oUnix:

```
aws rds modify-db-parameter-group \
  --db-parameter-group-name myparametergroup \
  --parameters "[{"ParameterName": "replicate-wild-do-table", "ParameterValue":
 "mydb.orders%,mydb.returns%", "ApplyMethod":"immediate"}]"
```

Per Windows:

```
aws rds modify-db-parameter-group ^
  --db-parameter-group-name myparametergroup ^
  --parameters "[{"ParameterName": "replicate-wild-do-table", "ParameterValue":
 "mydb.orders%,mydb.returns%", "ApplyMethod":"immediate"}]"
```

Example Escape di caratteri jolly nei nomi

Nell'esempio seguente viene illustrato come utilizzare il carattere escape \ per un carattere jolly che fa parte di un nome.

Si supponga di avere diversi nomi di tabelle nel database `mydb1` che iniziano con `my_tablee` si desidera includere queste tabelle nella replica. I nomi delle tabelle includono un carattere di sottolineatura, che è anche un carattere jolly, quindi l'esempio sfugge al carattere di sottolineatura nei nomi delle tabelle.

Per LinuxmacOS, oUnix:

```
aws rds modify-db-parameter-group \
  --db-parameter-group-name myparametergroup \
  --parameters "[{"ParameterName": "replicate-wild-do-table", "ParameterValue": "my
\_table%", "ApplyMethod":"immediate"}]"
```

Per Windows:

```
aws rds modify-db-parameter-group ^
  --db-parameter-group-name myparametergroup ^
  --parameters "[{"ParameterName": "replicate-wild-do-table", "ParameterValue": "my
\_table%", "ApplyMethod":"immediate"}]"
```

Example Esclusione di database dalla replica

Nell'esempio seguente vengono esclusi i database `mydb1` e `mydb2` dalla replica.

Per LinuxmacOS, oUnix:

```
aws rds modify-db-parameter-group \
  --db-parameter-group-name myparametergroup \
  --parameters "[{"ParameterName": "replicate-ignore-db", "ParameterValue":
 "mydb1,mydb2", "ApplyMethod":"immediate"}]"
```

Per Windows:

```
aws rds modify-db-parameter-group ^
```

```
  --db-parameter-group-name myparametergroup ^
  --parameters "[{"ParameterName": "replicate-ignore-db", "ParameterValue":
 "mydb1,mydb2", "ApplyMethod":"immediate"}]"
```

Example Esclusione di tabelle dalla replica

Nell'esempio seguente vengono escluse le tabelle `table1` e `table2` nel database `mydb1` dalla replica.

Per LinuxmacOS, oUnix:

```
aws rds modify-db-parameter-group \
  --db-parameter-group-name myparametergroup \
  --parameters "[{"ParameterName": "replicate-ignore-table", "ParameterValue":
 "mydb1.table1,mydb1.table2", "ApplyMethod":"immediate"}]"
```

Per Windows:

```
aws rds modify-db-parameter-group ^
  --db-parameter-group-name myparametergroup ^
  --parameters "[{"ParameterName": "replicate-ignore-table", "ParameterValue":
 "mydb1.table1,mydb1.table2", "ApplyMethod":"immediate"}]"
```

Example Esclusione di tabelle dalla replica utilizzando caratteri jolly

Nell'esempio seguente vengono escluse le tabelle con nomi che iniziano con `orders` e `returns` nel database `mydb` dalla replica.

Per LinuxmacOS, oUnix:

```
aws rds modify-db-parameter-group \
  --db-parameter-group-name myparametergroup \
  --parameters "[{"ParameterName": "replicate-wild-ignore-table", "ParameterValue":
 "mydb.orders%,mydb.returns%", "ApplyMethod":"immediate"}]"
```

Per Windows:

```
aws rds modify-db-parameter-group ^
```

```
  --db-parameter-group-name myparametergroup ^
  --parameters "[{"ParameterName": "replicate-wild-ignore-table", "ParameterValue":
"mydb.orders%,mydb.returns%", "ApplyMethod":"immediate"}]"
```

Visualizzazione dei filtri di replica per una replica di lettura

È possibile visualizzare i filtri di replica per una replica di lettura nei seguenti modi:

- Controllare le impostazioni dei parametri di filtro replica nel gruppo di parametri associato alla replica di lettura.

  Per istruzioni, consulta Visualizzazione dei valori dei parametri per un gruppo di parametri del database.
- In un client MariaDB, connettersi alla replica di lettura ed eseguire l'istruzione `SHOW REPLICA STATUS`.

  Nell'output, i campi seguenti mostrano i filtri di replica per la replica di lettura:
  - `Replicate_Do_DB`
  - `Replicate_Ignore_DB`
  - `Replicate_Do_Table`
  - `Replicate_Ignore_Table`
  - `Replicate_Wild_Do_Table`
  - `Replicate_Wild_Ignore_Table`

  Per ulteriori informazioni su questi campi, consulta Verifica dello stato della replica nella documentazione di MariaDB.

  > **Note**
  >
  > Le versioni precedenti di MariaDB utilizzavano `SHOW SLAVE STATUS` anziché `SHOW REPLICA STATUS`. Se si utilizza una versione di MariaDB precedente alla 10.5, utilizzare `SHOW SLAVE STATUS`.

Configurazione della replica ritardata con MariaDB

Puoi usare la replica ritardata come strategia per il disaster recovery. Con la replica ritardata puoi specificare il tempo minimo, in secondi, di ritardo della replica rispetto all'origine nella replica di

lettura. In caso di emergenza, come ad esempio l'eliminazione accidentale di una tabella, completa la seguente procedura per risolvere velocemente il problema:

- Arresta la replica sulla replica di lettura prima che la modifica che ha provocato il problema venga inviata.

  Per arrestare la replica, usa la procedura archiviata mysql.rds_stop_replication .
- Utilizza le istruzioni contenute in Promozione di una replica di lettura a istanza database standalone per promuovere la replica di lettura a nuova istanza database di origine.

> **Note**
>
> - La replica ritardata è supportata per MariaDB 10.6 e versioni successive.
> - Utilizza le procedure archiviate per configurare la replica ritardata. Non puoi configurare la replica ritardata tramite la AWS Management Console, la AWS CLI o l'API di Amazon RDS.
> - È possibile utilizzare la replica basata su identificatori di transazione globali (GTID) in una configurazione di replica ritardata.

Argomenti

- Configurazione della replica ritardata durante la creazione della replica di lettura
- Modifica della replica ritardata per una replica di lettura esistente
- Promozione di una replica di lettura

Configurazione della replica ritardata durante la creazione della replica di lettura

Per configurare la replica ritardata per eventuali repliche di lettura future create da un'istanza database, esegui la stored procedure mysql.rds_set_configuration con il parametro `target delay`.

Per configurare le replica ritardata durante la creazione della replica di lettura

1. Utilizzando un client MariaDB, connettersi all'istanza database MariaDB che sarà l'origine delle repliche di lettura come l'utente master.
2. Eseguire la procedura archiviata mysql.rds_set_configuration con il parametro `target delay`.

Ad esempio, eseguire la seguente procedura archiviata per specificare che la replica è ritardata per almeno un'ora (3.600 secondi) per le repliche di lettura create dall'istanza database corrente.

```
call mysql.rds_set_configuration('target delay', 3600);
```

> **Note**
>
> Dopo aver eseguito questa stored procedure, le eventuali repliche di lettura create tramite la AWS CLI o l'API di Amazon RDS vengono configurate con la replica ritardata per il numero specificato di secondi.

Modifica della replica ritardata per una replica di lettura esistente

Per modificare la replica ritardata per una replica di lettura esistente, esegui la stored procedure mysql.rds_set_source_delay.

Per modificare la replica ritardata per una replica di lettura esistente

1. Utilizzando un client MariaDB, connettersi alla replica di lettura come utente principale.
2. Usa la procedura archiviata mysql.rds_stop_replication per arrestare la replica.
3. Eseguire la procedura archiviata mysql.rds_set_source_delay.

   Ad esempio, eseguire la seguente stored procedure per specificare che la replica sulla replica di lettura è ritardata per almeno un'ora (3600 secondi).

   ```
   call mysql.rds_set_source_delay(3600);
   ```
4. Usare la procedura archiviata mysql.rds_start_replication per avviare la replica.

Promozione di una replica di lettura

Dopo l'arresto della replica, in uno scenario di disaster recovery, puoi promuovere la replica di lettura come nuova istanza database di origine. Per informazioni sulla promozione di una replica di lettura, consulta Promozione di una replica di lettura a istanza database standalone.

Aggiornamento di repliche di lettura con MariaDB

Le repliche di lettura sono progettate per supportare query di lettura, ma occasionalmente potrebbe essere necessario eseguire aggiornamenti. Ad esempio, potresti dover aggiungere un indice per accelerare i tipi specifici di query che accedono alla replica. Puoi abilitare gli aggiornamenti impostando il parametro `read_only` su 0 nel gruppo di parametri database per la replica di lettura.

Operare con le implementazioni Multi-AZ di repliche di lettura con MariaDB

Puoi creare una replica di lettura da implementazioni Single-AZ o Multi-AZ di istanze database. Puoi usare implementazioni Multi-AZ per migliorare la durabilità e la disponibilità di dati critici, ma non puoi usare l'istanza secondaria Multi-AZ per inviare query di sola lettura. Puoi invece creare repliche di lettura da istanze database Multi-AZ con traffico elevato per l'offload di query di sola lettura. Se viene eseguito il failover dell'istanza di origine di un'implementazione Multi-AZ all'istanza secondaria, tutte le repliche di lettura passeranno automaticamente a usare l'istanza secondaria (ora primaria) come origine della replica. Per ulteriori informazioni, consulta Configurazione e gestione di un'implementazione multi-AZ.

È possibile creare una replica di lettura come istanza database Multi-AZ. Amazon RDS crea una replica di standby in un'altra zona di disponibilità per il supporto del failover per la replica. La creazione della replica di lettura come un'istanza database Multi-AZ non dipende dal fatto che il database di origine sia un'istanza database Multi-AZ.

Utilizzo di repliche di lettura a cascata con RDS per MariaDB

RDS per MariaDB supporta le repliche di lettura a cascata. Con le repliche di lettura a cascata, puoi dimensionare le letture senza aggiungere sovraccarico all'istanza database RDS per MariaDB di origine.

Con le repliche di lettura a cascata, l'istanza database RDS per MariaDB invia i dati alla prima replica di lettura della catena. La replica di lettura invia quindi i dati alla seconda replica della catena e così via. Il risultato finale è che tutte le repliche di lettura nella catena includono le modifiche dall'istanza database RDS per MariaDB, ma senza sovraccaricare esclusivamente l'istanza database di origine.

È possibile creare una serie di fino a tre repliche di lettura in una catena da un'istanza database RDS per MariaDB di origine. Ad esempio, supponi di avere l'istanza database RDS per MariaDB `mariadb-main`. Puoi eseguire le operazioni indicate di seguito:

- A partire da `mariadb-main`, crea la prima replica di lettura nella catena, `read-replica-1`.

- Da `read-replica-1`, crea quindi la successiva replica di lettura nella catena, `read-replica-2`.
- Da `read-replica-2`, crea infine la terza replica di lettura nella catena, `read-replica-3`.

Non è possibile creare un'altra replica di lettura oltre la terza replica di lettura a cascata nella serie per `mariadb-main`. Una serie completa di istanze da un'istanza database RDS per MariaDB di origine fino alla fine di una serie di repliche di lettura a cascata può essere composta al massimo da quattro istanze database.

Affinché le repliche di lettura a cascata funzionino, ogni istanza database RDS per MariaDB di origine deve avere i backup automatici attivati. Per abilitare i backup automatici in una replica di lettura, crea prima di tutto la replica di lettura, quindi modificala in modo da abilitare i backup automatici. Per ulteriori informazioni, consulta Creazione di una replica di lettura.

Come per qualsiasi replica di lettura, puoi promuovere una replica di lettura appartenente a una cascata. La promozione di una replica di lettura all'interno di una catena di repliche di lettura rimuove la replica dalla catena. Ad esempio, supponi di voler spostare parte del carico di lavoro fuori dall'istanza database `mariadb-main` in una nuova istanza usata solo dal reparto contabile. Facendo riferimento alla catena di tre repliche di lettura dell'esempio, decidi di promuovere `read-replica-2`. La catena verrà modificata come segue:

- La promozione `read-replica-2` rimuove l'istanza dalla catena di replica.
  - Ora è un'istanza database completa di lettura/scrittura.
  - Continua a replicare su `read-replica-3`, proprio come prima della promozione.
- L'istanza `mariadb-main` continua a venire replicata su `read-replica-1`.

Per ulteriori informazioni sulla promozione delle repliche di lettura, consulta Promozione di una replica di lettura a istanza database standalone.

Monitoraggio delle repliche di lettura MariaDB

Per le repliche di lettura di MariaDB, puoi monitorare il ritardo di replica in Amazon CloudWatch visualizzando la metrica Amazon RDS. `ReplicaLag` Il parametro `ReplicaLag` segnala il valore del campo `Seconds_Behind_Master` del comando `SHOW REPLICA STATUS`.

> **Note**
>
> Le versioni precedenti di MariaDB utilizzavano `SHOW SLAVE STATUS` anziché `SHOW REPLICA STATUS`. Se si utilizza una versione di MariaDB precedente alla 10.5, utilizzare `SHOW SLAVE STATUS`.

Le cause comuni del ritardo di replica per MariaDB sono le seguenti:

- Interruzione della connessione di rete.
- Scrittura in tabelle con indici in una replica di lettura. Se il parametro `read_only` non è impostato su 0 nella replica di lettura, la replica può essere interrotta.
- Uso di un motore di storage non transazionale come MyISAM. La replica è supportata solo per il motore di storage InnoDB su MariaDB.

Quando il parametro `ReplicaLag` è 0, la replica ha raggiunto l'istanza del database di origine. Se il parametro `ReplicaLag` restituisce -1, la replica non è attualmente attiva. `ReplicaLag` = -1 equivale a `Seconds_Behind_Master` = `NULL`.

Avvio e arresto della replica con repliche di lettura MariaDB

Puoi arrestare e riavviare il processo di replica in un'istanza database Amazon RDS chiamando le stored procedure di sistema mysql.rds_stop_replication e mysql.rds_start_replication. Puoi procedere in questo modo quando esegui la replica tra due istanze Amazon RDS per operazioni a esecuzione prolungata, come la creazione di indici di grandi dimensioni. Devi arrestare e avviare la replica anche durante l'importazione o l'esportazione di database. Per ulteriori informazioni, consulta Importazione dei dati in un database Amazon RDS MariaDB o MySQL con tempi di inattività ridotti e Esportazione di dati da un'istanza database MySQL tramite la replica.

Se la replica viene arrestata per più di 30 giorni consecutivi, manualmente o a causa di un errore di replica, Amazon RDS termina la replica tra l'istanza database di origine e tutte le repliche di lettura. Questo avviene per evitare requisiti di storage maggiori sull'istanza database di origine e tempi di failover prolungati. L'istanza database della replica di lettura continua a essere disponibile. Tuttavia, la replica non può essere ripresa, perché i log binari richiesti dalla replica di lettura vengono eliminati dall'istanza database di origine una volta terminata la replica. Puoi creare una nuova replica di lettura per l'istanza database di origine per ristabilire la replica.

Risoluzione dei problemi relativi a una replica di lettura MariaDB

Le tecnologie di replica per MariaDB sono asincrone. Per questo motivo, devi occasionalmente aspettarti incrementi del parametro `BinLogDiskUsage` per l'istanza database di origine e del parametro `ReplicaLag` per la replica di lettura. Ad esempio, può verificarsi un elevato volume di scrittura in parallelo nell'istanza database di origine. Al contrario, le operazioni di scrittura nella replica di lettura vengono serializzate usando un singolo thread di I/O, causando un ritardo tra l'istanza di origine e la replica di lettura. Per ulteriori informazioni sulle repliche di sola lettura, consulta la panoramica della replica nella documentazione di MariaDB.

Puoi ridurre il ritardo tra gli aggiornamenti di un'istanza database di origine e i successivi aggiornamenti della replica di lettura in diversi modi, ad esempio:

- Dimensionando una replica di lettura in modo che dimensioni di storage e classe dell'istanza database siano equivalenti all'istanza database di origine.
- Assicurandoti che le impostazioni dei parametri nei gruppi di parametri database usati dall'istanza database di origine e dalla replica di lettura siano compatibili. Per ulteriori informazioni e un esempio, consulta la discussione sul parametro `max_allowed_packet` più avanti in questa sezione.

Amazon RDS monitora lo stato delle repliche di lettura e aggiorna il campo `Replication State` dell'istanza della replica di lettura con il valore `Error` se la replica viene arrestata per qualsiasi motivo. Un possibile esempio è quando query DML in esecuzione nella replica di lettura sono in conflitto con gli aggiornamenti eseguiti nell'istanza database di origine.

Puoi esaminare i dettagli dell'errore associato generato dal motore MariaDB visualizzando il campo `Replication Error`. Vengono generati anche eventi che indicano lo stato della replica di lettura, inclusi RDS-EVENT-0045, RDS-EVENT-0046 e RDS-EVENT-0047. Per ulteriori informazioni sugli eventi e sulla sottoscrizione a essi, consulta Utilizzo della notifica degli eventi di Amazon RDS. Se viene restituito un messaggio di errore MariaDB, verifica l'errore nella documentazione dei messaggi di errore MariaDB.

Un problema comune che può provocare errori di replica si verifica quando il valore del parametro `max_allowed_packet` per una replica di lettura è minore del parametro `max_allowed_packet` per l'istanza database di origine. Il parametro `max_allowed_packet` è un parametro personalizzato che puoi impostare in un gruppo dei parametri database usato per specificare la dimensione massima del codice DML che può essere eseguito nel database. In alcuni casi, il valore del parametro `max_allowed_packet` nel gruppo dei parametri database associato a un'istanza database di

origine è minore del valore del parametro `max_allowed_packet` nel gruppo dei parametri database associato alla replica di lettura dell'origine. In questi casi, il processo di replica può generare un errore (indicante che il pacchetto è maggiore dei byte specificati da "max_allowed_packet") e arrestare la replica. Puoi correggere questo errore impostando l'origine e la replica di lettura in modo che utilizzino i gruppi di parametri database con gli stessi valori del parametro `max_allowed_packet`.

Altre situazioni comuni che possono causare errori di replica includono le seguenti:

- Scrittura in tabelle su una replica di lettura. Se crei indici su una replica di lettura, il parametro `read_only` deve essere impostato su 0 affinché gli indici vengano creati. Se scrivi in tabelle nella replica di lettura, l'operazione può interrompere la replica.
- Utilizzo di un motore di storage non transazionale come MyISAM. Le repliche di lettura richiedono un motore di storage transazionale. La replica è supportata solo per il motore di storage InnoDB su MariaDB.
- Utilizzo di query non deterministiche non sicure come `SYSDATE()`. Per ulteriori informazioni, consulta la pagina relativa alla determinazione delle istruzioni sicure e non sicure nel log binario.

Se decidi che un errore possa essere ignorato, completa la procedura descritta in Ignorare l'errore di replica corrente. In caso contrario, puoi eliminare la replica di lettura e creare un'istanza usando lo stesso identificatore istanze DB in modo che l'endpoint sia identico a quello della replica di lettura precedente. Quando un problema relativo alla replica viene risolto, il campo `Replication State` (Stato di replica) cambia in replicating (replica in corso).

Per le istanze database MariaDB, in alcuni casi le repliche di lettura non possono essere passate alle istanze secondarie se alcuni eventi di registro binari (binlog) non vengono scaricati durante l'evento di errore. In questi casi, elimina e ricrea manualmente le repliche di lettura. Puoi ridurre le possibilità che si verifichi una situazione di questo tipo impostando i seguenti valori dei parametri: `sync_binlog=1` e `innodb_flush_log_at_trx_commit=1`. Tali impostazioni potrebbero ridurre le prestazioni, per cui ti consigliamo di testare il loro impatto prima di implementare le modifiche nell'ambiente di produzione.

Configurazione della replica basata su GTID con un'istanza di origine esterna

Puoi impostare la replica basata su identificatori di transazione globali (GTID) da un'istanza MariaDB esterna nella versione 10.0.24 o successiva in un'istanza database RDS for MariaDB. Seguire queste linee guida quando si imposta un'istanza di origine esterna e una replica su Amazon RDS:

- Monitora gli eventi di failover per l'istanza database RDS for MariaDB che rappresenta la tua replica. In caso di failover, l'istanza database che rappresenta la replica potrebbe essere ricreata in un nuovo host con un indirizzo di rete diverso. Per informazioni su come monitorare gli eventi di failover, consulta Utilizzo della notifica degli eventi di Amazon RDS.
- Conservare i log binari (binlog) sull'istanza di origine finché non si ha la conferma che siano stati applicati alla replica. Questa manutenzione assicura il ripristino dell'istanza di origine nel caso di errori.
- Attivare i backup automatici sull'istanza database MariaDB in Amazon RDS. L'attivazione dei backup automatici assicura il ripristino della replica in un determinato "point in time" nel caso fosse necessario risincronizzare l'istanza di origine e la replica. Per informazioni sui backup e sul ripristino point-in-time, consulta Backup, ripristino ed esportazione dei dati.

> **Note**
>
> Le autorizzazioni necessarie per avviare la replica in un'istanza database MariaDB sono limitate e non disponibili per l'utente master Amazon RDS. Per questo motivo, devi usare i comandi Amazon RDS, mysql.rds_set_external_master_gtid e mysql.rds_start_replication per configurare la replica tra il database live e il database di RDS for MariaDB.

Per avviare la replica tra un'istanza di origine esterna e un'istanza database MariaDB in Amazon RDS, attieniti alla procedura seguente.

Per avviare la replica

1. Rendere di sola lettura l'istanza MariaDB di origine:

```
mysql> FLUSH TABLES WITH READ LOCK;
mysql> SET GLOBAL read_only = ON;
```

2. Ottenere il GTID corrente dell'istanza MariaDB esterna. A tale scopo, utilizzare `mysql` o l'editor di query scelto per eseguire `SELECT @@gtid_current_pos;`.

   Il formato del GTID è il seguente: `<domain-id>-<server-id>-<sequence-id>`. Un tipico GTID è simile a **`0-1234510749-1728`**. Per ulteriori informazioni sui GTID e sulle parti da cui è composto, consulta la pagina Global Transaction ID nella documentazione di MariaDB.

3. Copiare il database dall'istanza MariaDB esterna all'istanza database MariaDB tramite `mysqldump`. Per database di dimensioni particolarmente elevate, è possibile utilizzare la procedura in Importazione dei dati in un database Amazon RDS MariaDB o MySQL con tempi di inattività ridotti.

   Per LinuxmacOS, oUnix:

```
mysqldump \
    --databases database_name \
    --single-transaction \
    --compress \
    --order-by-primary \
    -u local_user \
    -plocal_password | mysql \
        --host=hostname \
        --port=3306 \
        -u RDS_user_name \
        -pRDS_password
```

   Per Windows:

```
mysqldump ^
    --databases database_name ^
    --single-transaction ^
    --compress ^
    --order-by-primary \
    -u local_user \
    -plocal_password | mysql ^
        --host=hostname ^
        --port=3306 ^
        -u RDS_user_name ^
        -pRDS_password
```

   > **Note**
   >
   > Assicurarsi che non siano presenti spazi tra l'opzione `-p` e la password immessa. Specifica una password diversa dal prompt mostrato qui come best practice per la sicurezza.

Usare le opzioni `--host`, `--user (-u)`, `--port` e `-p` nel comando `mysql` per specificare il nome host, il nome utente, la porta e la password per la connessione all'istanza database MariaDB. Il nome host è il nome DNS dell'endpoint dell'istanza database MariaDB, ad esempio `myinstance.123456789012.us-east-1.rds.amazonaws.com`. Puoi trovare il valore dell'endpoint è disponibile nei dettagli dell'istanza, nella console di gestione Amazon RDS.

4. Rendi nuovamente scrivibile l'istanza MariaDB di origine.

```
mysql> SET GLOBAL read_only = OFF;
mysql> UNLOCK TABLES;
```

5. Nella Console di gestione di Amazon RDS aggiungi l'indirizzo IP del server che ospita il database MariaDB esterno al gruppo di sicurezza VPC per l'istanza database MariaDB. Per ulteriori informazioni sulla modifica di un gruppo di sicurezza VPC, consulta la sezione relativa ai Gruppi di sicurezza per il VPC nella Guida per l'utente di Amazon Virtual Private Cloud.

   L'indirizzo IP può cambiare quando vengono soddisfatte le seguenti condizioni:

   - Si sta utilizzando un indirizzo IP pubblico per la comunicazione tra l'istanza di origine esterna e l'istanza database.
   - L'istanza di origine esterna è stata arrestata e riavviata.

   Se queste condizioni vengono soddisfatte, verificare l'indirizzo IP prima di aggiungerlo.

   Potrebbe anche essere necessario configurare la rete locale per consentire le connessioni dall'indirizzo IP dell'istanza database MariaDB affinché possa comunicare con l'istanza database MariaDB esterna. Per individuare l'indirizzo IP dell'istanza database MariaDB, usa il comando `host`.

```
host db_instance_endpoint
```

   Il nome host è il nome DNS dell'endpoint dell'istanza database MariaDB.

6. Utilizzando il client scelto, eseguire la connessione all'istanza database MariaDB esterna e creare un utente MariaDB da utilizzare per la replica. Questo account viene utilizzato unicamente per la replica e deve essere limitato al dominio personale per aumentare la sicurezza. Di seguito è riportato un esempio.

```
CREATE USER 'repl_user'@'mydomain.com' IDENTIFIED BY 'password';
```

> **Note**
>
> Specifica una password diversa dal prompt mostrato qui come best practice per la sicurezza.

7. Per un'istanza MariaDB esterna, concedere i privilegi `REPLICATION CLIENT` e `REPLICATION SLAVE` all'utente della replica. Per concedere ad esempio i privilegi `REPLICATION CLIENT` e `REPLICATION SLAVE` su tutti i database per l'utente "`repl_user`" del proprio dominio, eseguire questo comando.

```
GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user'@'mydomain.com';
```

8. Configurare l'istanza database MariaDB come replica. Connettersi all'istanza database MariaDB come utente master e identificare il database MariaDB esterno come istanza origine di replica usando il comando mysql.rds_set_external_master_gtid. Utilizzare il GTID determinato alla fase 2. Di seguito è riportato un esempio.

```
CALL mysql.rds_set_external_master_gtid ('mymasterserver.mydomain.com', 3306,
 'repl_user', 'password', 'GTID', 0);
```

> **Note**
>
> Specifica una password diversa dal prompt mostrato qui come best practice per la sicurezza.

9. Nell'istanza database MariaDB, emettere il comando mysql.rds_start_replication per avviare la replica.

```
CALL mysql.rds_start_replication;
```

Configurazione della replica della posizione del file di log binario con un'istanza di origine esterna.

Puoi impostare la replica fra un'istanza database RDS for MySQL o MariaDB e un'istanza MySQL o MariaDB che è esterna ad Amazon RDS, utilizzando la replica del file di registro binario.

Argomenti

Prima di iniziare

È possibile configurare la replica utilizzando la posizione del file di log binario delle transazioni replicate.

Le autorizzazioni necessarie per avviare la replica in un'istanza database Amazon RDS sono limitate e non disponibili per l'utente master Amazon RDS. Per questo motivo, assicurati di usare i comandi mysql.rds_set_external_master e mysql.rds_start_replication in Amazon RDS per configurare la replica tra il database live e il database Amazon RDS.

Per impostare il formato di logging binario per un database MySQL o MariaDB, aggiornare il parametro `binlog_format`. Se l'istanza database utilizza il gruppo di parametri di istanza database predefinito, crea un nuovo gruppo di parametri di istanza database per modificare le impostazioni `binlog_format`. Ti consigliamo di mantenere le impostazioni predefinite per `binlog_format`, che è `MIXED`. Tuttavia, puoi anche impostare `binlog_format` su `ROW` o `STATEMENT` se hai bisogno di un formato di registro binario (binlog) specifico. Riavvia l'istanza database affinché venga applicata la modifica.

Per ulteriori informazioni sull'impostazione del parametro `binlog_format`, consulta Configurazione del log binario di MySQL Per ulteriori informazioni sulle implicazioni dei vari tipi di replica MySQL, consulta la pagina relativa a vantaggi e svantaggi della replica basata su istruzioni e basata su riga nella documentazione di MySQL.

> **Note**
>
> A partire dalla versione 8.0.36 di RDS per MySQL, Amazon RDS non replica il database. `mysql` Pertanto, se ci sono utenti nel database esterno di cui hai bisogno nella replica di Amazon RDS, assicurati di crearli manualmente.

Configurazione della replica della posizione del file di log binario con un'istanza di origine esterna.

Seguire queste linee guida quando si imposta un'istanza di origine esterna e una replica su Amazon RDS:

- Monitorare gli eventi di failover per l'istanza database di Amazon RDS che rappresenta la replica. In caso di failover, l'istanza database che rappresenta la replica potrebbe essere ricreata in un nuovo host con un indirizzo di rete diverso. Per informazioni su come monitorare gli eventi di failover, consulta Utilizzo della notifica degli eventi di Amazon RDS.
- Conservare i binlog sull'istanza di origine finché non si ha la conferma che siano stati applicati alla replica. Conservando questi file, si è certi di poter ripristinare l'istanza di origine in caso di errori.
- Attivare i backup automatici sull'istanza database di Amazon RDS. L'attivazione dei backup automatici assicura il ripristino della replica a un punto temporale specifico nel caso fosse necessario risincronizzare l'istanza di origine e la replica. Per informazioni su backup e point-in-time ripristino, consulta. Backup, ripristino ed esportazione dei dati

Per configurare la replica della posizione del file di log binario con un'istanza di origine esterna

1. Rendere l'istanza MySQL o MariaDB di origine di sola lettura.

   ```
   mysql> FLUSH TABLES WITH READ LOCK;
   mysql> SET GLOBAL read_only = ON;
   ```

2. Eseguire il comando `SHOW MASTER STATUS` nell'istanza database di MySQL o MariaDB di origine per determinare la posizione del binlog.

   Viene restituito un output simile all'esempio seguente.

   ```
   File                          Position
   ----------------------------------
   ```

```
 mysql-bin-changelog.000031      107
-----------------------------------
```

3. Copiare il database dall'istanza esterna all'istanza database Amazon RDS usando `mysqldump`. Per database di dimensioni particolarmente elevate, è possibile utilizzare la procedura in Importazione dei dati in un database Amazon RDS MariaDB o MySQL con tempi di inattività ridotti.

   Per LinuxmacOS, oUnix:

   ```
   mysqldump --databases database_name \
       --single-transaction \
       --compress \
       --order-by-primary \
       -u local_user \
       -plocal_password | mysql \
           --host=hostname \
           --port=3306 \
           -u RDS_user_name \
           -pRDS_password
   ```

   Per Windows:

   ```
   mysqldump --databases database_name ^
       --single-transaction ^
       --compress ^
       --order-by-primary ^
       -u local_user ^
       -plocal_password | mysql ^
           --host=hostname ^
           --port=3306 ^
           -u RDS_user_name ^
           -pRDS_password
   ```

   > **Note**
   >
   > Assicurarsi che non siano presenti spazi tra l'opzione `-p` e la password immessa.

   Utilizzare le opzioni `--host`, `--user (-u)`, `--port` e `-p` nel comando `mysql` per specificare il nome host, il nome utente, la porta e la password per la connessione all'istanza

database Amazon RDS. Il nome host è il nome DNS (Domain Name Service) dell'endpoint dell'istanza database di Amazon RDS, ad esempio `myinstance.123456789012.us-east-1.rds.amazonaws.com`. È possibile trovare il valore dell'endpoint nei dettagli dell'istanza nella AWS Management Console.

4. Rendere nuovamente scrivibile l'istanza MySQL o MariaDB di origine.

```
mysql> SET GLOBAL read_only = OFF;
mysql> UNLOCK TABLES;
```

Per ulteriori informazioni sulla creazione di backup da utilizzare con la replica, vedere la documentazione di MySQL.

5. Nel AWS Management Console, aggiungi l'indirizzo IP del server che ospita il database esterno al gruppo di sicurezza del cloud privato virtuale (VPC) per l'istanza database Amazon RDS. Per ulteriori informazioni sulla modifica di un gruppo di sicurezza VPC, consulta Gruppi di sicurezza per il VPC nella Guida per l'utente di Amazon Virtual Private Cloud.

L'indirizzo IP può cambiare quando vengono soddisfatte le seguenti condizioni:

- Si sta utilizzando un indirizzo IP pubblico per la comunicazione tra l'istanza di origine esterna e l'istanza database.
- L'istanza di origine esterna è stata arrestata e riavviata.

Se queste condizioni vengono soddisfatte, verificare l'indirizzo IP prima di aggiungerlo.

Potrebbe anche essere necessario configurare la rete locale per consentire le connessioni dall'indirizzo IP dell'istanza database di Amazon RDS, affinché possa comunicare con l'istanza MySQL o MariaDB esterna. Per individuare l'indirizzo IP dell'istanza database di Amazon RDS, utilizzare il comando `host`.

```
host db_instance_endpoint
```

Il nome host è il nome DNS dall'endpoint dell'istanza database di Amazon RDS.

6. Utilizzando il client scelto, eseguire la connessione all'istanza esterna e creare un utente da utilizzare per la replica. Utilizza questo account unicamente per la replica e limitalo al dominio personale per aumentare la sicurezza. Di seguito è riportato un esempio.

```
CREATE USER 'repl_user'@'mydomain.com' IDENTIFIED BY 'password';
```

> **Note**
>
> Specifica una password diversa dal prompt mostrato qui come best practice per la sicurezza.

7. Per l'istanza esterna, concedere i privilegi `REPLICATION CLIENT` e `REPLICATION SLAVE` all'utente della replica. Per concedere ad esempio i privilegi `REPLICATION CLIENT` e `REPLICATION SLAVE` su tutti i database per l'utente "`repl_user`" del proprio dominio, eseguire questo comando.

```
GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user'@'mydomain.com';
```

8. Definire l'istanza database di Amazon RDS come replica. A tale scopo, connettersi innanzitutto all'istanza database di Amazon RDS come l'utente master. Quindi, identificare il database MySQL o MariaDB esterno come istanza di origine utilizzando il comando mysql.rds_set_external_master. Utilizzare il nome e la posizione del file di log master recuperati nella fase 2. Di seguito è riportato un esempio.

```
CALL mysql.rds_set_external_master ('mymasterserver.mydomain.com', 3306,
 'repl_user', 'password', 'mysql-bin-changelog.000031', 107, 0);
```

> **Note**
>
> In RDS for MySQL puoi scegliere di usare la replica ritardata eseguendo invece la procedura archiviata mysql.rds_set_external_master_with_delay. Su RDS for MySQL, una ragione per utilizzare la replica ritardata è attuare il ripristino di emergenza con la procedura archiviata mysql.rds_start_replication_until. Attualmente RDS for MariaDB supporta la replica ritardata ma non supporta la procedura `mysql.rds_start_replication_until`.

9. Nell'istanza database di Amazon RDS eseguire il comando mysql.rds_start_replication per avviare la replica.

```
CALL mysql.rds_start_replication;
```

Opzioni per il motore di database MariaDB

Vengono fornite descrizioni delle opzioni o delle funzionalità aggiuntive disponibili per le istanze Amazon RDS che eseguono il motore di database MariaDB. Per attivare queste opzioni, devi aggiungerle a un gruppo di opzioni personalizzate e quindi associare il gruppo di opzioni all'istanza database. Per ulteriori informazioni sull'utilizzo di gruppi di opzioni, consulta Uso di gruppi di opzioni.

Amazon RDS supporta le seguenti opzioni per MariaDB:

ID opzione	Versioni del motore
`MARIADB_AUDIT_PLUGIN`	MariaDB 10.3 e versioni successive

Supporto del plug-in per audit MariaDB

Amazon RDS supporta l'uso del plug-in per audit MariaDB sulle istanze database MariaDB. Il plug-in per audit MariaDB registra le attività del database, ad esempio gli utenti che accedono al database, le query eseguite sul database e altro ancora. Il record con le attività del database è archiviato in un file di log.

Impostazioni dell'opzione relativa al plug-in per audit

Amazon RDS supporta le seguenti impostazioni per l'opzione relativa al plug-in per audit MariaDB.

> **Note**
>
> Se non configuri un'impostazione di opzione nella console, RDS utilizza l'impostazione predefinita.

Impostazione opzioni	Valori validi	Valore predefinito	Descrizione
`SERVER_AUDIT_FILE_PATH`	`/rdsdbdata/log/audit/`	`/rdsdbdata/log/audit/`	La posizione del file di log. Il file di log contiene il record dell'attività specificata in `SERVER_AUDIT_EVENTS` . Per ulteriori informazioni, consulta Visualizzazione ed elenco dei file di

Impostazione opzioni	Valori validi	Valore predefinito	Descrizione
			log del database e File di log del database MariaDB.
SERVER_AUDIT_FILE_ROTATE_SIZE	1–1000000000	1000000	La dimensione in byte che, una volta raggiunta , provoca la rotazione del file. Per ulteriori informazioni, consulta Dimensione del file di registro.
SERVER_AUDIT_FILE_ROTATIONS	0–100	9	Il numero di rotazioni dei log da salvare se `server_audit_output_type=file` . Se impostata su 0, la rotazione del file di log non viene mai eseguita. Per ulteriori informazioni, consulta Dimensione del file di registro e Download di un file di log di database.

Impostazione opzioni	Valori validi	Valore predefinito	Descrizione
SERVER_AUDIT_EVENTS	CONNECT, QUERY, TABLE, QUERY_DDL , QUERY_DML , QUERY_DML_NO_SELECT , QUERY_DCL	CONNECT, QUERY	I tipi di attività da registrate nel log. Viene registrata anche l'installazione del plug-in per audit MariaDB. • CONNECT: registrazione delle connessioni al database con esito positivo e negativo e delle disconnessioni dal database. • QUERY: registrazione del testo di tutte le query eseguite sul database. • TABLE: registrazione delle tabelle su cui hanno impatto le query eseguite sul database. • QUERY_DDL : simile all'evento QUERY, ma restituisce solo le query DDL (Data Definition Language), (CREATE, ALTER e così via). • QUERY_DML : simile all'evento QUERY, ma restituisce solo le query DML (Data Manipulation Language), (INSERT, UPDATE, e così via e anche SELECT). • QUERY_DML_NO_SELECT : simile all'evento QUERY_DML ma non registra le query SELECT. • QUERY_DCL : simile all'evento QUERY, ma restituisce solo le query DCL (Data Control Language), (GRANT, REVOKE e così via).

Impostazione opzioni	Valori validi	Valore predefinito	Descrizione
SERVER_AUDIT_INCL_USERS	Più valori separati da virgola	Nessuna	Sono incluse solo le attività degli utenti specificati. Per impostazione predefinita, l'attività viene registrata per tutti gli utenti. SERVER_AUDIT_INCL_USERS e SERVER_AUDIT_EXCL_USERS si escludono a vicenda. Se si aggiungono valori a SERVER_AUDIT_INCL_USERS, è necessario assicurarsi che non venga aggiunto alcun valore a SERVER_AUDIT_EXCL_USERS.

Impostazione opzioni	Valori validi	Valore predefinito	Descrizione
SERVER_AUDIT_EXCL_USERS	Più valori separati da virgola	Nessuna	Sono escluse le attività degli utenti specifica ti. Per impostazione predefinita, l'attività viene registrata per tutti gli utenti. SERVER_AU DIT_INCL_USERS e SERVER_AU DIT_EXCL_USERS si escludono a vicenda. Se si aggiungono valori a SERVER_AU DIT_EXCL_USERS , è necessario assicurar si che non venga aggiunto alcun valore a SERVER_AUDIT_INCL_USERS . L'utente rdsadmin esegue query sul database ogni secondo per verificare l'integrità del database. In base alle altre impostazioni, questa attività può causare un rapido ed eccessivo aumento delle dimensioni del file di log. Se non desideri registrare questa attività, aggiungi l'utente rdsadmin all'elenco SERVER_AUDIT_EXCL_USERS . **Note** CONNECTL'attività viene sempre registrata per tutti gli utenti, anche se l'utente è specificato per l'impostazione di questa opzione.
SERVER_AUDIT_LOGGING	ON	ON	La registrazione è attiva. L'unico valore valido è ON. Amazon RDS non supporta la disattiva zione del logging. Se desideri disattivare la registrazione, rimuovi il plug-in per audit MariaDB. Per ulteriori informazioni, consulta Rimozione del plug-in per audit MariaDB.

Impostazione opzioni	Valori validi	Valore predefinito	Descrizione
SERVER_AUDIT_QUERY_LOG_LIMIT	0–2147483647	1.024	Il limite di lunghezza della stringa di query in un record.

Aggiunta del plug-in per audit MariaDB

Di seguito è riportato il processo generale per aggiungere il plug-in per audit MariaDB a un'istanza database:

1. Creare un nuovo gruppo di opzioni oppure copiare o modificare un gruppo di opzioni esistente.
2. Aggiungere l'opzione al gruppo di opzioni.
3. Associare il gruppo di opzioni a questa istanza database.

Dopo aver aggiunto il plug-in per audit MariaDB, non dovrai riavviare la tua istanza database. Non appena il gruppo di opzioni è attivo, inizia immediatamente l'audit.

Per aggiungere il plug-in per audit MariaDB

1. Determinare il gruppo di opzioni che si desidera usare. È possibile creare un nuovo gruppo di opzioni oppure usare un gruppo di opzioni esistente. Se si desidera usare un gruppo di opzioni esistente, puoi passare alla fase successiva. Altrimenti, creare un gruppo di opzioni database personalizzato. Scegli mariadb per Engine (Motore), quindi 10.3 o versione successiva per Major engine version (Versione del motore principale). Per ulteriori informazioni, consulta Creazione di un gruppo di opzioni.
2. Aggiungere l'opzione MARIADB_AUDIT_PLUGIN al gruppo di opzioni e configurare le impostazioni dell'opzione. Per ulteriori informazioni sull'aggiunta di opzioni, consulta Aggiunta di un'opzione a un gruppo di opzioni. Per ulteriori informazioni su ciascuna impostazione, consulta Impostazioni dell'opzione relativa al plug-in per audit.
3. Applicare il gruppo di opzioni a un'istanza database nuova o esistente.
   - Per una nuova istanza database, si applica il gruppo di opzioni quando viene avviata l'istanza. Per ulteriori informazioni, consulta Creazione di un'istanza database Amazon RDS.

- Per un'istanza database esistente, viene applicato il gruppo di opzioni modificando l'istanza database e collegando il nuovo gruppo di opzioni. Per ulteriori informazioni, consulta Modifica di un'istanza database Amazon RDS.

Visualizzazione e download del log del plug-in per audit MariaDB

Dopo avere abilitato il plug-in per audit MariaDB, potrai accedere ai risultati nei file di log nello stesso modo in cui accedi a qualsiasi altro file di log basato su testo. I file di log per audit si trovano in `/rdsdbdata/log/audit/`. Per ulteriori informazioni sulla visualizzazione del file di log nella console, consulta Visualizzazione ed elenco dei file di log del database. Per informazioni sul download del file di log, consulta Download di un file di log di database.

Modifica delle impostazioni del plug-in per audit MariaDB

Dopo aver abilitato il plug-in per audit MariaDB, puoi modificare le relative impostazioni. Per ulteriori informazioni su come modificare le impostazioni dell'opzione, consulta Modifica di un'impostazione di un'opzione. Per ulteriori informazioni su ciascuna impostazione, consulta Impostazioni dell'opzione relativa al plug-in per audit.

Rimozione del plug-in per audit MariaDB

Amazon RDS non supporta la disattivazione della registrazione nel plug-in per audit MariaDB. Puoi tuttavia rimuovere il plug-in da un'istanza database. Dopo aver rimosso il plug-in per audit MariaDB, l'istanza database viene riavviata automaticamente per arrestare l'audit.

Per rimuovere il plug-in per audit MariaDB da un'istanza database, procedi in uno dei seguenti modi:

- Rimuovere l'opzione relativa al plug-in per audit MariaDB dal gruppo di opzioni a cui appartiene. Questa modifica coinvolge tutte le istanze database che usano il gruppo di opzioni. Per ulteriori informazioni, consulta Rimozione di un'opzione da un gruppo di opzioni
- Modificare l'istanza database e specificare un diverso gruppo di opzioni che non comprenda il plug-in. Questa modifica coinvolge una singola istanza database. È possibile specificare il gruppo di opzioni predefinito (vuoto) o un gruppo di opzioni personalizzato diverso. Per ulteriori informazioni, consulta Modifica di un'istanza database Amazon RDS.

Parametri per MariaDB

Per impostazione predefinita, un'istanza database MariaDB utilizza un gruppo di parametri database specifico a un database MariaDB. Questo gruppo di parametri contiene alcuni ma non tutti i parametri inclusi nel gruppo di parametri database Amazon RDS per il motore di database MySQL. Contiene inoltre vari parametri nuovi specifici di MariaDB. Per informazioni sull'utilizzo dei gruppi di parametri e sull'impostazione dei parametri, consulta Utilizzo di gruppi di parametri.

Visualizzazione dei parametri MariaDB

I parametri di RDS for MariaDB sono impostati sui valori predefiniti del motore di storage selezionati. Per ulteriori informazioni sui parametri di MariaDB, consulta la documentazione di MariaDB. Per ulteriori informazioni sui motori di storage MariaDB, consulta Motori di storage supportati per MariaDB in Amazon RDS.

È possibile visualizzare i parametri disponibili per una versione RDS for MariaDB specifica utilizzando la console RDS o la AWS CLI. Per informazioni sulla visualizzazione dei parametri in un gruppo di parametri MariaDB nella console RDS, consulta Visualizzazione dei valori dei parametri per un gruppo di parametri del database.

Utilizzando AWS CLI, è possibile visualizzare i parametri di una versione RDS for MariaDB eseguendo il comando `describe-engine-default-parameters`. Indica uno dei valori seguenti per l'opzione `--db-parameter-group-family`:

- `mariadb10.11`
- `mariadb10.6`
- `mariadb10.5`
- `mariadb10.4`
- `mariadb10.3`

Ad esempio, per visualizzare i parametri supportati per RDS for MariaDB versione 10.6 esegui il comando seguente.

```
aws rds describe-engine-default-parameters --db-parameter-group-family mariadb10.6
```

L'output avrà un aspetto simile al seguente.

```
{
```

```
    "EngineDefaults": {
        "Parameters": [
            {
                "ParameterName": "alter_algorithm",
                "Description": "Specify the alter table algorithm.",
                "Source": "engine-default",
                "ApplyType": "dynamic",
                "DataType": "string",
                "AllowedValues": "DEFAULT,COPY,INPLACE,NOCOPY,INSTANT",
                "IsModifiable": true
            },
            {
                "ParameterName": "analyze_sample_percentage",
                "Description": "Percentage of rows from the table ANALYZE TABLE will
 sample to collect table statistics.",
                "Source": "engine-default",
                "ApplyType": "dynamic",
                "DataType": "float",
                "AllowedValues": "0-100",
                "IsModifiable": true
            },
            {
                "ParameterName": "aria_block_size",
                "Description": "Block size to be used for Aria index pages.",
                "Source": "engine-default",
                "ApplyType": "static",
                "DataType": "integer",
                "AllowedValues": "1024-32768",
                "IsModifiable": false
            },
            {
                "ParameterName": "aria_checkpoint_interval",
                "Description": "Interval in seconds between automatic checkpoints.",
                "Source": "engine-default",
                "ApplyType": "dynamic",
                "DataType": "integer",
                "AllowedValues": "0-4294967295",
                "IsModifiable": true
            },
        ...
```

Per visualizzare i parametri supportati per RDS for MariaDB versione 10.6 esegui il comando seguente.

Per LinuxmacOS, oUnix:

```
aws rds describe-engine-default-parameters --db-parameter-group-family mariadb10.6 \
   --query 'EngineDefaults.Parameters[?IsModifiable==`true`]'
```

Per Windows:

```
aws rds describe-engine-default-parameters --db-parameter-group-family mariadb10.6 ^
   --query "EngineDefaults.Parameters[?IsModifiable==`true`]"
```

Parametri MySQL non disponibili

I seguenti parametri MySQL non sono disponibili nei gruppi di parametri database specifici di MariaDB:

- bind_address
- binlog_error_action
- binlog_gtid_simple_recovery
- binlog_max_flush_queue_time
- binlog_order_commits
- binlog_row_image
- binlog_rows_query_log_events
- binlogging_impossible_mode
- block_encryption_mode
- core_file
- default_tmp_storage_engine
- div_precision_increment
- end_markers_in_json
- enforce_gtid_consistency
- eq_range_index_dive_limit
- explicit_defaults_for_timestamp
- gtid_executed
- gtid-mode
- gtid_next

- gtid_owned
- gtid_purged
- log_bin_basename
- log_bin_index
- log_bin_use_v1_row_events
- log_slow_admin_statements
- log_slow_slave_statements
- log_throttle_queries_not_using_indexes
- master-info-repository
- optimizer_trace
- optimizer_trace_features
- optimizer_trace_limit
- optimizer_trace_max_mem_size
- optimizer_trace_offset
- relay_log_info_repository
- rpl_stop_slave_timeout
- slave_parallel_workers
- slave_pending_jobs_size_max
- slave_rows_search_algorithms
- storage_engine
- table_open_cache_instances
- timed_mutexes
- transaction_allow_batching
- validate-password
- validate_password_dictionary_file
- validate_password_length
- validate_password_mixed_case_count
- validate_password_number_count
- validate_password_policy
- validate_password_special_char_count

Per ulteriori informazioni sui parametri di MySQL, consulta la documentazione di MySQL.

Migrazione dei dati da uno snapshot DB MySQL a un'istanza database MariaDB

Puoi eseguire la migrazione di uno snapshot DB RDS for MySQL a una nuova istanza database che esegue MariaDB utilizzando AWS Management Console, AWS CLI, o l'API Amazon RDS. Devi utilizzare lo snapshot di database creato da un'istanza database Amazon RDS che esegue MySQL 5.6. o 5.7. Per informazioni su come creare uno snapshot database RDS for MySQL, consulta Creazione di uno snapshot DB per un'istanza DB Single-AZ.

La migrazione dello snapshot non influisce sull'istanza database originale da cui è stato acquisito lo snapshot. È possibile testare e convalidare la nuova istanza database prima di indirizzarvi il traffico in sostituzione dell'istanza database originale.

Dopo aver effettuato la migrazione da MySQL a MariaDB, l'istanza database MariaDB sarà associata al gruppo di parametri database e al gruppo di opzioni predefiniti. Dopo aver ripristinato lo snapshot DB, è possibile associare un gruppo di parametri database personalizzato alla nuova istanza database. Tuttavia, un gruppo di parametri MariaDB ha un set diverso di variabili di sistema configurabili. Per informazioni sulle differenze tra le variabili di sistema MySQL e MariaDB, consultare System Variable Differences Between MariaDB and MySQL. Per informazioni sui gruppi di parametri database, consulta Utilizzo di gruppi di parametri. Per informazioni sui gruppi di opzioni, consulta Uso di gruppi di opzioni.

Esecuzione della migrazione

È possibile eseguire la migrazione di uno snapshot DB RDS for MySQL in una nuova istanza database di MariaDB utilizzando la AWS Management Console, AWS CLI, o l'API RDS.

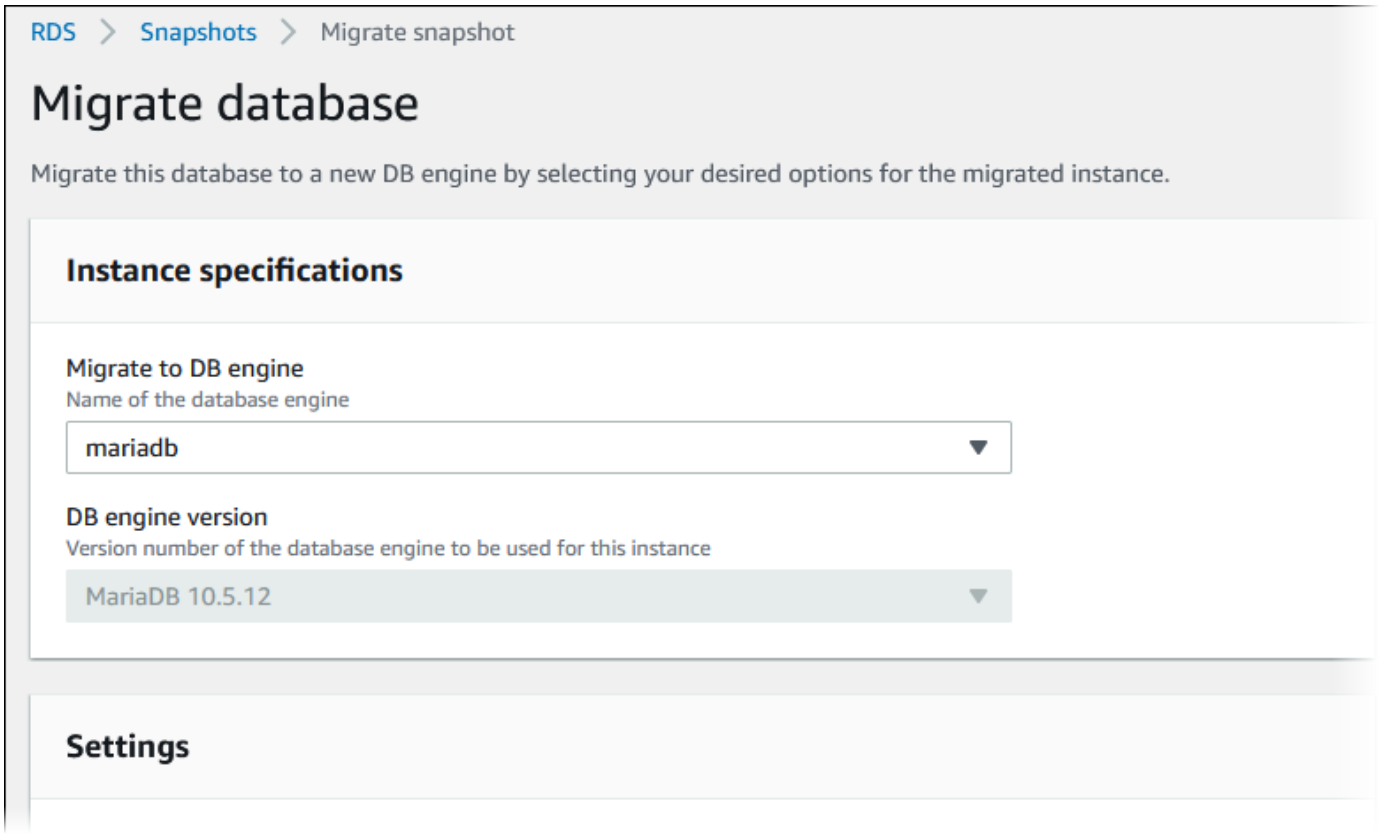
Console

Per effettuare la migrazione di una snapshot DB MySQL a un'istanza database MariaDB

1. Accedi alla AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, scegliere Snapshots (Snapshot), quindi selezionare la snapshot DB MySQL di cui si desidera effettuare la migrazione.
3. Per Actions (Operazioni) scegliere Migrate Snapshot (Migrazione dello snapshot). Viene visualizzata la pagina Migrate Database (Migrazione database).

4. Per Migrate to DB Engine (Migra al motore del database), scegliere mariadb.

   Amazon RDS seleziona automaticamente la versione motore di database. Non è possibile modificare la versione del motore di database.



5. Per le restanti sezioni, specifica le impostazioni dell'istanza database. Per informazioni su ciascuna impostazione, consulta Impostazioni per istanze database.
6. Scegliere Migrate (Migrazione).

AWS CLI

Per eseguire la migrazione dei dati da uno snapshot DB MySQL a un'istanza database MariaDB, utilizzare il comando AWS CLI `restore-db-instance-from-db-snapshot` con i parametri seguenti:

- -- db-instance-identifier — Nome dell'istanza DB da creare dallo snapshot del DB.
- -- db-snapshot-identifier — L'identificatore per lo snapshot del DB da cui eseguire il ripristino.
- --engine – Il motore di database da utilizzare per la nuova istanza.

Example

PerLinux, omacOS: Unix

```
aws rds restore-db-instance-from-db-snapshot \
    --db-instance-identifier newmariadbinstance \
    --db-snapshot-identifier mysqlsnapshot \
    --engine mariadb
```

Per Windows:

```
aws rds restore-db-instance-from-db-snapshot ^
    --db-instance-identifier newmariadbinstance ^
    --db-snapshot-identifier mysqlsnapshot ^
    --engine mariadb
```

API

Per eseguire la migrazione dei dati da uno snapshot database MySQL a un'istanza database MariaDB, chiamare l'operazione `RestoreDBInstanceFromDBSnapshot` dell'API Amazon RDS.

Incompatibilità tra MariaDB e MySQL

Le incompatibilità tra MySQL e MariaDB includono quanto segue:

- Non puoi eseguire la migrazione di una snapshot DB creata con MySQL 8.0 a MariaDB.
- Se il database MySQL di origine utilizza un hash della password SHA256, è necessario reimpostare le password utente con hash SHA256 prima di effettuare la connessione al database MariaDB. Il seguente codice mostra come reimpostare una password con hash SHA256.

  ```
  SET old_passwords = 0;
  UPDATE mysql.user SET plugin = 'mysql_native_password',
  Password = PASSWORD('new_password')
  WHERE (User, Host) = ('master_user_name', %);
  FLUSH PRIVILEGES;
  ```

- Se l'account utente master RDS utilizza l'hash della password SHA-256, è necessario reimpostare la password utilizzando la AWS Management Console, il comando `modify-db-instance`

AWS CLI o l'operazione API RDS ModifyDBInstance. Per ulteriori informazioni sulla modifica di un'istanza database, consulta Modifica di un'istanza database Amazon RDS.

- MariaDB non supporta il plugin Memcached. Tuttavia, i dati utilizzati dal plugin Memcached sono archiviati come tabelle InnoDB. Dopo aver effettuato la migrazione di una snapshot DB MySQL, è possibile accedere ai dati utilizzati dal plugin Memcached mediante SQL. Per ulteriori informazioni sul database innodb_memcache, consulta la pagina relativa a plugin interni InnoDB memcached.

MariaDB sul riferimento SQL di Amazon RDS

Di seguito, sono disponibili descrizioni di stored procedure di sistema disponibili alle istanze Amazon RDS che eseguono il motore di database MariaDB.

Puoi utilizzare tutte le stored procedure del sistema disponibili per le istanze database MySQL e MariaDB. Le procedure archiviate sono documentate in Riferimento delle stored procedure RDS per MySQL. Le istanze database MariaDB supportano tutte le procedure archiviate, ad eccezione di `mysql.rds_start_replication_until` e `mysql.rds_start_replication_until_gtid`.

Inoltre, le seguenti procedure archiviate nel sistema sono supportate solo per le istanze database Amazon RDS che eseguono MariaDB:

- mysql.rds_replica_status
- mysql.rds_set_external_master_gtid
- mysql.rds_kill_query_id

mysql.rds_replica_status

Mostra lo stato di replica di una replica di lettura MariaDB.

Richiamare questa procedura sulla replica di lettura per visualizzare le informazioni sullo stato dei parametri essenziali dei thread di replica.

Sintassi

```
CALL mysql.rds_replica_status;
```

Note per l'utilizzo

Questa procedura è supportata solo per le istanze database MariaDB che eseguono MariaDB versione 10.5 e successive.

Questa procedura è l'equivalente del comando `SHOW REPLICA STATUS`. Questo comando non è supportato per le istanze database MariaDB versione 10.5 e successive.

Nelle versioni precedenti di MariaDB, il comando `SHOW SLAVE STATUS` equivalente richiedeva il privilegio `REPLICATION SLAVE`. In MariaDB 10.5 e versioni successive, richiede il privilegio

`REPLICATION REPLICA ADMIN`. Per proteggere la gestione RDS delle istanze database MariaDB 10.5 e versioni successive, questo nuovo privilegio non viene concesso all'utente master RDS.

Examples (Esempi)

L'esempio seguente mostra lo stato di una replica di lettura MariaDB:

```
call mysql.rds_replica_status;
```

La risposta è simile a quella riportata di seguito.

```
*************************** 1. row ***************************
               Replica_IO_State: Waiting for master to send event
                    Source_Host: XX.XX.XX.XXX
                    Source_User: rdsrepladmin
                    Source_Port: 3306
                  Connect_Retry: 60
                Source_Log_File: mysql-bin-changelog.003988
            Read_Source_Log_Pos: 405
                 Relay_Log_File: relaylog.011024
                  Relay_Log_Pos: 657
          Relay_Source_Log_File: mysql-bin-changelog.003988
             Replica_IO_Running: Yes
            Replica_SQL_Running: Yes
                Replicate_Do_DB:
            Replicate_Ignore_DB:
             Replicate_Do_Table:
         Replicate_Ignore_Table:
mysql.rds_sysinfo,mysql.rds_history,mysql.rds_replication_status
        Replicate_Wild_Do_Table:
    Replicate_Wild_Ignore_Table:
                     Last_Errno: 0
                     Last_Error:
                   Skip_Counter: 0
            Exec_Source_Log_Pos: 405
                Relay_Log_Space: 1016
                Until_Condition: None
                 Until_Log_File:
                  Until_Log_Pos: 0
             Source_SSL_Allowed: No
             Source_SSL_CA_File:
             Source_SSL_CA_Path:
                Source_SSL_Cert:
```

```
               Source_SSL_Cipher:
                  Source_SSL_Key:
           Seconds_Behind_Master: 0
   Source_SSL_Verify_Server_Cert: No
                   Last_IO_Errno: 0
                   Last_IO_Error:
                  Last_SQL_Errno: 0
                  Last_SQL_Error:
     Replicate_Ignore_Server_Ids:
                Source_Server_Id: 807509301
                  Source_SSL_Crl:
              Source_SSL_Crlpath:
                      Using_Gtid: Slave_Pos
                     Gtid_IO_Pos: 0-807509301-3980
         Replicate_Do_Domain_Ids:
     Replicate_Ignore_Domain_Ids:
                   Parallel_Mode: optimistic
                       SQL_Delay: 0
             SQL_Remaining_Delay: NULL
       Replica_SQL_Running_State: Reading event from the relay log
              Replica_DDL_Groups: 15
Replica_Non_Transactional_Groups: 0
    Replica_Transactional_Groups: 3658
1 row in set (0.000 sec)

Query OK, 0 rows affected (0.000 sec)
```

mysql.rds_set_external_master_gtid

Configura la replica basata su GTID da un'istanza MariaDB in esecuzione all'esterno di Amazon RDS a un'istanza database MariaDB. Questa procedura archiviata è supportata solo se l'istanza esterna di MariaDB è la versione 10.0.24 o successiva. Quando configuri una replica dove una o entrambe le istanze non supportano gli ID globale di transazione (GTID) di MariaDB, utilizza mysql.rds_set_external_master.

L'utilizzo di GTID per la replica offre funzioni di sicurezza contro l'arresto anomalo non offerte dalla replica dei log binari. Pertanto è consigliata per la replica di istanze che li supportano.

Sintassi

```
CALL mysql.rds_set_external_master_gtid(
```

```
  host_name
  , host_port
  , replication_user_name
  , replication_user_password
  , gtid
  , ssl_encryption
);
```

Parametri

host_name

Stringa. Il nome host o l'indirizzo IP dell'istanza MariaDB in esecuzione all'esterno di Amazon RDS che diventerà l’istanza di origine.

host_port

Numero intero. La porta utilizzata dall'istanza MariaDB in esecuzione all'esterno di Amazon RDS da configurare come istanza di origine. Se la configurazione della rete include la replica della porta SSH che converte il numero di porta, specifica il numero di porta esposto da SSH.

replication_user_name

Stringa. L'ID di un utente con autorizzazioni `REPLICATION SLAVE` nell'istanza database MariaDB da configurare come replica di lettura.

replication_user_password

Stringa. La password dell'ID utente specificata in `replication_user_name`.

gtid

Stringa. L'ID globale di transazione sull’istanza di origine dalla quale la replica dovrebbe iniziare.

Puoi utilizzare `@@gtid_current_pos` per ottenere il GTID corrente se l’istanza di origine è stata bloccata durante la configurazione della replica, così il log binario non cambia tra i punti quando ottieni il GTID e quando la replica inizia.

In alternativa, se utilizzi `mysqldump` versione 10.0.13 o successiva per compilare l'istanza di replica prima di avviare la replica, puoi ottenere la posizione GTID nell'output utilizzando l'opzione `--master-data` o `--dump-slave`. Se non utilizzi `mysqldump` versione 10.0.13 o successiva, puoi eseguire `SHOW MASTER STATUS` o utilizzare quelle stesse opzioni `mysqldump` per ottenere il nome e la posizione del file di log binario, quindi convertirle in un GTID eseguendo `BINLOG_GTID_POS` sull'istanza MariaDB esterna:

```
SELECT BINLOG_GTID_POS('<binary log file name>', <binary log file position>);
```

Per ulteriori informazioni sull'implementazione di GTID di MariaDB, consulta ID globale di transazione (GTID) nella documentazione di MariaDB.

ssl_encryption

Un valore che specifica se la crittografia Secure Socket Layer (SSL) è utilizzata sulla connessione di replica. 1 indica che la crittografia SSL deve essere utilizzata; 0 specifica che la crittografia non deve essere utilizzata. Il valore predefinito è 0.

> **Note**
>
> L'opzione `MASTER_SSL_VERIFY_SERVER_CERT` non è supportata. Questa opzione è impostata su 0, il che significa che la connessione è crittografata, ma i certificati non sono verificati.

Note per l'utilizzo

La procedura `mysql.rds_set_external_master_gtid` deve essere eseguita dall'utente master. Deve essere eseguita sull'istanza database MariaDB che stai configurando come replica di un'istanza MariaDB in esecuzione all'esterno di Amazon RDS. Prima di eseguire `mysql.rds_set_external_master_gtid`, devi aver configurato l'istanza di MariaDB in esecuzione all'esterno di Amazon RDS come istanza di origine. Per ulteriori informazioni, consulta Importazione di dati in un'istanza database MariaDB.

> **Warning**
>
> Non usare `mysql.rds_set_external_master_gtid` per gestire la replica tra due istanze database Amazon RDS. Utilizzala soltanto nel caso della replica con un'istanza di MariaDB in esecuzione all'esterno di RDS. Per ulteriori informazioni sulla gestione della replica tra istanze database Amazon RDS, consulta Uso delle repliche di lettura dell'istanza database.

Dopo aver chiamato `mysql.rds_set_external_master_gtid` per configurare un'istanza database di Amazon RDS come una replica di lettura, puoi chiamare mysql.rds_start_replication nella replica per avviare il processo di replica. Puoi chiamare mysql.rds_reset_external_master per rimuovere la configurazione della replica di lettura.

Quando `mysql.rds_set_external_master_gtid` viene chiamato, Amazon RDS registra l'ora, l'utente e un'operazione di "impostazione master" nelle tabelle `mysql.rds_history` e `mysql.rds_replication_status`.

Examples (Esempi)

Nel caso di esecuzione di un'istanza database MariaDB, l'esempio seguente la configura come replica di un'istanza di MariaDB in esecuzione all'esterno di Amazon RDS.

```
call mysql.rds_set_external_master_gtid
 ('Sourcedb.some.com',3306,'ReplicationUser','SomePassW0rd','0-123-456',0);
```

mysql.rds_kill_query_id

Termina una query in esecuzione sul server MariaDB.

Sintassi

```
CALL mysql.rds_kill_query_id(queryID);
```

Parametri

queryID

Numero intero. L'identità della query da terminare.

Note per l'utilizzo

Per arrestare una query in esecuzione nel server MariaDB, utilizza la procedura `mysql.rds_kill_query_id` e invia l'ID di quella query. Per ottenere l'ID query, esegui la query nella Information Schema PROCESSLIST Table di MariaDB come mostrato di seguito:

```
SELECT USER, HOST, COMMAND, TIME, STATE, INFO, QUERY_ID FROM
               INFORMATION_SCHEMA.PROCESSLIST WHERE USER = '<user name>';
```

La connessione al server MariaDB viene mantenuta.

Examples (Esempi)

L'esempio seguente termina una query con un ID query di 230040:

```
call mysql.rds_kill_query_id(230040);
```

Fuso orario locale per le istanze database MariaDB

Per impostazione predefinita, il fuso orario per un'istanza database di MariaDB è in formato Universal Time Coordinated (UTC). Puoi impostare il fuso orario per l'istanza database sul fuso orario locale dell'applicazione.

Per impostare il fuso orario locale per un'istanza database, imposta il parametro `time_zone` nel gruppo di parametri per l'istanza database su uno dei valori supportati elencati più avanti in questa sezione. Quando imposti il parametro `time_zone` per un gruppo di parametri, tutte le istanze database e le repliche di lettura che utilizzano tale gruppo di parametri cambiano per utilizzare il nuovo fuso orario locale. Per informazioni sull'impostazione dei parametri in un gruppo di parametri, consulta Utilizzo di gruppi di parametri.

Dopo aver impostato il fuso orario locale, tutte le nuove connessioni al database riflettono la modifica. Se ci sono connessioni aperte al database quando modifichi il fuso orario locale, questo non viene aggiornato fino a quando non chiudi la connessione e ne apri una nuova.

Puoi impostare un fuso orario locale diverso per un'istanza database e una o più delle relative repliche di lettura. A tale scopo, utilizza un gruppo di parametri diverso per l'istanza database e la replica o le repliche e imposta il parametro `time_zone` in ogni gruppo di parametri su un fuso orario locale diverso.

Se esegui la replica tra Regioni AWS, l'istanza database di origine e la replica di lettura utilizzano gruppi di parametri diversi (i gruppi di parametri sono univoci per una Regione AWS). Per utilizzare lo stesso fuso orario locale per ogni istanza, imposta il parametro `time_zone` nei gruppi di parametri dell'istanza e della replica di lettura.

Quando ripristini un'istanza database da uno snapshot DB, il fuso orario locale è impostato su UTC. Puoi aggiornare il fuso orario impostandolo sul fuso orario locale dopo il completamento del ripristino. Se ripristini un'istanza database a un punto nel tempo, il fuso orario locale per l'istanza database ripristinata corrisponde all'impostazione del fuso orario per il gruppo di parametri dell'istanza database ripristinata.

Internet Assigned Numbers Authority (IANA) pubblica nuovi fusi orari all'indirizzo https://www.iana.org/time-zones più volte all'anno. Ogni volta che RDS rilascia una nuova versione di manutenzione secondaria di MariaDB, la versione viene fornita con i dati sul fuso orario più recenti al momento del rilascio. Quando utilizzi le versioni più recenti di RDS per MariaDB, hai a disposizione i dati recenti relativi ai fusi orari di RDS. Per assicurarti che l'istanza DB disponga dei dati più aggiornati relativi ai fusi orari, ti consigliamo di eseguire l'aggiornamento a una versione superiore

del motore DB. In alternativa, puoi modificare manualmente le tabelle dei fusi orari nelle istanze DB MariaDB. A tale scopo, puoi utilizzare i comandi SQL o eseguire lo strumento mysql_tzinfo_to_sql in un client SQL. Dopo l'aggiornamento manuale dei dati dei fusi orari, avvia l'istanza database per applicare le modifiche. RDS non modifica né ripristina i dati dei fusi orari delle istanze DB in esecuzione. I nuovi dati dei fusi orari vengono installati solo quando si esegue un aggiornamento della versione del motore di database.

Puoi impostare il fuso orario locale su uno dei valori seguenti.

Africa/Cairo	Asia/Riyadh
Africa/Casablanca	Asia/Seoul
Africa/Harare	Asia/Shanghai
Africa/Monrovia	Asia/Singapore
Africa/Nairobi	Asia/Taipei
Africa/Tripoli	Asia/Tehran
Africa/Windhoek	Asia/Tokyo
America/Araguaina	Asia/Ulaanbaatar
America/Asuncion	Asia/Vladivostok
America/Bogota	Asia/Yakutsk
America/Buenos_Aires	Asia/Yerevan
America/Caracas	Atlantic/Azores
America/Chihuahua	Australia/Adelaide
America/Cuiaba	Australia/Brisbane
America/Denver	Australia/Darwin
America/Fortaleza	Australia/Hobart

America/Guatemala	Australia/Perth
America/Halifax	Australia/Sydney
America/Manaus	Brazil/East
America/Matamoros	Canada/Newfoundland
America/Monterrey	Canada/Saskatchewan
America/Montevideo	Canada/Yukon
America/Phoenix	Europe/Amsterdam
America/Santiago	Europe/Athens
America/Tijuana	Europe/Dublin
Asia/Amman	Europe/Helsinki
Asia/Ashgabat	Europe/Istanbul
Asia/Baghdad	Europe/Kaliningrad
Asia/Baku	Europe/Moscow
Asia/Bangkok	Europe/Paris
Asia/Beirut	Europe/Prague
Asia/Calcutta	Europe/Sarajevo
Asia/Damascus	Pacific/Auckland
Asia/Dhaka	Pacific/Fiji
Asia/Irkutsk	Pacific/Guam
Asia/Jerusalem	Pacific/Honolulu
Asia/Kabul	Pacific/Samoa

Asia/Karachi	US/Alaska
Asia/Kathmandu	US/Central
Asia/Krasnoyarsk	US/Eastern
Asia/Magadan	US/East-Indiana
Asia/Muscat	US/Pacific
Asia/Novosibirsk	UTC

Problemi e limitazioni note per RDS per MariaDB

I seguenti elementi sono problemi e limitazioni noti quando si utilizza RDS per MariaDB.

> **Note**
>
> L'elenco non è completo.

Argomenti

Limiti delle dimensioni dei file MariaDB in Amazon RDS

Per le istanze database MariaDB, le dimensioni massime di una tabella è 16 TB quando si usano spazi tabelle file-per-table InnoDB. Questo valore limita anche il tablespace di sistema a una dimensione massima di 16 TB. Gli spazi tabelle file-per-table InnoDB (con ciascuna tabella nel relativo spazio tabelle) sono configurati per impostazione predefinita per le istanze database MariaDB. Questo limite non è correlato al limite massimo di archiviazione per le istanze database MariaDB. Per ulteriori informazioni sui limiti di archiviazione, consulta Storage delle istanze di database Amazon RDS.

Ci sono vantaggi e svantaggi nell'utilizzare i tablespaces file-per-table InnoDB, a seconda dell'applicazione. Per stabilire l'approccio migliore per l'applicazione, consulta File-Per-Table Tablespaces nella documentazione MySQL.

Non è consigliabile consentire alle tabelle di crescere fino alla dimensione massima del file. In generale, una pratica migliore consiste nel partizionare i dati in tabelle più piccole, che possano migliorare le prestazioni e i tempi di ripristino.

Un'opzione che è possibile utilizzare per suddividere una tabella di grandi dimensioni in tabelle più piccole è rappresentata dal partizionamento. Il partizionamento distribuisce parti della tabella di grandi dimensioni in file separati in base alle regole specificate. Ad esempio, se si archiviano le transazioni per data, è possibile creare regole di partizionamento che distribuiscono le transazioni

meno recenti in file separati mediante il partizionamento. Quindi, periodicamente, è possibile archiviare i dati storici della transazione che non devono essere prontamente disponibili per l'applicazione. Per ulteriori informazioni, consulta Partitioning nella documentazione MySQL.

Determinazione della dimensione di tutti gli spazi tabella InnoDB

- Utilizza il seguente comando SQL per stabilire se qualche tabella supera le dimensioni consentite e può essere scelta per il partizionamento.

> **Note**
>
> Per MariaDB 10.6 e versioni successive, questa query restituisce anche la dimensione dello spazio tabella del sistema InnoDB.
> Per le versioni di MariaDB precedenti alla 10.6, non è possibile determinare la dimensione dello spazio tabella del sistema InnoDB mediante query sulle tabelle di sistema. Consigliamo di eseguire l'aggiornamento a una versione più recente.

```
SELECT SPACE,NAME,ROUND((ALLOCATED_SIZE/1024/1024/1024), 2)
as "Tablespace Size (GB)"
FROM information_schema.INNODB_SYS_TABLESPACES ORDER BY 3 DESC;
```

Determinazione della dimensione delle tabelle utente non InnoDB

- Utilizza il seguente comando SQL per stabilire se qualche tabella utente non InnoDB ha dimensioni eccessive.

```
SELECT TABLE_SCHEMA, TABLE_NAME, round(((DATA_LENGTH + INDEX_LENGTH+DATA_FREE)
/ 1024 / 1024/ 1024), 2) As "Approximate size (GB)" FROM information_schema.TABLES
WHERE TABLE_SCHEMA NOT IN ('mysql', 'information_schema', 'performance_schema')
and ENGINE<>'InnoDB';
```

Per abilitare gli spazi tabelle file-per-table InnoDB

- Imposta il parametro `innodb_file_per_table` su 1 nel gruppo di parametri per l'istanza database.

Per disabilitare gli spazi tabelle file-per-table InnoDB

- Imposta il parametro `innodb_file_per_table` su `0` nel gruppo di parametri per l'istanza database.

Per informazioni sull'aggiornamento di un gruppo di parametri database, consulta Utilizzo di gruppi di parametri.

Una volta abilitati o disabilitati gli spazi tabelle file-per-table InnoDB, puoi eseguire il comando `ALTER TABLE`. È possibile utilizzare questo comando per spostare una tabella dallo spazio tabelle globale al proprio spazio tabelle. Oppure è possibile spostare una tabella dal proprio spazio tabelle allo spazio tabelle globale. Di seguito è riportato un esempio.

```
ALTER TABLE table_name ENGINE=InnoDB, ALGORITHM=COPY;
```

Parola riservata InnoDB

`InnoDB` è una parola riservata per RDS for MariaDB. Non è possibile utilizzare questo nome per un database MariaDB.

Porte personalizzate

Amazon RDS blocca le connessioni alla porta personalizzata 33060 per il motore MariaDB. Scegli una porta diversa per il motore MariaDB.

Approfondimenti sulle prestazioni

I contatori InnoDB non sono visibili in Approfondimenti sulle prestazioni di Amazon RDS per MariaDB versione 10.11 perché la community MariaDB non li supporta più.

Amazon RDS for Microsoft SQL Server

Amazon RDS supporta l'esecuzione di versioni ed edizioni diverse di Microsoft SQL Server. Nella seguente tabella è riportata la versione più recente supportata di ciascuna versione principale. Per l'elenco completo delle versioni, delle edizioni e delle versioni del motore RDS supportate, consulta Versioni di Microsoft SQL Server su Amazon RDS.

Versione principale	Service Pack/ GDR	Aggiornamento cumulativo	Versione secondaria	Articolo della Knowledge Base	Data di rilascio
SQL Server 2022	GDR	CU12	160,4120,1	KB5036343	9 aprile 2024
SQL Server 2019	–	CU26	15,0,4365,2	KB5035123	11 aprile 2024
SQL Server 2017	GDR	CU31	14.0.34651	KB5029376	10 ottobre 2023
SQL Server 2016	SP3 GDR	–	13,0,6435,1	KB5029186	10 ottobre 2023
SQL Server 2014	SP3 GDR	CU4	12,0,6449,1	KB5029185	10 ottobre 2023

Per ulteriori informazioni sulla licenza di SQL Service Limits, consulta Licenza per Microsoft SQL Server su Amazon RDS. Per informazioni su come si compila il Server SQL, consulta l'articolo di supporto Microsoft sull'ultimo SQL Server.

Con Amazon RDS, puoi creare istanze DB e snapshot DB, point-in-time ripristini e backup automatici o manuali. È possibile utilizzare istanze database che eseguono SQL Service all'interno di un VPC. Puoi anche utilizzare il protocollo Secure Sockets Layer (SSL) per connetterti a un'istanza database che esegue SQL Server e utilizzare Transparent Data Encryption (TDE) per crittografare i dati a riposo. Amazon RDS attualmente supporta le implementazioni Multi-AZ per SQL Server tramite SQL

Server Database Mirroring (DBM) o i gruppi di disponibilità Always On come soluzione di failover a disponibilità elevata.

Per fornire un'esperienza di servizio gestito, Amazon RDS non fornisce accesso shell alle istanze database e limita l'accesso a certe procedure e tabelle di sistema che richiedono privilegi avanzati. Amazon RDS supporta l'accesso ai database su un'istanza database mediante qualsiasi applicazione client SQL standard come Microsoft SQL Server Management Studio. Amazon RDS non permette l'accesso host diretto a un'istanza database tramite Telnet, Secure Shell (SSH) o Connessione Desktop remoto Windows. Quando crei un'istanza database, l'utente master viene assegnato al ruolo db_owner per tutti i database dell'utente nell'istanza e dispone di tutte le autorizzazioni a livello di database ad eccezione di quelle utilizzate per i backup. Amazon RDS gestisce i backup per tuo conto.

Prima di creare la prima istanza database, è necessario completare le fasi nella sezione di questa guida relativa alla configurazione. Per ulteriori informazioni, consulta Configurazione di Amazon RDS.

Argomenti

- Connessione a un'istanza database che esegua il motore di database di Microsoft SQL Server
- Utilizzo di Active Directory con RDS per SQL Server
- Aggiornamento delle applicazioni per la connessione a istanze di database Microsoft SQL Server utilizzando nuovi certificati SSL/TLS
- Aggiornamento del motore di database Microsoft SQL Server
- Importazione ed esportazione di database SQL Server mediante backup e ripristino nativi
- Utilizzo di repliche di lettura per Microsoft SQL Server in Amazon RDS
- Implementazioni Multi-AZ per Amazon RDS for Microsoft SQL Server
- Funzionalità opzionali per Microsoft SQL Server su Amazon RDS
- Opzioni per il motore di database di Microsoft SQL Server
- Attività DBA frequenti per Microsoft SQL Server

Attività di gestione frequenti per Microsoft SQL Server su Amazon RDS

Di seguito sono riportate le attività di gestione più frequenti che puoi eseguire con un'istanza database Amazon RDS for SQL Server, con collegamenti alla documentazione rilevante per ciascuna attività.

Area attività	Documentazione di riferimento
Classi delle istanze, storage e PIOPS Se stai creando un'istanza database per la produzione, è necessario comprendere come funzionano in Amazon RDS le classi di istanze, i tipi di storage e le Provisioned IOPS.	Supporto classe istanza database per Microsoft SQL Server Tipi di storage Amazon RDS
Implementazioni Multi-AZ Un'istanza database in produzione deve utilizzare implement azioni Multi-AZ. Le implementazioni Multi-AZ forniscono alle istanze database maggior disponibilità, longevità dei dati e tolleranza ai guasti. Le implementazioni Multi-AZ per SQL Server sono implementate mediante la tecnologia AG o DBM nativa di SQL Server.	Configurazione e gestione di un'implementazione multi-AZ Le implementazioni Multi-AZ utilizzando Microsoft SQL Server Database Mirroring o i gruppi di disponibilità Always On

Area attività	Documentazione di riferimento
Amazon Virtual Private Cloud (VPC) Se il tuo AWS account ha un VPC predefinito, l'istanza DB viene creata automaticamente all'interno del VPC predefinito. Se l'account non ha un VPC predefinito e desideri che l'istanza database sia in un VPC, è necessario creare il VPC e i gruppi di sottoreti prima di creare l'istanza database.	Uso di un'istanza database in un VPC
Gruppi di sicurezza Per impostazione predefinita, le istanze database vengono create con un firewall che ne impedisce l'accesso. Per accedere all'istanza database, devi quindi creare un gruppo di sicurezza con gli indirizzi IP e la configurazione di rete corretti.	Controllo dell'accesso con i gruppi di sicurezza
Gruppi di parametri Se l'istanza database richiede parametri database specifici , è necessario creare un gruppo di parametri prima di creare l'istanza database.	Utilizzo di gruppi di parametri
Gruppi di opzioni Se l'istanza database richiede opzioni database specifiche, è necessario creare un gruppo di opzioni prima di creare l'istanza database.	Opzioni per il motore di database di Microsoft SQL Server
Connessione all'istanza database Dopo aver creato un gruppo di sicurezza e averlo associato a un'istanza database, puoi effettuare la connessione all'istanza database mediante un'applicazione client SQL standard come Microsoft SQL Server Management Studio.	Connessione a un'istanza database che esegua il motore di database di Microsoft SQL Server

Area attività	Documentazione di riferimento
Backup e ripristino Quando crei l'istanza database, puoi configurarla per eseguire backup automatici. Puoi inoltre eseguire il backup e il ripristino dei database manualmente utilizzando file di backup completi (file .bak).	Introduzione ai backup Importazione ed esportazione di database SQL Server mediante backup e ripristino nativi
Monitoraggio Puoi monitorare la tua istanza DB di SQL Server utilizzando metriche, eventi e monitoraggio avanzato di CloudWatch Amazon RDS.	Visualizzazione dei parametri nella console Amazon RDS Visualizzazione di eventi Amazon RDS
File di log Puoi accedere ai file di log per l'istanza database SQL Server.	Monitoraggio dei file di log di Amazon RDS File di log di database Microsoft SQL Server

Esistono inoltre attività amministrative avanzate per l'utilizzo di istanze database che eseguono SQL Service. Per ulteriori informazioni, consulta la seguente documentazione:

- Attività DBA frequenti per Microsoft SQL Server.
- Utilizzo di Active Directory gestito da AWS con RDS per SQL Server
- Accesso al database tempdb

Restrizioni per le istanze database di Microsoft SQL Server

L'implementazione Amazon RDS di Microsoft SQL Server su un'istanza database presenta alcune limitazioni da tenere presente:

- Il numero massimo di database supportati su un'istanza database dipende dal tipo di classe di istanza e dalla modalità di disponibilità: zona di disponibilità singola, DBM (Database Mirroring) Multi-AZ o Gruppi di disponibilità Multi-AZ. I database di sistema Microsoft SQL Server non vanno oltre questo limite.

La tabella seguente mostra il numero massimo di database supportati per ogni tipo di classe di istanza e modalità di disponibilità. L'utilizzo di questa tabella ti aiuta a decidere se è possibile passare da un tipo di istanza a un altro o da una modalità di disponibilità a un'altra. Se l'istanza database di origine ha più database del tipo di istanza di destinazione o più modalità di disponibilità di quanti ne possa supportare, la modifica delle istanza database non riesce. È possibile visualizzare lo stato della richiesta nel riquadro Events (Eventi).

Tipo di classe di istanza	Single-AZ	Multi-AZ con DBM	Multi-AZ con gruppi di disponibilità sempre attivi
Da db.*.micro a db.*.medium	30	N/D	N/D
db.*.large	30	30	30
Da db.*.xlarge a db.*.16xlarge	100	50	75
db.*.24xlarge	100	50	100

* Rappresenta i tipi di classi di istanza differenti.

Ad esempio, supponiamo che l'istanza database venga eseguita su una db.*.16xlarge con una Single-AZ e che abbia 76 database. Modifica l'istanza database a cui effettuare l'aggiornamento utilizzando Multi-AZ sempre attivo. Questo aggiornamento non riesce perché l'istanza database contiene altri database rispetto a quelli che la configurazione di destinazione è in grado di supportare. Se invece si aggiorna il tipo di classe istanza a db.*.24xlarge, la modifica riesce.

Se l'aggiornamento non riesce, vengono visualizzati messaggi ed eventi simili ai seguenti:

- Impossibile modificare la classe dell'istanza database. L'istanza dispone di 76 database ma, dopo la conversione, ne supporterebbe solo 75.
- Impossibile convertire l'istanza database in Multi-AZ: l'istanza dispone di 76 database ma, dopo la conversione, ne supporterebbe solo 75.

Se il point-in-time ripristino o il ripristino dello snapshot falliscono, vengono visualizzati eventi e messaggi simili ai seguenti:

  - Istanza database inserita in un ripristino incompatibile. L'istanza dispone di 76 database ma, dopo la conversione, ne supporterebbe solo 75.
- Le seguenti porte sono riservate a Amazon RDS e non è possibile utilizzarle quando si crea un'istanza database: `1234`, `1434`, `3260`, `3343`, `3389`, `47001`, and `49152-49156`.
- Le connessioni client da indirizzi IP all'interno dell'intervallo 169.254.0.0/16 non sono consentite. Si tratta di un intervallo APIPA (Automatic Private IP Addressing), utilizzato per gli indirizzi con collegamenti locali.
- SQL Server Standard Edition utilizza solo un sottogruppo di processori disponibili se l'istanza database ha più processori dei limiti software (24 core, 4 socket e 128GB RAM). Ne sono esempi le classi di istanza db.m5.24xlarge e db.r5.24xlarge.

  Per ulteriori informazioni, consulta la tabella dei limiti di scala in Edizioni e funzionalità supportate di SQL Server 2019 (15.x) nella documentazione Microsoft.
- Amazon RDS for SQL Server non supporta l'importazione di dati nel database msdb.
- Non è possibile rinominare i database su un'istanza database presente in un'implementazione Multi-AZ di SQL Server.
- Assicurarsi di utilizzare queste linee guida quando si impostano i seguenti parametri DB su RDS per SQL Server:
  - `max server memory (mb)` >= 256 MB
  - `max worker threads` >= (numero di CPU logiche * 7)

  Per ulteriori informazioni sull'impostazione dei parametri DB, consulta Utilizzo di gruppi di parametri.
- La dimensione massima dello storage per le istanze database SQL Service è la seguente:
  - Storage General Purpose (SSD) – 16 TiB per tutte le edizioni
  - Storage IOPS fornito – 16 TiB per tutte le edizioni
  - Storage magnetico – 1 TiB per tutte le edizioni

  Se si dispone di uno scenario che richiede una quantità maggiore di spazio di archiviazione, è possibile utilizzare lo sharding su più istanze database per aggirare il limite. Questo approccio richiede una logica di routing dipendente dai dati nelle applicazioni che si collegano al sistema frammentato. È possibile utilizzare un framework di sharding esistente o scrivere un codice personalizzato per abilitare lo sharding. Se si utilizza un framework esistente, il framework non può installare alcun componente sullo stesso server dell'istanza database.
- La dimensione minima dell'archiviazione per le istanze database SQL Service è la seguente:

  - Storage per uso generico (SSD) – 20 GiB per edizioni Enterprise, Standard, Web ed Express
  - Storage IOPS assegnate – 20 GiB per edizioni Enterprise, Standard, Web ed Express
  - Storage magnetico – 20 GiB per edizioni Enterprise, Standard, Web ed Express
- Amazon RDS non supporta l'esecuzione di questi servizi sullo stesso server dell'istanza database RDS:
  - Servizi di Qualità dei Dati
  - Servizi dei dati principali

  Per utilizzare queste caratteristiche, consigliamo di installare il server SQL su un'istanza Amazon EC2 o utilizzare un'istanza SQL Server locale. In questi casi, l'istanza EC2 o SQL Server agisce da server Master Data Services per l'istanza database SQL Server su Amazon RDS. È possibile installare SQL Server in un'istanza Amazon EC2 con lo storage Amazon EBS, in conformità alle policy di licenza Microsoft.
- A causa delle limitazioni del Server di Microsoft SQL, il ripristino a un punto nel tempo prima dell'esecuzione corretta di un `DROP DATABASE` potrebbe non riflettere lo stato di quel database in quel momento. Ad esempio, il database rilasciato viene generalmente ripristinato al suo stato fino a 5 minuti prima che venga emesso il comando `DROP DATABASE`. Questo tipo di ripristino indica che è impossibile ripristinare le transazioni effettuate durante questi pochi minuti sul database rilasciato. Per risolvere il problema, è possibile riemettere il comando `DROP DATABASE` al termine dell'operazione di ripristino. L'eliminazione di un database rimuove i registri delle transazioni per quel database.
- In SQL Server, i database vengono creati dopo l'istanza database. I nomi dei database seguono le normali regole di denominazione di SQL Server con le seguenti differenze:
  - I nomi dei database non possono iniziare con `rdsadmin`.
  - Non possono iniziare né terminare con uno spazio o una tabulazione.
  - Non possono contenere nessuno dei caratteri che crea una nuova riga.
  - Non possono contenere una virgoletta singola (`'`).
  - RDS per SQL Server attualmente non supporta gli aggiornamenti automatici delle versioni secondarie. Per ulteriori informazioni, consulta Gestione della versione in Amazon RDS.
- SQL Server Web Edition consente di utilizzare il modello Dev/Test solo durante la creazione di una nuova istanza DB RDS per SQL Server.

Supporto classe istanza database per Microsoft SQL Server

La capacità di calcolo e memoria di un'istanza database è determinata dalla sua classe di istanza database. La classe di istanza database di cui hai bisogno dipende dalla potenza di elaborazione e dai requisiti di memoria specifici. Per ulteriori informazioni, consulta Classi di istanze database.

Il seguente elenco delle classi di istanza database supportate per Microsoft SQL Server viene qui fornito per maggiore comodità. Per l'elenco aggiornato, visita la console RDS: https://console.aws.amazon.com/rds/.

Non tutte le classi di istanze database sono disponibili in tutte le versioni secondarie supportate di SQL Server. Ad esempio, alcune classi di istanze database più recenti come db.r6i non sono disponibili nelle versioni secondarie precedenti. È possibile utilizzare il AWS CLI comando describe-orderable-db-instance-options per scoprire quali classi di istanze DB sono disponibili per l'edizione e la versione di SQL Server.

SQL Server Edition	Intervallo di supporto 2022	Intervallo di supporto 2019	Intervallo di supporto 2017 e 2016	Intervallo di supporto 2014
Enterprise Edition	db.t3.xlarge –db.t3.2xlarge	db.t3.xlarge –db.t3.2xlarge	db.t3.xlarge –db.t3.2xlarge	db.t3.xlarge –db.t3.2xlarge
	db.r5.large –db.r5.24xlarge	db.r5.xlarge –db.r5.24xlarge	db.r3.xlarge –db.r3.8xlarge	db.r3.xlarge –db.r3.8xlarge
	db.r5b.large –db.r5b.24xlarge	db.r5b.xlarge –db.r5b.24xlarge	db.r4.xlarge –db.r4.16xlarge	db.r4.xlarge –db.r4.8xlarge
	db.r5d.large –db.r5d.24xlarge	db.r5d.xlarge –db.r5d.24xlarge	db.r5.xlarge –db.r5.24xlarge	db.r5.xlarge –db.r5.24xlarge
	db.r6i.large –db.r6i.32xlarge	db.r6i.xlarge –db.r6i.32xlarge	db.r5b.xlarge –db.r5b.24xlarge	db.r5b.xlarge –db.r5b.24xlarge

SQL Server Edition	Intervallo di supporto 2022	Intervallo di supporto 2019	Intervallo di supporto 2017 e 2016	Intervallo di supporto 2014
	db.m5.large –db.m5.24xlarge	db.m5.xlarge –db.m5.24xlarge	db.r5d.xlarge –db.r5d.24xlarge	db.r5d.xlarge –db.r5d.24xlarge
	db.m5d.large –db.m5d.24xlarge	db.m5d.xlarge –db.m5d.24xlarge	db.r6i.xlarge –db.r6i.32xlarge	db.r6i.xlarge –db.r6i.32xlarge
	db.m6i.large –db.m6i.32xlarge	db.m6i.xlarge –db.m6i.32xlarge	db.m4.xlarge –db.m4.16xlarge	db.m4.xlarge –db.m4.10xlarge
	db.x2iedn.xlarge –db.x2iedn.32xlarge	db.x1.16xlarge –db.x1.32xlarge	db.m5.xlarge –db.m5.24xlarge	db.m5.xlarge –db.m5.24xlarge
	db.z1d.large –db.z1d.12xlarge	db.x1e.xlarge –db.x1e.32xlarge	db.m5d.xlarge –db.m5d.24xlarge	db.m5d.xlarge –db.m5d.24xlarge
		db.x2iedn.xlarge –db.x2iedn.32xlarge	db.m6i.xlarge –db.m6i.32xlarge	db.m6i.xlarge –db.m6i.32xlarge
		db.z1d.xlarge –db.z1d.12xlarge	db.x1.16xlarge –db.x1.32xlarge	db.x1.16xlarge –db.x1.32xlarge
			db.x1e.xlarge –db.x1e.32xlarge	db.x1e.xlarge –db.x1e.32xlarge
			db.x2iedn.xlarge –db.x2iedn.32xlarge	db.x2iedn.xlarge –db.x2iedn.32xlarge

SQL Server Edition	Intervallo di supporto 2022	Intervallo di supporto 2019	Intervallo di supporto 2017 e 2016	Intervallo di supporto 2014
			`db.z1d.xlarge –db.z1d.12xlarge`	

SQL Server Edition	Intervallo di supporto 2022	Intervallo di supporto 2019	Intervallo di supporto 2017 e 2016	Intervallo di supporto 2014
Standard Edition	db.t3.xlarge –db.t3.2xlarge db.r5.large –db.r5.24xlarge db.r5b.large –db.r5b.8xlarge db.r5d.large –db.r5d.24xlarge db.r6i.large –db.r6i.8xlarge db.m5.large –db.m5.24xlarge db.m5d.large –db.m5d.24xlarge db.m6i.large –db.m6i.8xlarge db.x2iedn.xlarge –db.x2iedn.8xlarge	db.t3.xlarge –db.t3.2xlarge db.r5.large –db.r5.24xlarge db.r5b.large –db.r5b.24xlarge db.r5d.large –db.r5d.24xlarge db.r6i.large –db.r6i.8xlarge db.m5.large –db.m5.24xlarge db.m5d.large –db.m5d.24xlarge db.m6i.large –db.m6i.8xlarge db.x1.16xlarge –db.x1.32xlarge	db.t3.xlarge –db.t3.2xlarge db.r4.large –db.r4.16xlarge db.r5.large –db.r5.24xlarge db.r5b.large –db.r5b.24xlarge db.r5d.large –db.r5d.24xlarge db.r6i.large –db.r6i.8xlarge db.m4.large –db.m4.16xlarge db.m5.large –db.m5.24xlarge db.m5d.large –db.m5d.24xlarge	db.t3.xlarge –db.t3.2xlarge db.r3.large –db.r3.8xlarge db.r4.large –db.r4.8xlarge db.r5.large –db.r5.24xlarge db.r5b.large –db.r5b.24xlarge db.r5d.large –db.r5d.24xlarge db.r6i.large –db.r6i.8xlarge db.m3.medium –db.m3.2xlarge db.m4.large –db.m4.10xlarge

SQL Server Edition	Intervallo di supporto 2022	Intervallo di supporto 2019	Intervallo di supporto 2017 e 2016	Intervallo di supporto 2014
	db.z1d.large –db.z1d.12xlarge	db.x1e.xlarge –db.x1e.32xlarge db.x2iedn.xlarge –db.x2iedn.32xlarge db.z1d.large –db.z1d.12xlarge	db.m6i.large –db.m6i.8xlarge db.x1.16xlarge –db.x1.32xlarge db.x1e.xlarge –db.x1e.32xlarge db.x2iedn.xlarge –db.x2iedn.32xlarge db.z1d.large –db.z1d.12xlarge	db.m5.large –db.m5.24xlarge db.m5d.large –db.m5d.24xlarge db.m6i.large –db.m6i.8xlarge db.x1.16xlarge –db.x1.32xlarge db.x1e.xlarge –db.x1e.32xlarge db.x2iedn.xlarge –db.x2iedn.32xlarge

SQL Server Edition	Intervallo di supporto 2022	Intervallo di supporto 2019	Intervallo di supporto 2017 e 2016	Intervallo di supporto 2014
Web Edition	db.t3.small – db.t3.xlarge db.r5.large – db.r5.4xlarge db.r5b.large – db.r5b.4xlarge db.r5d.large – db.r5d.4xlarge db.r6i.large – db.r6i.4xlarge db.m5.large – db.m5.4xlarge db.m5d.large – db.m5d.4xlarge db.m6i.large – db.m6i.4xlarge db.z1d.large – db.z1d.13xlarge	db.t3.small – db.t3.2xlarge db.r5.large – db.r5.4xlarge db.r5b.large – db.r5b.4xlarge db.r5d.large – db.r5d.4xlarge db.r6i.large – db.r6i.4xlarge db.m5.large – db.m5.4xlarge db.m5d.large – db.m5d.4xlarge db.m6i.large – db.m6i.4xlarge db.z1d.large – db.z1d.3xlarge	db.t2.small – db.t2.medium db.t3.small – db.t3.2xlarge db.r4.large – db.r4.2xlarge db.r5.large – db.r5.4xlarge db.r5b.large – db.r5b.4xlarge db.r5d.large – db.r5d.4xlarge db.r6i.large – db.r6i.4xlarge db.m4.large – db.m4.4xlarge db.m5.large – db.m5.4xlarge	db.t2.small – db.t2.medium db.t3.small – db.t3.2xlarge db.r3.large – db.r3.2xlarge db.r4.large – db.r4.2xlarge db.r5.large – db.r5.4xlarge db.r5b.large – db.r5b.4xlarge db.r5d.large – db.r5d.4xlarge db.r6i.large – db.r6i.4xlarge db.m3.medium – db.m3.2xlarge

SQL Server Edition	Intervallo di supporto 2022	Intervallo di supporto 2019	Intervallo di supporto 2017 e 2016	Intervallo di supporto 2014
			db.m5d.large –db.m5d.4xlarge db.m6i.large –db.m6i.4xlarge db.z1d.large –db.z1d.3xlarge	db.m4.large –db.m4.4xlarge db.m5.large –db.m5.4xlarge db.m5d.large –db.m5d.4xlarge db.m6i.large –db.m6i.4xlarge
Express Edition	db.t3.micro –db.t3.xlarge	db.t3.micro –db.t3.xlarge	db.t2.micro –db.t2.medium db.t3.micro –db.t3.xlarge	db.t2.micro –db.t2.medium db.t3.micro –db.t3.xlarge

Sicurezza del Server Microsoft SQL

Il motore del database del Server Microsoft SQL utilizza la protezione basata sui ruoli. Il nome utente principale che si utilizza quando si crea un'istanza database è un accesso di autenticazione del Server SQL che è un membro del `processadmin`, `public`, e `setupadmin` ruoli del server fisso.

Quando un utente crea un database, gli viene assegnato il ruolo db_owner per tale database e dispone di tutte le autorizzazioni a livello di database ad eccezione di quelle utilizzate per i backup. Amazon RDS gestisce i backup per tuo conto.

I seguenti ruoli a livello di server non sono disponibili in Amazon RDS for SQL Server:

- bulkadmin
- dbcreator
- diskadmin
- securityadmin
- serveradmin
- sysadmin

Le seguenti autorizzazioni a livello server non sono disponibili sulle istanze database RDS per SQL Server:

- ALTER ANY DATABASE
- ALTER ANY EVENT NOTIFICATION
- ALTER RESOURCES
- ALTER SETTINGS (è possibile utilizzare le operazioni API del gruppo di parametri del database per modificare i parametri; per ulteriori informazioni, consulta Utilizzo di gruppi di parametri)
- AUTENTICAZIONE AL SERVER
- CONTROL_SERVER
- CREAZIONE NOTIFICA AD EVENTO DDL
- CREAZIONE ENDPOINT
- CREAZIONE DI UN RUOLO SERVER
- CREAZIONE NOTIFICA SULL'EVENTO TRACE
- DROP ANY DATABASE
- ASSEMBLAGGIO ACCESSO ESTERNO
- ARRESTO (È possibile invece utilizzare l'opzione di riavvio RDS)
- ASSEMBLAGGIO NON SICURO
- ALTER ANY AVAILABILITY GROUP
- CREATE ANY AVAILABILITY GROUP

Supporto del Programma di Conformità per le istanze di database di Microsoft SQL Server

AWS I servizi in questione sono stati completamente valutati da un revisore esterno e hanno prodotto una certificazione, un attestato di conformità o l'Authority to Operate (ATO). Per ulteriori informazioni, consulta Servizi AWS coperti dal programma di compliance.

Supporto HIPAA per le istanze di database di Microsoft SQL Server

Puoi utilizzare i database Amazon RDS for Microsoft SQL Server per creare applicazioni conformi a HIPAA. Puoi archiviare informazioni sanitarie, inclusi dati sanitari protetti (PHI), in base a un Contratto di società in affari (BAA) con AWS. Per ulteriori informazioni, consulta Compliance HIPAA.

Amazon RDS for SQL Server supporta HIPAA per le seguenti versioni ed edizioni:

- Edizioni SQL Server 2022 Enterprise, Standard e Web
- SQL Server 2019: edizioni Enterprise, Standard e Web
- SQL Server 2017: edizioni Enterprise, Standard e Web
- SQL Server 2016: edizioni Enterprise, Standard e Web
- SQL Server 2014: edizioni Enterprise, Standard e Web

Per abilitare il supporto HIPAA sull'istanza database, impostare i seguenti tre componenti.

Componente	Informazioni
Audit	Per impostare il controllo, impostare il parametro `rds.sqlserver_audit` sul valore `fedramp_hipaa` . Se l'istanza database non sta già utilizzando un gruppo di parametri database personalizzati, è necessario creare un gruppo di parametri personalizzati e collegarlo all'istanza database prima di poter modificare il parametro `rds.sqlserver_audit` Per ulteriori informazioni, consulta Utilizzo di gruppi di parametri.
Crittografia del trasporto	Per impostare la crittografia del trasporto, forzare tutte le connessioni all'istanza database per utilizzare Secure Sockets Layer (SSL). Per ulteriori

Componente	Informazioni
	informazioni, consulta Imposizione dell'utilizzo di SSL per le connessioni all'istanza database.
Crittografia dei dati inattivi	Per impostare la crittografia a riposo, hai due opzioni: 1. Se utilizzi SQL Server 2014—2022 Enterprise Edition o 2022 Standard Edition, puoi utilizzare Transparent Data Encryption (TDE) per ottenere la crittografia a riposo. Per ulteriori informazioni, consulta Supporto per Transparent Data Encryption in SQL Server. 2. È possibile configurare la crittografia inattiva utilizzando le chiavi di crittografia AWS Key Management Service ()AWS KMS. Per ulteriori informazioni, consulta Crittografia delle risorse Amazon RDS.

Supporto SSL per istanze database di Microsoft SQL Server

Ora puoi utilizzare Secure Sockets Layer (SSL) per crittografare le connessioni tra le applicazioni dei tuoi client e le istanze database Amazon RDS che eseguono Microsoft SQL Server. Puoi inoltre imporre a tutte le connessioni per la tua istanza database di utilizzare SSL. Se forzi l'utilizzo di SSL per le connessioni, ciò avviene in modo trasparente per il client e il client non deve effettuare alcuna operazione per utilizzare SSL.

SSL è supportato in tutte le AWS regioni e per tutte le edizioni di SQL Server supportate. Per ulteriori informazioni, consulta Utilizzo di SSL con un'istanza database Microsoft SQL Server.

Versioni di Microsoft SQL Server su Amazon RDS

Quando crei una nuova istanza database, puoi specificare qualsiasi versione di Microsoft SQL Server attualmente supportata. Puoi specificare la versione principale di Microsoft SQL Server (ad esempio Microsoft SQL Server 14.00) e qualsiasi versione secondaria supportata per la versione principale specificata. Se non viene specificata alcuna versione, Amazon RDS utilizza per impostazione predefinita una versione supportata, in genere la più recente. Se viene specificata una versione principale ma non una secondaria, per impostazione predefinita Amazon RDS utilizza una release recente della versione principale specificata.

La tabella seguente mostra le versioni supportate per tutte le edizioni e tutte le AWS regioni, tranne dove indicato. È inoltre possibile utilizzare il `describe-db-engine-versions` AWS CLI comando per visualizzare un elenco di versioni supportate, nonché i valori predefiniti per le istanze DB appena create.

Versioni di SQL Server supportate in RDS

Versione principale	Versione secondaria	API **`EngineVersion`** e CLI RDS **`engine-version`**
SQL Server 2022	16.00.4120.1 (CU12 GDR)	`16.00.4120.1.v1`
	16,00,4115,5 (CU12)	`16.00.4115.5.v1`
	16,00,4105,2 (CU11)	`16.00.4105.2.v1`
	16,00,4095,4 (CU10)	`16.00.4095.4.v1`
	16,00,4085,2 (CU9)	`16.00.4085.2.v1`
SQL Server 2019	15,00,4365,2 (CU26)	`15.00.4365.2`
	15,00,435,3 (CU25)	`15.00.4355.3.v1`
	15,00,4345,5 (CU24)	`15.00.4345.5.v1`
	15.00.4335.1 (CU23)	`15.00.4335.1.v1`
	15.00.4322.2 (CU22)	`15.00.4322.2.v1`
	15.00.4316.3 (CU21)	`15.00.4316.3.v1`
	15.00.4312.2 (CU20)	`15.00.4312.2.v1`
	15.00.4236.7 (CU16)	`15.00.4236.7.v1`
	15.00.4198.2 (CU15)	`15.00.4198.2.v1`
	15.00.4153.1 (CU12)	`15.00.4153.1.v1`
	15.00.4073.23 (CU8)	`15.00.4073.23.v1`
	15.00.4043.16 (CU5)	

Versione principale	Versione secondaria	API **EngineVersion** e CLI RDS **engine-version**
		15.00.4043.16.v1
SQL Server 2017	14.00.3465.1 (CU31)	14.00.3465.1.v1
	14.00.3460.9 (CU31)	14.00.3460.9.v1
	14.00.3451.2 (CU30)	14.00.3451.2.v1
	14.00.3421.10 (CU27)	14.00.3421.10.v1
	14.00.3401.7 (CU25)	14.00.3401.7.v1
	14.00.3381.3 (CU23)	14.00.3381.3.v1
	14.00.3356.20 (CU22)	14.00.3356.20.v1
	14.00.3294.2 (CU20)	14.00.3294.2.v1
	14.00.3281.6 (CU19)	14.00.3281.6.v1
SQL Server 2016	13.00.6435.1 (GDR)	13.00.6435.1.v1
	13.00.6430.49 (GDR)	13.00.6430.49.v1
	13.00.6419.1 (SP3 + Hotfix)	13.00.6419.1.v1
	13.00.6300.2 (SP3)	13.00.6300.2.v1
SQL Server 2014	12.00.6449.1 (SP3 CU4 GDR)	12.00.6449.1.v1
	12.00.6444.4 (SP3 CU4 GDR)	12.00.6444.4.v1
	12.00.6439.10 (SP3 CU4 SU)	12.00.6439.10.v1
	12.00.6433.1 (SP3 CU4 SU)	12.00.6433.1.v1
	12.00.6329.1 (SP3 CU4)	12.00.6329.1.v1
	12.00.6293.0 (SP3 CU3)	12.00.6293.0.v1

Gestione della versione in Amazon RDS

Amazon RDS include una gestione flessibile della versione che consente di controllare quando e come applicare una patch o un aggiornamento all'istanza database. Ciò consente di svolgere le seguenti operazioni con il motore del database:

- Gestire la compatibilità con versioni di patch del motore del database.
- Testare nuove versioni di patch per verificare che funzionino con la propria applicazione prima di distribuirle in produzione.
- Pianificare ed eseguire aggiornamenti della versione per rispondere ai contratti sul livello di servizio e ai requisiti temporali

Creazione di patch del motore Microsoft SQL Server in Amazon RDS

Amazon RDS aggrega periodicamente le patch di database Microsoft SQL Server ufficiali a una versione del motore dell'istanza database specifica per Amazon RDS. Per ulteriori informazioni sulle patch Microsoft SQL Server in ogni versione del motore, consulta Versione e caratteristiche di supporto su Amazon RDS.

Attualmente, è possibile eseguire manualmente gli aggiornamenti del motore sull'istanza database. Per ulteriori informazioni, consulta Aggiornamento del motore di database Microsoft SQL Server.

Pianificazione dell'impostazione come obsoleto per le versioni principali del motore di Microsoft SQL Server su Amazon RDS

La tabella seguente visualizza le pianificazioni delle impostazioni come obsoleto per le versioni principali del motore di Microsoft SQL Server.

Data	Informazioni
9 luglio 2024	Microsoft interromperà gli aggiornamenti delle patch critiche per SQL Server 2014. P consulta Microsoft SQL Server 2014 nella documentazione Microsoft.
1 giugno 2024	Amazon RDS prevede di terminare il supporto di Microsoft SQL Server 2014 su RDS quel momento, tutte le istanze rimanenti verranno pianificate per la migrazione a SQL versione minore disponibile). Per ulteriori informazioni, consulta Announcement: Ama

Data	Informazioni
	Server ending support for SQL Server 2014 major versions (Annuncio: Amazon RDS termina il supporto delle versioni principali di SQL Server 2014). Per evitare un aggiornamento automatico da Microsoft SQL Server 2014, è possibile amento nel momento più comodo. Per ulteriori informazioni, consulta Aggiornamento motore di un'istanza database.
12 luglio 2022	Microsoft interromperà gli aggiornamenti delle patch critiche per SQL Server 2012. P consulta Microsoft SQL Server 2012 nella documentazione Microsoft.
1 giugno 2022	Amazon RDS prevede di terminare il supporto di Microsoft SQL Server 2012 su RDS quel momento, tutte le istanze rimanenti verranno pianificate per la migrazione a SQL versione minore disponibile). Per ulteriori informazioni, consulta Announcement: Ama Server ending support for SQL Server 2012 major versions. Per evitare un aggiornamento automatico da Microsoft SQL Server 2012, puoi effettu momento più comodo per te. Per ulteriori informazioni, consulta Aggiornamento della un'istanza database.
1 settembre 2021	Amazon RDS sta iniziando a disabilitare la creazione di nuovi RDS per istanze datab utilizzando Microsoft SQL Server 2012. Per ulteriori informazioni, consulta Announce SQL Server ending support for SQL Server 2012 major versions.
12 luglio 2019	Il team Amazon RDS ha impostato come obsoleto il supporto ìper Microsoft SQL Ser 2019. È in corso la migrazione di tutte le istanze rimanenti di Microsoft SQL Server 2 2012 (ultima versione minore disponibile). Per evitare un aggiornamento automatico da Microsoft SQL Server 2008 R2, puoi eff nel momento più comodo per te. Per ulteriori informazioni, consulta Aggiornamento d di un'istanza database.
25 aprile 2019	Entro la fine di aprile 2019, non sarà più possibile creare nuove istanze database Am Server utilizzando Microsoft SQL Server 2008R2.

Funzionalità di Microsoft SQL Server su Amazon RDS

Le versioni supportate di SQL Server su Amazon RDS includono le seguenti funzionalità. In generale, una versione include anche funzionalità delle versioni precedenti, salvo diversa indicazione nella documentazione Microsoft.

Argomenti

Funzionalità di Microsoft SQL Server 2022

SQL Server 2022 include molte nuove funzionalità, come le seguenti:

- Parameter Sensitive Plan Optimization: consente più piani memorizzati nella cache per una singola istruzione parametrizzata, riducendo potenzialmente i problemi relativi allo sniffing dei parametri.
- SQL Server Ledger: offre la possibilità di dimostrare crittograficamente che i dati non sono stati alterati senza autorizzazione.
- Inizializzazione istantanea dei file per gli eventi di crescita dei file di registro delle transazioni: consente un'esecuzione più rapida degli eventi di crescita dei log fino a 64 MB, anche per i database con TDE abilitato.
- Miglioramenti della concorrenza del sistema Page Latch: riduce il conflitto tra latch di pagine durante l'allocazione e la deallocazione di pagine ed estensioni di dati, offrendo significativi miglioramenti delle prestazioni in caso di carichi di lavoro pesanti. `tempdb`

Per l'elenco completo delle funzionalità di SQL Server 2022, vedi Novità di SQL Server 2022 (16.x) nella documentazione Microsoft.

Per un elenco delle funzionalità non supportate, consulta Caratteristiche non supportate e caratteristiche con supporto limitato.

Funzionalità di Microsoft SQL Server 2019

Il Server SQL 2019 include molte nuove caratteristiche, come le seguenti:

- Recupero accelerato del database (ADR) – Riduce il tempo di ripristino dell'arresto anomalo dopo un riavvio o un rollback delle transazioni a esecuzione prolungata.
- Elaborazione intelligente delle query (IQP):
  - Feedback di concessione di memoria in modalità riga – Corregge automaticamente le concessioni eccessive, che altrimenti comporterebbero una perdita di memoria e una riduzione della concorrenza.
  - Modalità batch su rowstore – Consente l'esecuzione in modalità batch per carichi di lavoro analitici senza richiedere indici columnstore.
  - Compilazione differita variabile tabella – Migliora la qualità del piano e le prestazioni complessive per le query che fanno riferimento alle variabili di tabella.
- Prestazioni intelligenti:
  - `OPTIMIZE_FOR_SEQUENTIAL_KEY` opzione di indice – Migliora la velocità effettiva per gli inserimenti ad alta concorrenza negli indici.
  - Miglioramento della scalabilità dei checkpoint indiretti – Aiuta i database con carichi di lavoro DML pesanti.
  - Aggiornamenti PFS (Concurrent Page Free Space) – Consente la gestione come latch condiviso anziché come latch esclusivo.
- Monitoraggio dei miglioramenti
  - `WAIT_ON_SYNC_STATISTICS_REFRESH` tipo di attesa – Mostra il tempo accumulato a livello di istanza impiegato per le operazioni di aggiornamento delle statistiche sincrone.
  - Configurazioni con ambito database – Include `LIGHTWEIGHT_QUERY_PROFILING` e `LAST_QUERY_PLAN_STATS`.
  - Funzioni di gestione dinamica (DMF) – Include `sys.dm_exec_query_plan_stats` e `sys.dm_db_page_info`.
- Avvertenze di troncamento dettagliato – Il messaggio di errore di troncamento dei dati per impostazione predefinita include i nomi delle tabelle e delle colonne e il valore troncato.
- Creazione di indici in linea ripristinabili – In SQL Server 2017, è supportata solo la ricostruzione dell'indice online ripristinabili.

Per l'elenco completo delle caratteristiche di SQL Server 2019, consulta Novità di SQL Server 2019 (15.x) nella documentazione Microsoft.

Per un elenco delle funzionalità non supportate, consulta Caratteristiche non supportate e caratteristiche con supporto limitato.

Funzionalità di Microsoft SQL Server 2017

Il Server SQL 2017 include molte nuove caratteristiche, come le seguenti:

- Elaborazione adattiva delle query
- Correzione automatica del piano (caratteristica di regolazione automatica)
- GraphDB
- Ricostruzione di indici ripristinabili

Per l'elenco completo delle caratteristiche di SQL Server 2017, consulta Novità di SQL Server 2017 nella documentazione Microsoft.

Per un elenco delle funzionalità non supportate, consulta Caratteristiche non supportate e caratteristiche con supporto limitato.

Funzionalità di Microsoft SQL Server 2016

Amazon RDS supporta le seguenti versioni del Server SQL 2016:

- Sempre crittografato
- Supporto JSON
- Analisi operative
- Archiviazione query
- Tabelle globali

Per l'elenco completo delle caratteristiche di SQL Server 2016, consulta Novità di SQL Server 2016 nella documentazione Microsoft.

Funzionalità di Microsoft SQL Server 2014

Oltre alle caratteristiche supportate di SQL Server 2012, Amazon RDS supporta il nuovo ottimizzatore di query disponibile in SQL Server 2014 e anche la funzionalità di durata ritardata.

Per un elenco delle funzionalità non supportate, consulta Caratteristiche non supportate e caratteristiche con supporto limitato.

SQL Server 2014 supporta tutti i parametri di SQL Server 2012 e utilizza gli stessi valori predefiniti. SQL Server 2014 include un nuovo parametro, il checksum del backup predefinito. Per ulteriori informazioni, consulta Come abilitare l'opzione CHECKSUM se le utilità di backup non espongono l'opzione nella documentazione di Microsoft.

Fine del supporto di Microsoft SQL Server 2012 su Amazon RDS

SQL Server 2012 ha raggiunto la fine del supporto su Amazon RDS.

RDS inizierà ad aggiornare tutte le istanze database esistenti che ancora utilizzano SQL Server 2012 all'ultima versione minore di SQL Server 2014. Per ulteriori informazioni, consulta Gestione della versione in Amazon RDS.

Fine del supporto Microsoft SQL Server 2008 R2 su Amazon RDS

SQL Server 2008 R2 ha raggiunto la fine del supporto su Amazon RDS.

RDS inizierà ad aggiornare tutte le istanze database esistenti che ancora utilizzano SQL Server 2008 R2 all'ultima versione minore di SQL Server 2012. Per ulteriori informazioni, consulta Gestione della versione in Amazon RDS.

Cambia il supporto Data Capture per le istanze del database di Microsoft SQL Server

Amazon RDS supporta Change Data Capture (CDC) per le istanze di database che eseguono Microsoft SQL Server. CDC acquisisce le modifiche apportate ai dati nelle tabelle e memorizza i metadati relativi a ogni modifica a cui è possibile accedere successivamente. Per ulteriori informazioni, consulta Change Data Capture nella documentazione di Microsoft.

Amazon RDS supporta CDC per le seguenti edizioni e versioni di SQL Server:

- Microsoft SQL Server Enterprise Edition (tutte le versioni)
- Microsoft SQL Server Standard Edition:
  - 2022
  - 2019
  - 2017

- 2016 versione 13.00.4422.0 SP1 CU2 e successive

Per utilizzare il CDC con il tuo Amazon RDS Istanze di database, prima abilitare o disabilitare il CDC a livello di database utilizzando le procedure di archiviazione fornite da RDS. Successivamente, qualsiasi utente che abbia il ruolo `db_owner` per quel database può utilizzare le procedure di archiviazione originarie di Microsoft per controllare il CDC su quel database. Per ulteriori informazioni, consulta Uso di Change Data Capture.

È possibile utilizzare CDC e AWS Database Migration Service abilitare la replica continua dalle istanze DB di SQL Server.

Caratteristiche non supportate e caratteristiche con supporto limitato

Le seguenti caratteristiche di Microsoft SQL Server non sono supportate su Amazon RDS:

- Backup su Archiviazione BLOB di Microsoft Azure
- Estensione del pool di buffer
- Policy di password personalizzate
- Servizi di Qualità dei Dati
- Log di database
- Snapshot del database (Amazon RDS supporta solo snapshot di istanze database)
- Stored procedure estese, incluso xp_cmdshell
- Supporto FILESTREAM
- Tabelle di file
- Machine Learning ed R Services (richiedono l'accesso al SO per l'installazione)
- Piani di manutenzione
- Prestazioni della raccolta dati
- Gestione basata sulla Policy
- PolyBase
- Replica
- Direttore delle risorse
- Trigger a livello di server

- Endpoint del broker del servizio
- Database elastico
- Proprietà del database TRUSTWORTHY (richiede il ruolo sysadmin)
- Endpoint T-SQL (tutte le operazioni che usano CREATE ENDPOINT (Crea endpoint) non sono disponibili)
- Servizi dei dati WCF

Le seguenti caratteristiche di Microsoft SQL Service hanno un supporto limitato su Amazon RDS:

- Query distribuite/Server collegati. Per ulteriori informazioni, consulta Implement linked servers with Amazon RDS for Microsoft SQL Server.
- Common Runtime Language (CLR). Su RDS per SQL Server 2016 e versioni inferiori, CLR è supportato in modo SAFE e utilizzando solo i bit di assemblaggio. CLR non è supportato su RDS per SQL Server 2017 e versioni successive. Per ulteriori informazioni, consulta Integrazione del Common Runtime Language nella documentazione di Microsoft.

Le seguenti funzionalità non sono supportate su Amazon RDS con SQL Server 2022:

- Sospendi il database per lo snapshot
- Fonte di dati esterna
- Backup e ripristino su storage di oggetti compatibile con S3
- Integrazione con Object Store
- TLS 1.3 e MS-TDS 8.0
- Offload della compressione di backup con QAT
- Servizi di analisi SQL Server (SSAS)
- Mirroring del database con implementazioni Multi-AZ. SQL Server Always On è l'unico metodo supportato con distribuzioni Multi-AZ.

Le implementazioni Multi-AZ utilizzando Microsoft SQL Server Database Mirroring o i gruppi di disponibilità Always On

Amazon RDS supporta le implementazioni Multi-AZ per le istanze database che eseguono Microsoft SQL Server utilizzando i gruppi di disponibilità Always On (AG) o il mirroring del database

(DBM) di SQL Server. Le implementazioni Multi-AZ forniscono alle istanze database maggior disponibilità, longevità dei dati e tolleranza ai guasti. In caso di manutenzione pianificata del database o di interruzione non pianificata del servizio, Amazon RDS esegue automaticamente il failover sulla replica up-to-date secondaria in modo che le operazioni del database possano riprendere rapidamente senza interventi manuali. Le istanze primarie e secondarie usano lo stesso endpoint, il cui indirizzo di rete fisico passa alla replica secondaria passiva come parte del processo di failover. Non è necessario riconfigurare l'applicazione quando si verifica un failover.

Amazon RDS gestisce il failover monitorando attivamente l'implementazione Multi-AZ e avviando un failover quando si verifica un problema con quello primario. Il failover ha luogo solo se le istanze di standby e primarie non sono completamente sincronizzate. Amazon RDS gestisce attivamente l'implementazione Multi-AZ riparando automaticamente le istanze database non funzionanti e ristabilendo la replica sincrona. Non è necessario gestire nulla. Amazon RDS gestisce l'istanza primaria, il testimone e l'istanza di standby al posto tuo. Quando configuri SQL Server Multi-AZ, RDS configura le istanze secondarie passive per tutti i database su un'istanza.

Per ulteriori informazioni, consulta Implementazioni Multi-AZ per Amazon RDS for Microsoft SQL Server.

Uso della crittografia dei dati trasparente per crittografare i dati inattivi

Amazon RDS supporta l'utilizzo di Microsoft SQL Server Transparent Data Encryption (TDE), che esegue in modo trasparente la crittografia dei dati archiviati. Amazon RDS utilizza gruppi di opzioni per abilitare e configurare queste funzionalità. Per ulteriori informazioni sull'opzione TDE, consulta Supporto per Transparent Data Encryption in SQL Server.

Funzioni e procedure archiviate per Amazon RDS for Microsoft SQL Server

Di seguito, puoi trovare un elenco di funzioni e stored procedure Amazon RDS che consentono di automatizzare le attività di SQL Server.

Tipo di attività	Procedura o funzione	Dove viene usata
Attività amministrative	`rds_drop_database`	Rimozione di un database Microsoft SQL Server
	`rds_failover_time`	Determinazione dell'ora dell'ultimo failover
	`rds_modify_db_name`	Ridenominazione di un database Microsoft SQL Server in un'implementazione Multi-AZ
	`rds_read_error_log`	Visualizzazione dei log dell'agente e degli errori
	`rds_set_configuration`	Questa operazione viene utilizzata per impostare varie configurazioni di istanze database: • Change Data Capture per istanze Multi-AZ • Impostazione del periodo di retention dei file di traccia e dei file dump • Compressione dei file di backup
	`rds_set_database_online`	Transizione di un database Microsoft SQL Server da OFFLINE a ONLINE
	`rds_set_system_database_sync_objects`	Attivazione della replica di processo SQL Server Agent
	`rds_fn_get_system_database_`	

Tipo di attività	Procedura o funzione	Dove viene usata
	`sync_objects` `rds_fn_server_object_last_sync_time`	
	`rds_show_configuration`	Per vedere i valori impostati utilizzando `rds_set_configuration` , consulta i seguenti argomenti: • Change Data Capture per istanze Multi-AZ • Impostazione del periodo di retention dei file di traccia e dei file dump
	`rds_shrink_tempdbfile`	Riduzione del database tempdb
Change Data Capture (CDC)	`rds_cdc_disable_db`	Disabilitazione di CDC
	`rds_cdc_enable_db`	Attivazione CDC
Posta elettronica database	`rds_fn_sysmail_allitems`	Visualizzazione di messaggi, log e allegati
	`rds_fn_sysmail_event_log`	Visualizzazione di messaggi, log e allegati

Tipo di attività	Procedura o funzione	Dove viene usata
	`rds_fn_sysmail_mailattachments`	Visualizzazione di messaggi, log e allegati
	`rds_sysmail_control`	Questa operazione viene utilizzata per avviare e arrestare la coda di posta: • Avvio della coda di posta • Arresto della coda di posta
	`rds_sysmail_delete_mailitems_sp`	Eliminazione dei messaggi
Backup nativo e ripristino	`rds_backup_database`	Backup di un database
	`rds_cancel_task`	Annullamento di un'attività
	`rds_finish_restore`	Completamento di un ripristino del database
	`rds_restore_database`	Ripristino di un database
	`rds_restore_log`	Ripristino di un log

Tipo di attività	Procedura o funzione	Dove viene usata
Trasferimento di file Simple Storage Service (Amazon S3)	`rds_delete_from_filesystem`	Eliminazione di file nell'istanza database RDS
	`rds_download_from_s3`	Download di file da un bucket Amazon S3 in un'istanza database SQL Server
	`rds_gather_file_details`	Visualizzazione di file nell'istanza database RDS
	`rds_upload_to_s3`	Aggiornamenti di file da un'istanza database SQL Server in un bucket Amazon S3
Microsoft Distributed Transaction Coordinator (MSDTC)	`rds_msdtc_transaction_tracing`	Utilizzo del tracciamento delle transazioni
Audit in SQL Server	`rds_fn_get_audit_file`	Visualizzazione dei log di audit

Tipo di attività	Procedura o funzione	Dove viene usata
Transparent Data Encryption	`rds_backup_tde_certificate` `rds_drop_tde_certificate` `rds_restore_tde_certificate` `rds_fn_list_user_tde_certificates`	Supporto per Transparent Data Encryption in SQL Server

Tipo di attività	Procedura o funzione	Dove viene usata
Microsoft Business Intelligence (MSBI)	`rds_msbi_task`	Questa operazione viene utilizzata con SQL Server Analysis Services (SSAS): • Distribuzione di progetti SSAS su Amazon RDS • Aggiunta di un utente di dominio come amministratore di database • Backup di un database SSAS • Ripristino di un database SSAS Questa operazione viene utilizzata anche con SQL Server Integration Services (SSIS): • Autorizzazioni amministrative su SSISDB • Distribuzione di un progetto SSIS Questa operazione viene utilizzata anche con SQL Server Reporting Services (SSRS): • Concessione dell'accesso agli utenti del dominio • Revoca delle autorizzazioni a livello di sistema
	`rds_fn_task_status`	Questa operazione mostra lo stato delle attività MSBI: • SSAS: Monitoraggio dello stato di un'attività di distribuzione • SSIS: Monitoraggio dello stato di un'attività di distribuzione • SSRS: Monitoraggio dello stato di un'attività
SSIS	`rds_drop_ssis_database`	Eliminazione del database SSISDB

Tipo di attività	Procedura o funzione	Dove viene usata
	`rds_sqlagent_proxy`	Creazione di un proxy SSIS
SSRS	`rds_drop_ssrs_databases`	Eliminazione dei database SSRS

Fuso orario locale per le istanze di database di Microsoft SQL Server

Il fuso orario di un'istanza database Amazon RDS che esegue Microsoft SQL Server è impostato in modo predefinito. L'impostazione predefinita corrente è Universal Coordinated Time (UTC). Ora puoi invece impostare il fuso orario delle istanze database su un fuso orario locale, per farlo corrispondere a quello delle applicazioni.

Puoi impostare il fuso orario quando si crea prima l'istanza database. Puoi creare la tua istanza DB utilizzando l'azione AWS Management ConsoleCreateDBInstance dell'API Amazon RDS o il comando. AWS CLI create-db-instance

Se l'istanza database fa parte di un'implementazione Multi-AZ (usando il SQL Server DBM o AG), quando si esegue il failover, il fuso orario rimane il fuso orario locale impostato. Per ulteriori informazioni, consulta Le implementazioni Multi-AZ utilizzando Microsoft SQL Server Database Mirroring o i gruppi di disponibilità Always On.

Quando richiedi un point-in-time ripristino, specifichi l'ora in cui eseguire il ripristino. L'ora viene visualizzata nel fuso orario locale. Per ulteriori informazioni, consulta Ripristino a un'ora specifica per un'istanza database.

Le seguenti limitazioni riguardano l'impostazione del fuso orario locale sull'istanza database:

- Non è possibile modificare il fuso orario di un'istanza database esistente del server SQL.
- Non è possibile ripristinare uno snapshot da un'istanza database in un fuso orario a un'istanza database in un fuso orario diverso.

- Consigliamo vivamente di non ripristinare un file di backup da un fuso orario a un fuso orario diverso. Se ripristini un file di backup da un fuso orario in un fuso orario diverso, devi controllare le query e le applicazioni per verificare gli effetti del cambiamento di fuso orario. Per ulteriori informazioni, consulta Importazione ed esportazione di database SQL Server mediante backup e ripristino nativi.

Fusi orari supportati

Puoi impostare il fuso orario locale su uno dei valori elencati nella tabella di seguito.

Fusi orari supportati per Amazon RDS su SQL Server

Time zone (Fuso orario)	Offset temporale standard	Descrizione	Note
Orario standard Afghanistan	(UTC+04:30)	Kabul	Questo fuso orario non osserva l'ora legale.
Orario standard Alaska	(UTC–09:00)	Alaska	
Orario standard delle Isole Aleutine	(UTC–10:00)	Isole Aleutine	
Orario standard Altai	(UTC+07:00)	Barnaul, Gorno-Altaysk	
Orario standard arabo	(UTC+03:00)	Kuwait, Riyad	Questo fuso orario non osserva l'ora legale.
Orario standard Arabia	(UTC+04:00)	Abu Dhabi, Mascate	
Orario standard arabo	(UTC+03:00)	Baghdad	Questo fuso orario non osserva l'ora legale.

Time zone (Fuso orario)	Offset temporale standard	Descrizione	Note
Orario standard Argentina	(UTC–03:00)	Città di Buenos Aires	Questo fuso orario non osserva l'ora legale.
Orario standard di Astrakhan	(UTC+04:00)	Astrakhan, Ulyanovsk	
Orario standard Atlantico	(UTC–04:00)	Orario Atlantico (Canada)	
Orario standard Australia centrale	(UTC+09:30)	Darwin	Questo fuso orario non osserva l'ora legale.
Orario standard Australia centrale	(UTC+ 08:45)	Eucla	
Orario standard Australia orientale	(UTC+10:00)	Canberra, Melbourne, Sydney	
Orario standard dell'Azer baigian	(UTC+04:00)	Baku	
Orario standard delle Azzorre	(UTC–01:00)	Azzorre	
Orario standard di Bahia	(UTC–03:00)	Salvador	
Orario standard del Bangladesh	(UTC+06:00)	Dacca	Questo fuso orario non osserva l'ora legale.
Orario standard Bielorussia	(UTC+03:00)	Minsk	Questo fuso orario non osserva l'ora legale.

Time zone (Fuso orario)	Offset temporale standard	Descrizione	Note
Orario standard di Bougainville	(UTC+11:00)	Isola di Bougainville	
Orario standard Canada centrale	(UTC–06:00)	Saskatchewan	Questo fuso orario non osserva l'ora legale.
Orario standard Capo Verde	(UTC–01:00)	Capo Verde II.	Questo fuso orario non osserva l'ora legale.
Orario standard del Caucaso	(UTC+04:00)	Yerevan	
Cen. Ora standard Australia	(UTC+09:30)	Adelaide	
Orario standard America centrale	(UTC–06:00)	America centrale	Questo fuso orario non osserva l'ora legale.
Orario standard Asia centrale	(UTC+06:00)	Astana	Questo fuso orario non osserva l'ora legale.
Orario standard Brasile centrale	(UTC–04:00)	Cuiaba	
Orario standard Europa centrale	(UTC+01:00)	Belgrado, Bratislava, Budapest, Lubiana, Praga	
Orario standard Europeo centrale	(UTC+01:00)	Sarajevo, Skopje, Varsavia, Zagabria	
Orario standard Pacifico centrale	(UTC+11:00)	Isole Salomone, Nuova Caledonia	Questo fuso orario non osserva l'ora legale.

Time zone (Fuso orario)	Offset temporale standard	Descrizione	Note
Orario standard centrale	(UTC–06:00)	Orario Centrale (Stati Uniti e Canada)	
Orario standard centrale (Messico)	(UTC–06:00)	Guadalajara, Città del Messico, Monterrey	
Orario standard Isole Chatham	(UTC+ 12:45)	Isole Chatham	
Orario standard Cina	(UTC+08:00)	Pechino, Chongqing, Hong Kong, Urumqi	Questo fuso orario non osserva l'ora legale.
Orario standard Cuba	(UTC–05:00)	L'Avana	
Orario standard della Dateline	(UTC–12:00)	Linea di data internazionale Ovest	Questo fuso orario non osserva l'ora legale.
Ora standard Africa orientale	(UTC+03:00)	Nairobi	Questo fuso orario non osserva l'ora legale.
Ora standard Australia orientale	(UTC+10:00)	Brisbane	Questo fuso orario non osserva l'ora legale.
Ora standard Europa orientale	(UTC+02:00)	Chisinau	
Ora standard Sud America orientale	(UTC–03:00)	Brasilia	
Orario standard Isola di Pasqua	(UTC–06:00)	Isola di Pasqua	

Time zone (Fuso orario)	Offset temporale standard	Descrizione	Note
Orario standard orientale	(UTC–05:00)	Orario orientale (Stati Uniti e Canada)	
Orario standard orientale (Messico)	(UTC–05:00)	Chetumal	
Orario standard Egitto	(UTC+02:00)	Il Cairo	
Orario standard Ekaterinburg	(UTC+ 05:00)	Ekaterinburg	
Orario standard Fiji	(UTC+12:00)	Figi	
Orario standard FLE	(UTC+02:00)	Helsinki, Kiev, Riga, Sofia, Tallinn, Vilnius	
Orario standard Georgia	(UTC+04:00)	Tbilisi	Questo fuso orario non osserva l'ora legale.
Orario standard GMT	(UTC)	Dublino, Edimburgo, Lisbona, Londra	Questo fuso orario non è lo stesso di Greenwich Mean Time. Questo fuso orario osserva l'ora legale.
Orario standard Groenlandia	(UTC–03:00)	Groenlandia	
Orario standard Greenwich	(UTC)	Monrovia, Reykjavik	Questo fuso orario non osserva l'ora legale.
Orario standard GTB	(UTC+02:00)	Atene, Bucarest	

Time zone (Fuso orario)	Offset temporale standard	Descrizione	Note
Orario standard di Haiti	(UTC–05:00)	Haiti	
Orario standard Hawaii	(UTC–10:00)	Hawaii	
Orario standard India	(UTC+05:30)	Chennai, Kolkata, Mumbai, Nuova Delhi	Questo fuso orario non osserva l'ora legale.
Orario standard Iran	(UTC+ 03:30)	Teheran	
Orario standard di Israele	(UTC+02:00)	Gerusalemme	
Orario standard Giordania	(UTC+02:00)	Amman	
Orario standard di Kaliningrad	(UTC+02:00)	Kaliningrad	
Orario standard Kamchatka	(UTC+12:00)	Petropavlovsk-Kamchatsky – Vecchio	
Orario standard Corea	(UTC+09:00)	Seoul	Questo fuso orario non osserva l'ora legale.
Orario standard Libia	(UTC+02:00)	Tripoli	
Ora standard Isole Line	(UTC+ 14:00)	Isola di Kiritimati	
Orario standard Lord Howe	(UTC+ 10:30)	Isola di Lord Howe	
Orario standard Magadan	(UTC+11:00)	Magadan	Questo fuso orario non osserva l'ora legale.
Orario standard Magallanes	(UTC–03:00)	Punta Arenas	

Time zone (Fuso orario)	Offset temporale standard	Descrizione	Note
Orario standard delle Marchesi	(UTC–09:30)	Isole Marchesi	
Orario standard delle Mauritius	(UTC+04:00)	Port Louis	Questo fuso orario non osserva l'ora legale.
Orario standard Medio Oriente	(UTC+02:00)	Beirut	
Orario standard di Montevideo	(UTC–03:00)	Montevideo	
Orario standard del Marocco	(UTC+01:00)	Casablanca	
Orario standard di montagna	(UTC–07:00)	Orario di montagna (Stati Uniti e Canada)	
Orario standard di montagna (Messico)	(UTC–07:00)	Chihuahua, La Paz, Mazatlan	
Orario standard del Myanmar	(UTC+ 06:30)	Yangon (Rangoon)	Questo fuso orario non osserva l'ora legale.
Ora standard Asia centrale settentrionale	(UTC+07:00)	Novosibirsk	
Orario standard della Namibia	(UTC+02:00)	Windhoek	
Orario standard del Nepal	(UTC+ 05:45)	Katmandu	Questo fuso orario non osserva l'ora legale.
Orario standard Nuova Zelanda	(UTC+12:00)	Auckland, Wellington	
Orario standard Terranova	(UTC–03:30)	Terranova	

Time zone (Fuso orario)	Offset temporale standard	Descrizione	Note
Orario standard di Norfolk	(UTC+11:00)	Isola di Norfolk	
Orario standard dell'Asia nord-orientale	(UTC+08:00)	Irkutsk	
Orario standard dell'Asia settentrionale	(UTC+07:00)	Krasnoyarsk	
Orario standard Corea del Nord	(UTC+09:00)	Pyongyang	
Orario standard di Omsk	(UTC+06:00)	Omsk	
Orario standard Pacifico SA	(UTC–03:00)	Santiago	
Orario standard Pacifico	(UTC–08:00)	Orario del Pacifico (Stati Uniti e Canada)	
Orario standard Pacifico (Messico)	(UTC–08:00)	Bassa California	
Orario standard del Pakistan	(UTC+ 05:00)	Islamabad, Karachi	Questo fuso orario non osserva l'ora legale.
Orario standard Paraguay	(UTC–04:00)	Asuncion	
Orario standard Romance	(UTC+01:00)	Bruxelles, Copenaghen, Madrid, Parigi	
Russia Fuso orario 10	(UTC+11:00)	Chokurdakh	
Russia Fuso orario 11	(UTC+12:00)	Anadyr, Petropavlovsk-Kamchatsky	

Time zone (Fuso orario)	Offset temporale standard	Descrizione	Note
Russia Fuso orario 3	(UTC+04:00)	Izhevsk, Samara	
Orario standard Russia	(UTC+03:00)	Mosca, San Pietroburgo, Volgograd	Questo fuso orario non osserva l'ora legale.
Orario standard SA orientali	(UTC–03:00)	Cayenna, Fortaleza	Questo fuso orario non osserva l'ora legale.
Orario standard Pacifico SA	(UTC–05:00)	Bogotà, Lima, Quito, Rio Branco	Questo fuso orario non osserva l'ora legale.
Orario standard SA occidentali	(UTC–04:00)	Georgetown, La Paz, Manaus, San Juan	Questo fuso orario non osserva l'ora legale.
Orario standard Saint Pierre	(UTC–03:00)	Saint Pierre e Miquelon	
Orario standard Sakhalin	(UTC+11:00)	Sakhalin	
Orario standard Samoa	(UTC+ 13:00)	Samoa	
Orario standard di Sao Tomé	(UTC+01:00)	São Tomé	
Orario standard di Saratov	(UTC+04:00)	Saratov	
Orario standard Asia sud-orientale	(UTC+07:00)	Bangkok, Hanoi, Giacarta	Questo fuso orario non osserva l'ora legale.
Orario standard Singapore	(UTC+08:00)	Kuala Lumpur, Singapore	Questo fuso orario non osserva l'ora legale.

Time zone (Fuso orario)	Offset temporale standard	Descrizione	Note
Orario standard Africa meridionale	(UTC+02:00)	Harare, Pretoria	Questo fuso orario non osserva l'ora legale.
Ora standard dello Sri Lanka	(UTC+05:30)	Sri Jayawarde nepura	Questo fuso orario non osserva l'ora legale.
Ora standard Sudan	(UTC+02:00)	Khartum	
Orario standard Siria	(UTC+02:00)	Damasco	
Orario standard di Taipei	(UTC+08:00)	Taipei	Questo fuso orario non osserva l'ora legale.
Orario standard della Tasmania	(UTC+10:00)	Hobart	
Orario standard Tocantins	(UTC–03:00)	Araguaina	
Orario standard Tokyo	(UTC+09:00)	Osaka, Sapporo, Tokyo	Questo fuso orario non osserva l'ora legale.
Orario standard di Tomsk	(UTC+07:00)	Tomsk	
Orario standard di Tonga	(UTC+ 13:00)	Nuku'alofa	Questo fuso orario non osserva l'ora legale.
Orario standard di Transbaikal	(UTC+09:00)	Chita	
Orario standard della Turchia	(UTC+03:00)	Istanbul	
Orario standard di Turks e Caicos	(UTC–05:00)	Turks e Caicos	

Time zone (Fuso orario)	Offset temporale standard	Descrizione	Note
Orario standard di Ulaanbaatar	(UTC+08:00)	Ulaanbaatar	Questo fuso orario non osserva l'ora legale.
Orario standard Stati Uniti orientali	(UTC–05:00)	Indiana (Est)	
Orario standard di montagna Stati Uniti	(UTC–07:00)	Arizona	Questo fuso orario non osserva l'ora legale.
UTC	UTC	Tempo coordinato universale	Questo fuso orario non osserva l'ora legale.
UTC–02	(UTC–02:00)	Tempo coordinato universale–02	Questo fuso orario non osserva l'ora legale.
UTC–08	(UTC–08:00)	Tempo coordinato universale–08	
UTC–09	(UTC–09:00)	Tempo coordinato universale–09	
UTC–11	(UTC–11:00)	Tempo coordinato universale–11	Questo fuso orario non osserva l'ora legale.
UTC+12	(UTC+12:00)	Tempo coordinato universale+12	Questo fuso orario non osserva l'ora legale.
UTC+13	(UTC+ 13:00)	Tempo coordinato universale+13	

Time zone (Fuso orario)	Offset temporale standard	Descrizione	Note
Orario standard del Venezuela	(UTC–04:00)	Caracas	Questo fuso orario non osserva l'ora legale.
Orario standard di Vladivostok	(UTC+10:00)	Vladivostok	
Orario standard di Volgograd	(UTC+04:00)	Volgograd	
Ora standard Australia occidentale	(UTC+08:00)	Perth	Questo fuso orario non osserva l'ora legale.
Ora standard Africa centrale occidentale	(UTC+01:00)	Africa centro-oc cidentale	Questo fuso orario non osserva l'ora legale.
Ora standard Europa occidentale	(UTC+01:00)	Amsterdam, Berlino, Berna, Roma, Stoccolma, Vienna	
Ora standard Mongolia occidentale	(UTC+07:00)	Hovd	
Orario standard dell'Asia occidentale	(UTC+ 05:00)	Ashgabat, Tashkent	Questo fuso orario non osserva l'ora legale.
Orario standard della Cisgiordania	(UTC+02:00)	Gaza, Hebron	
Orario standard Pacifico occidentale	(UTC+10:00)	Guam, Port Moresby	Questo fuso orario non osserva l'ora legale.
Orario standard di Yakutsk	(UTC+09:00)	Yakutsk	

Licenza per Microsoft SQL Server su Amazon RDS

Quando configuri un'istanza database Amazon RDS per Microsoft SQL Server, la licenza software è inclusa.

Non dovrai pertanto acquistare separatamente le licenze per SQL Server. AWS ha la licenza per il software del database SQL Server. Il prezzo di Amazon RDS include la licenza software, le risorse hardware sottostanti e le funzionalità di gestione di Amazon RDS.

Amazon RDS supporta le edizioni di Microsoft SQL Server seguenti:

- Enterprise
- Standard
- App
- Express

> **Note**
>
> La licenza per SQL Server Web Edition supporta solo servizi Web, applicazioni Web, siti Web e pagine Web accessibili pubblicamente o tramite Internet. Questo livello di supporto è necessario ai fini della conformità ai diritti di utilizzo di Microsoft. Per ulteriori informazioni, consulta Termini del servizio AWS.

Amazon RDS supporta le implementazioni Multi-AZ per le istanze database che eseguono Microsoft SQL Server utilizzando i gruppi di disponibilità Always On (AG) o il mirroring del database (DBM) di SQL Server. Non sono previsti requisiti di licenza aggiuntivi per le implementazioni Multi-AZ. Per ulteriori informazioni, consulta Implementazioni Multi-AZ per Amazon RDS for Microsoft SQL Server.

Ripristino di istanze database terminate in base alla licenza

Amazon RDS acquisisce snapshot di istanze database terminate in base alla licenza. Se l'istanza è terminata a causa di problemi con la licenza, puoi ripristinarla dalla snapshot in una nuova istanza database, che includerà una licenza.

Per ulteriori informazioni, consulta Ripristino di istanze database terminate in base alla licenza.

Sviluppo e test

A causa dei requisiti di licenza, SQL Server Developer Edition non è supportato su Amazon RDS. Potrai utilizzare l'edizione Express per molte attività di sviluppo e test e per altre esigenze non di produzione. Tuttavia, se sono necessarie le funzionalità complete di un'installazione di livello aziendale di SQL Server per lo sviluppo, è possibile scaricare e installare SQL Server Developer Edition su RDS Custom per SQL Server utilizzando un CEV con BYOM. Per ulteriori informazioni, consulta Preparazione di una CEV utilizzando il modello Porta i tuoi media (BYOM). Per Developer Edition non è necessaria un'infrastruttura dedicata. Tramite il tuo host, potrai inoltre ottenere l'accesso ad altre funzionalità di programmazione non disponibili su Amazon RDS. Per ulteriori informazioni sulla differenza tra le edizioni di SQL Server, vedere Edizioni e funzionalità supportate di SQL Server 2019 nella documentazione Microsoft.

Connessione a un'istanza database che esegua il motore di database di Microsoft SQL Server

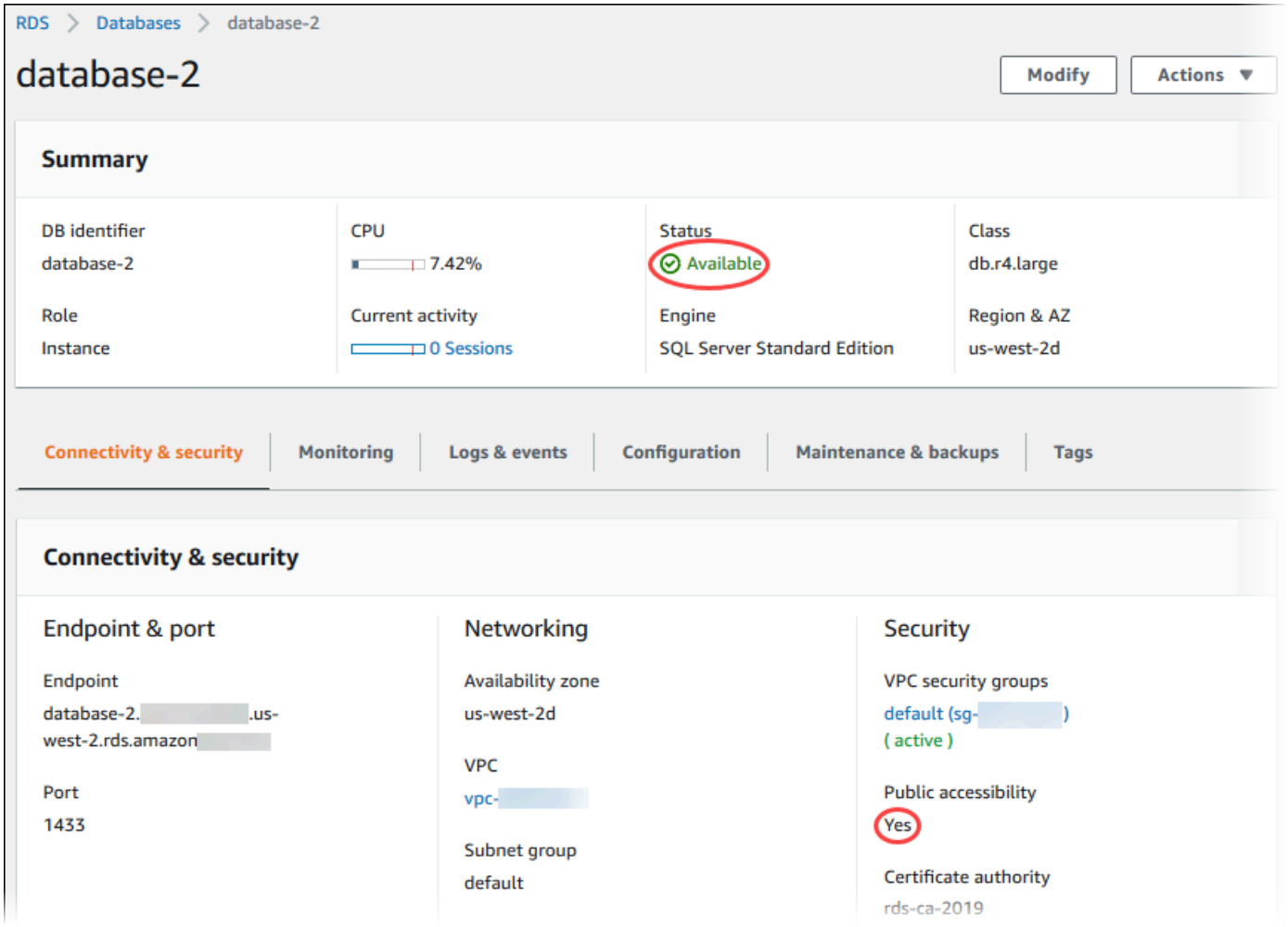
Dopo che Amazon RDS ha effettuato il provisioning della tua istanza database, puoi utilizzare qualsiasi applicazione client SQL standard per connetterti all'istanza database. Questo argomento descrive come connetterti alla tua istanza database utilizzando Microsoft SQL Server Management Studio (SSMS) o SQL Workbench/J.

Per un esempio che illustra il processo di creazione e di connessione a un'istanza database di esempio, consulta Creazione e connessione a un'istanza database Microsoft SQL Server.

Prima di connetterti

Prima di poterti connettere all'istanza database, essa deve essere disponibile e accessibile.

1. Assicurati che lo stato sia `available`. Puoi verificarlo nella pagina dei dettagli della tua istanza in AWS Management Console o utilizzando il describe-db-instances AWS CLI comando.



2. Assicurati che la tua origine possa accedervi. A seconda dello scenario, potrebbe non essere accessibile a livello pubblico. Per ulteriori informazioni, consulta VPC di Amazon VPC e Amazon RDS.
3. Assicurati che le regole in entrata del gruppo di sicurezza VPC consentano l'accesso all'istanza database. Per ulteriori informazioni, consulta Impossibile connettersi all'istanza database di Amazon RDS.

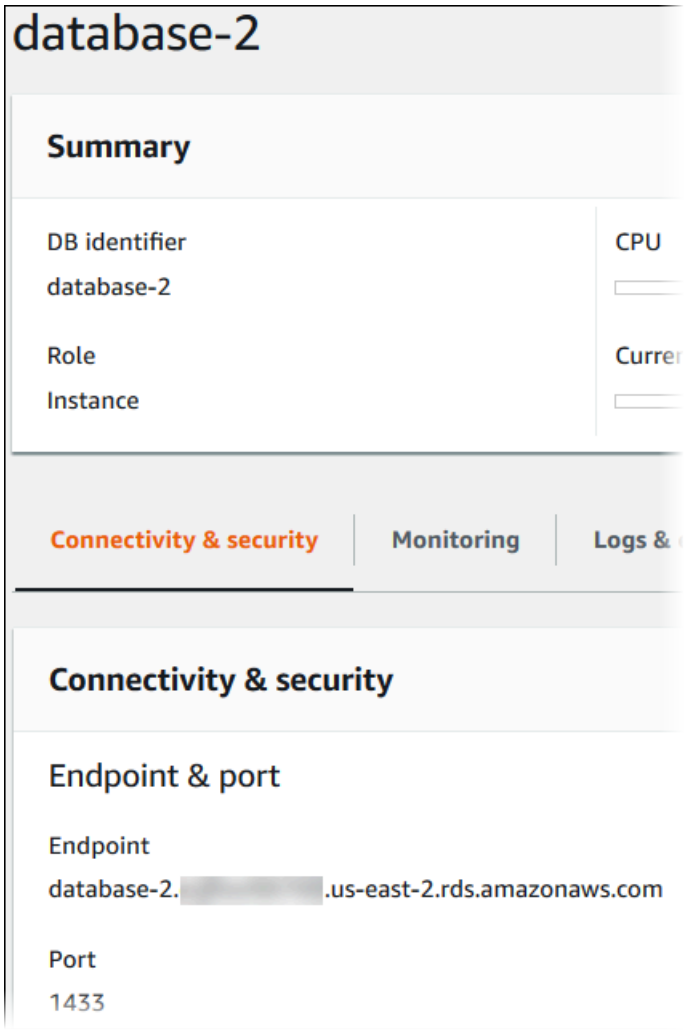
Individuazione dell'endpoint e del numero di porta dell'istanza database

L'endpoint e il numero di porta sono necessari per la connessione all'istanza database.

Per trovare l'endpoint e la porta

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.

2. Nell'angolo in alto a destra della console Amazon RDS, scegli la AWS regione della tua istanza DB.
3. Trovare il nome Domain Name System (DNS) (endpoint) e il numero di porta per l'istanza database:
   a. Aprire la console RDS e selezionare Databases (Database) per visualizzare un elenco delle istanze database.
   b. Scegliere il nome dell'istanza database SQL Server per visualizzarne i dettagli.
   c. Nella scheda Connectivity & security (Connettività e sicurezza), copiare l'endpoint.



   d. Annotare il numero di porta.

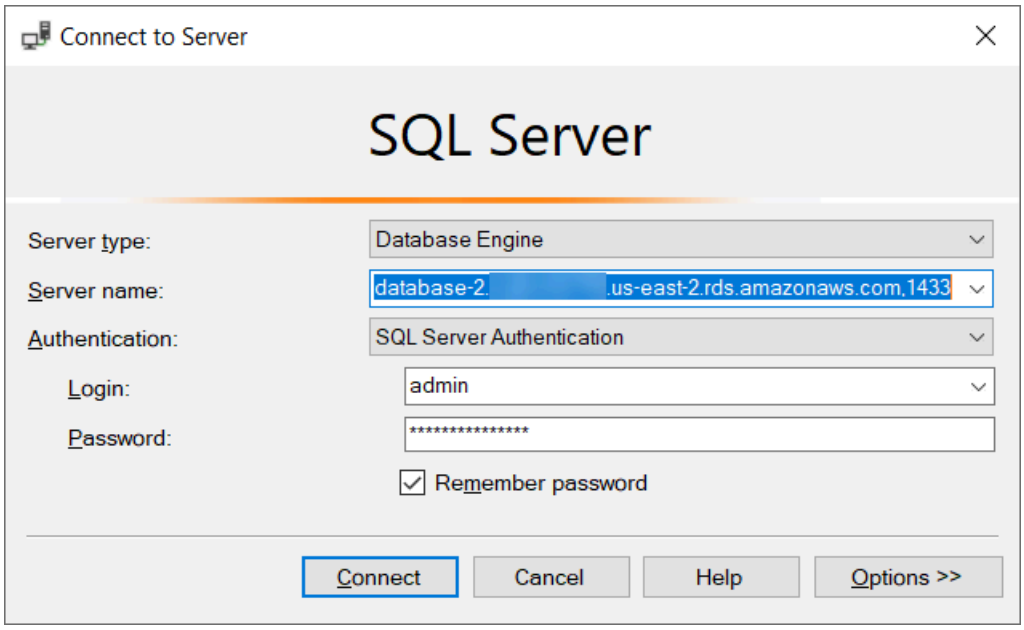
Connessione all'istanza database con Microsoft SQL Server Management Studio

In questa procedura, eseguirai la connessione all'istanza database di esempio utilizzando Microsoft SQL Server Management Studio (SSMS). Per scaricare una versione standalone di questa utilità, consulta Download di SQL Server Management Studio (SSMS) nella documentazione di Microsoft.

Per effettuare la connessione a un'istanza database utilizzando SSMS

1. Avviare SQL Server Management Studio.

   Viene visualizzata la finestra di dialogo Connect to Server (Connettiti al server).



2. Fornire le informazioni per l'istanza database:

   a. In Server type (Tipo di server) scegliere Database Engine (Motore di database).

   b. Per Nome server inserire il nome DNS (endpoint) e il numero di porta dell'istanza database, separati da una virgola.

   > **Important**
   >
   > Sostituire i due punti tra l'endpoint e il numero porta con una virgola.

   L'aspetto del nome server deve essere simile al seguente.

```
database-2.cg034itsfake.us-east-1.rds.amazonaws.com,1433
```

c. In Authentication (Autenticazione) selezionare SQL Server Authentication (Autenticazione SQL Server).

d. Per Login (Accesso) inserire il nome utente master per l'istanza database.

e. Per Password inserire la password per l'istanza database.

3. Scegliere Connetti.

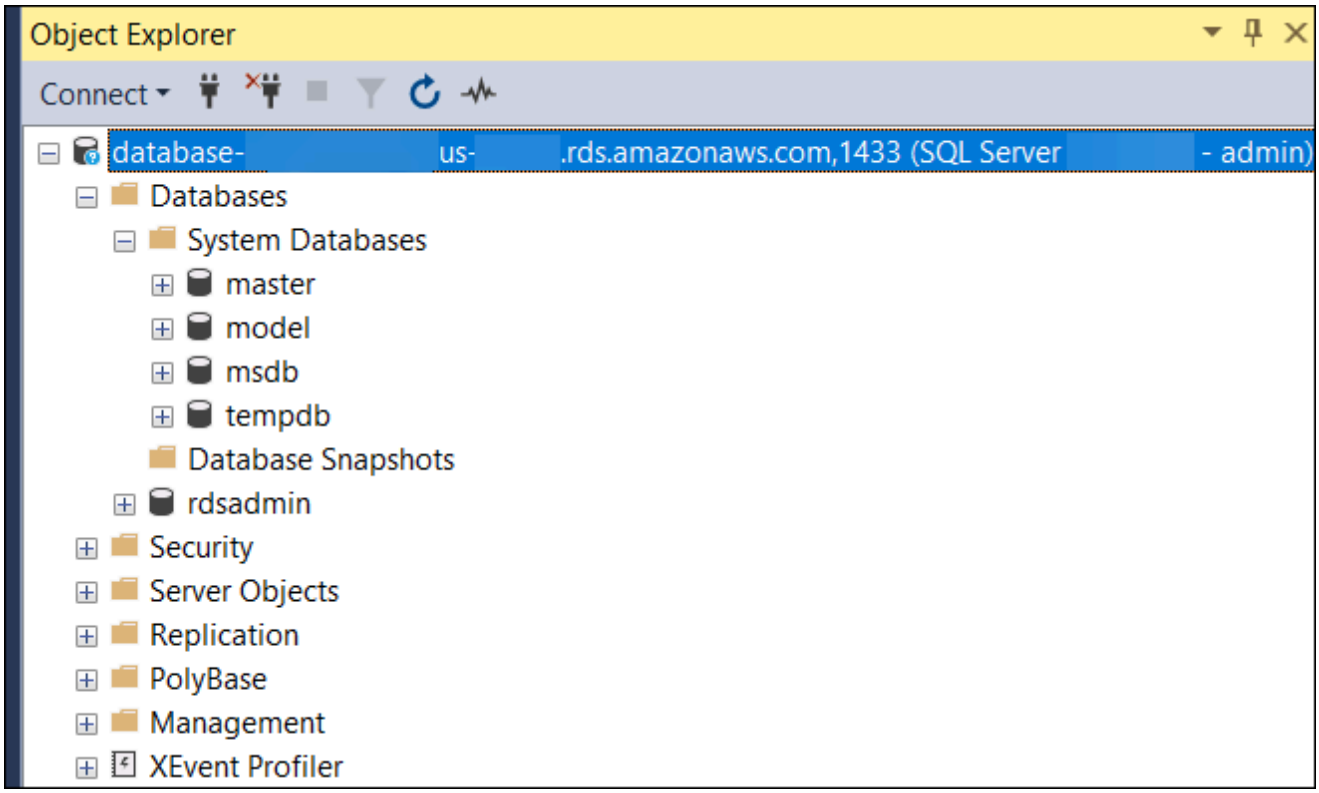
   Dopo qualche secondo, SSMS effettua la connessione all'istanza database.

   In caso di problemi di connessione all'istanza database, consultare Considerazioni relative al gruppo di sicurezza e Risoluzione dei problemi relativi alle connessioni all'istanza database di SQL Server.

4. L'istanza database SQL Server integra i database di sistema standard di SQL Server (`master`, `model`, `msdb` e `tempdb`). Per esaminare i database di sistema, procedere nel modo seguente:

   a. In SSMS, nel menu View (Visualizza), scegliere Object Explorer.

   b. Espandere l'istanza database, espandere Database ed espandere Database di sistema.



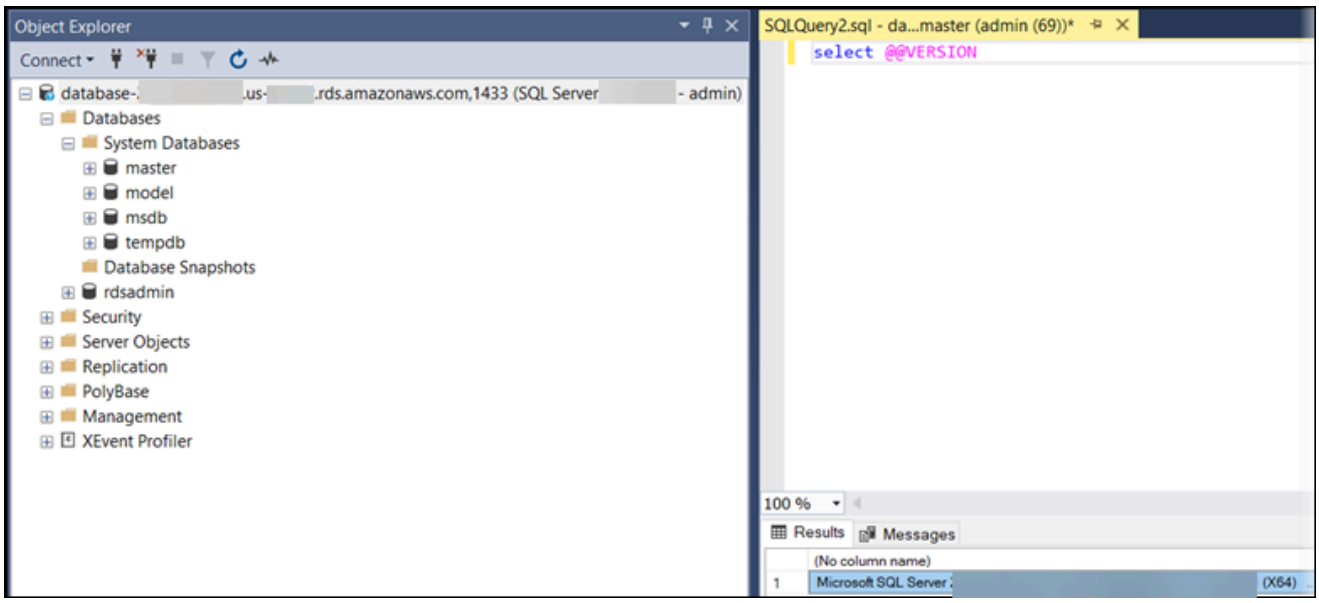
5. L'istanza database di SQL Server viene inoltre fornita con un database denominato `rdsadmin`. Amazon RDS usa questo database per archiviare gli oggetti usati per gestire il database. Il database `rdsadmin` include anche le stored procedure che puoi eseguire per svolgere attività avanzate. Per ulteriori informazioni, consulta Attività DBA frequenti per Microsoft SQL Server.
6. A questo punto, puoi iniziare a creare database personali ed eseguire normalmente query sull'istanza e sui database. Per eseguire una query di test dell'istanza database, utilizzare la seguente procedura:

   a. In SSMS, nel menu File selezionare New (Nuovo), quindi scegliere Query with Current Connection (Query con connessione corrente).

   b. Inserire la query SQL seguente.

   ```
   select @@VERSION
   ```

   c. Eseguire la query. SSMS restituisce la versione di SQL Server dell'istanza database Amazon RDS.



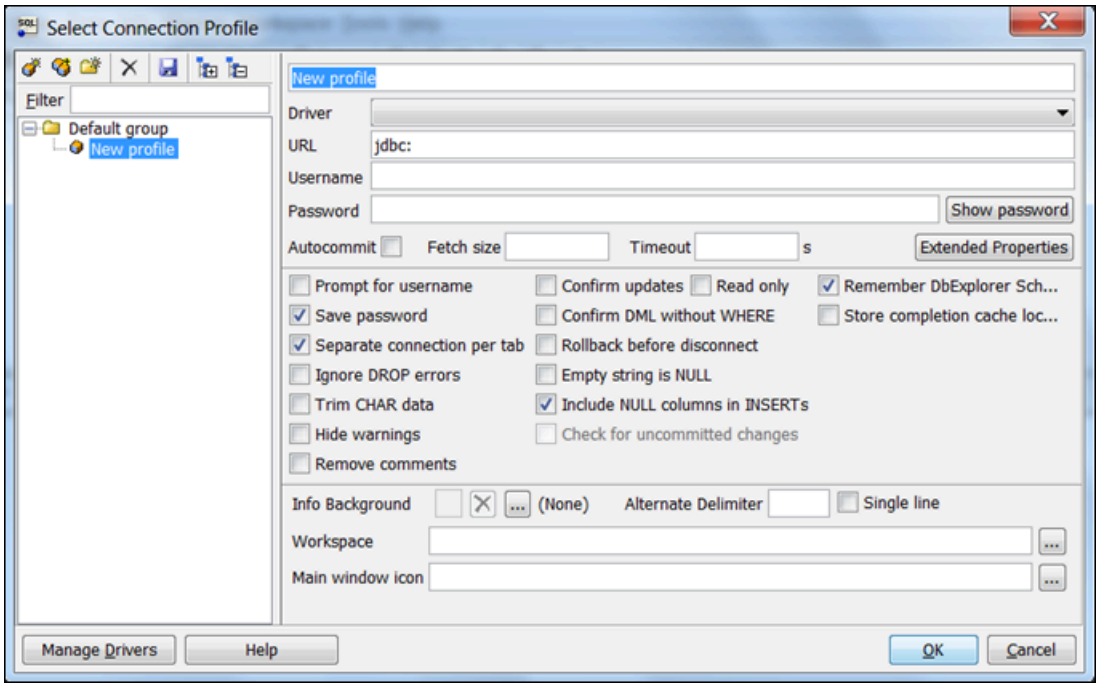
Connessione all'istanza database con SQL Workbench/J

Questo esempio mostra come connettersi a un'istanza database che esegue il motore di database Microsoft SQL Server utilizzando lo strumento di database SQL Workbench/J. Per scaricare SQL Workbench/J, consulta SQL Workbench/J.

SQL Workbench/J utilizza JDBC per connettersi all'istanza database. È inoltre necessario il driver JDBC per SQL Server. Per scaricare questo driver, vedere Microsoft JDBC Driver 6.0 per SQL Server.

Per connettersi a un'istanza database utilizzando SQL Workbench/J

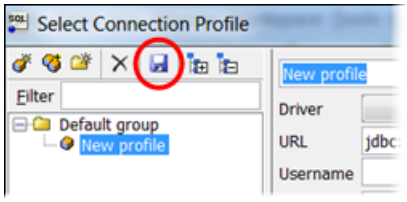
1. Aprire SQL Workbench/J. Viene visualizzata la finestra di dialogo Seleziona profilo di connessione, mostrata di seguito.



2. Nella prima casella nella parte superiore della finestra di dialogo immettere un nome per il profilo.
3. Per Driver scegliere **`SQL JDBC 4.0`**.
4. Per URL inserire **`jdbc:sqlserver://`**, quindi inserire l'endpoint dell'istanza database. Ad esempio, il valore dell'URL potrebbe essere il seguente.

```
jdbc:sqlserver://sqlsvr-pdz.abcd12340.us-west-2.rds.amazonaws.com:1433
```

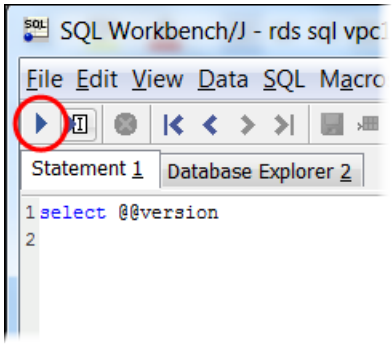
5. Per Username (Nome utente) inserire il nome utente master per l'istanza database.
6. Per Password inserire la password per l'utente master.
7. Scegliere l'icona per eseguire il salvataggio sulla barra degli strumenti della finestra di dialogo, come mostrato di seguito.



8. Scegliere OK. Dopo qualche istante, SQL Workbench/J si connette all'istanza database. In caso di problemi di connessione all'istanza database, consultare Considerazioni relative al gruppo di sicurezza e Risoluzione dei problemi relativi alle connessioni all'istanza database di SQL Server.
9. Nel riquadro delle query inserire la seguente Query SQL.

```
select @@VERSION
```

10. Scegliere l'icona `Execute` sulla barra degli strumenti, come mostrato di seguito.



La query restituisce informazioni sulla versione per l'istanza database simili a quelle mostrate di seguito.

```
Microsoft SQL Server 2017 (RTM-CU22) (KB4577467) - 14.0.3356.20 (X64)
```

Considerazioni relative al gruppo di sicurezza

Per connettersi all'istanza database, è necessario che sia associata a un gruppo di sicurezza. Questo gruppo di sicurezza contiene gli indirizzi IP e la configurazione di rete che si utilizza per accedere all'istanza database. L'istanza database potrebbe essere stata associata a un gruppo di sicurezza appropriato durante la creazione dell'istanza database. Se hai assegnato un gruppo di sicurezza predefinito non configurato quando hai creato la tua istanza database, il firewall dell'istanza database impedisce le connessioni.

In alcuni casi, potrebbe essere necessario creare un nuovo gruppo di sicurezza per rendere possibile l'accesso. Per istruzioni sulla creazione di un nuovo gruppo di sicurezza, consulta Controllo

dell'accesso con i gruppi di sicurezza. Per informazioni su come configurare le regole per il tuo gruppo di sicurezza VPC, consulta Tutorial: Creazione di un Amazon VPC da utilizzare con un'istanza database (solo IPv4).

Dopo aver creato il nuovo gruppo di sicurezza, modifica l'istanza database per associarla al gruppo di sicurezza. Per ulteriori informazioni, consulta Modifica di un'istanza database Amazon RDS.

Puoi aumentare la sicurezza utilizzando la crittografia SSL per proteggere le connessioni alla tua istanza database. Per ulteriori informazioni, consulta Utilizzo di SSL con un'istanza database Microsoft SQL Server.

Risoluzione dei problemi relativi alle connessioni all'istanza database di SQL Server

Nella tabella seguente vengono visualizzati i messaggi di errore che potrebbero verificarsi quando si tenta di connettersi all'istanza database di SQL Server.

Problema	Suggerimenti sulla risoluzione dei problemi
Could not open a connection to SQL Server – Microsoft SQL Server, Error: 53 (Impossibile aprire una connessione a SQL Server – Microsoft SQL Server, errore: 53)	Assicurati di aver specificato correttamente il nome del server. In Server name (Nome server) inserire il nome DNS e il numero di porta dell'istanza database di esempio, separati da una virgola. **Important** Se tra il nome DNS e il numero di porta sono presenti i due punti, cambia i due punti in una virgola. L'aspetto del nome server deve essere simile al seguente. `sample-instance.cg034itsfake.us-east-1.rds.amazonaws.com,1433`
No connection could be made because the target machine actively refused it – Microsoft SQL Server,	È stato possibile raggiungere l'istanza database, ma la connessione è stata rifiutata. Questo problema è in genere causato da una specifica errata della password o del nome utente. Verifica il nome utente e la password e riprova.

Problema	Suggerimenti sulla risoluzione dei problemi
Error: 10061 (Impossibile stabilire una connessione perché rifiutata attivamente dal computer di destinazione – Microsoft SQL Server, errore: 10061)	
A network-related or instance-specific error occurred while establishing a connection to SQL Server. The server was not found or was not accessible... The wait operation timed out – Microsoft SQL Server, Error: 258 (Si è verificato un errore correlato alla rete o specifico dell'istanza durante il tentativo di stabilire una connessione a SQL Server. Impossibile trovare il server o server non accessibile... L'operazione di attesa è scaduta – Microsoft SQL Server, errore: 258	Le regole di accesso applicate dal firewall locale e gli indirizzi IP autorizzati per accedere all'istanza database potrebbero non corrispondere. Il problema è probabilmente correlato alle regole in entrata del gruppo di sicurezza. Per ulteriori informazioni, consulta Sicurezza in Amazon RDS. L'istanza database deve essere accessibile pubblicamente. Per eseguire la connessione dall'esterno del VPC, all'istanza deve essere assegnato un indirizzo IP pubblico.

> **Note**
>
> Per ulteriori informazioni sui problemi di connessione, consulta Impossibile connettersi all'istanza database di Amazon RDS.

Utilizzo di Active Directory con RDS per SQL Server

È possibile aggiungere un'istanza database RDS per SQL Server a un dominio Microsoft Active Directory (AD). Il dominio AD può essere ospitato su AD gestito da AWS all'interno di AWS o su AD gestito dal cliente in una posizione di tua scelta, inclusi i data center aziendali, su AWS EC2 o con altri provider di servizi cloud.

Puoi autenticare gli utenti del dominio utilizzando l'autenticazione NTLM con Active Directory gestito dal cliente. Puoi utilizzare l'autenticazione Kerberos e NTLM con Active Directory gestito da AWS.

Nelle sezioni seguenti, puoi trovare informazioni sull'utilizzo di Active Directory gestito dal cliente e Active Directory gestito da AWS per Microsoft SQL Server su Amazon RDS.

Argomenti

Utilizzo di Active Directory gestito dal cliente con un'istanza database Amazon RDS per SQL Server

Puoi aggiungere le tue istanze DB RDS for SQL Server direttamente al tuo dominio Active Directory (AD) autogestito, indipendentemente da dove è ospitato il tuo AD: nei data center aziendali, su AWS EC2 o con altri provider cloud. Con AD gestito dal cliente, utilizzi l'autenticazione NTLM per controllare direttamente l'autenticazione di utenti e servizi sulle istanze database RDS per SQL Server senza utilizzare domini intermedi e trust tra foreste. Quando gli utenti eseguono l'autenticazione con un'istanza database RDS per SQL Server aggiunta al dominio AD gestito dal cliente, le richieste di autenticazione vengono inoltrate a un dominio AD gestito dal cliente specificato.

Argomenti

Disponibilità di regioni e versioni

Amazon RDS supporta AD gestito dal cliente per SQL Server mediante l'autenticazione NTLM in tutte le Regioni AWS.

Requisiti

Assicurati di soddisfare i seguenti requisiti prima di aggiungere un'istanza database RDS per SQL Server al tuo dominio AD gestito dal cliente.

Argomenti

- Configurazione della connettività di rete
- Configurazione dell'account del servizio di dominio AD

Configurazione di AD on-premise

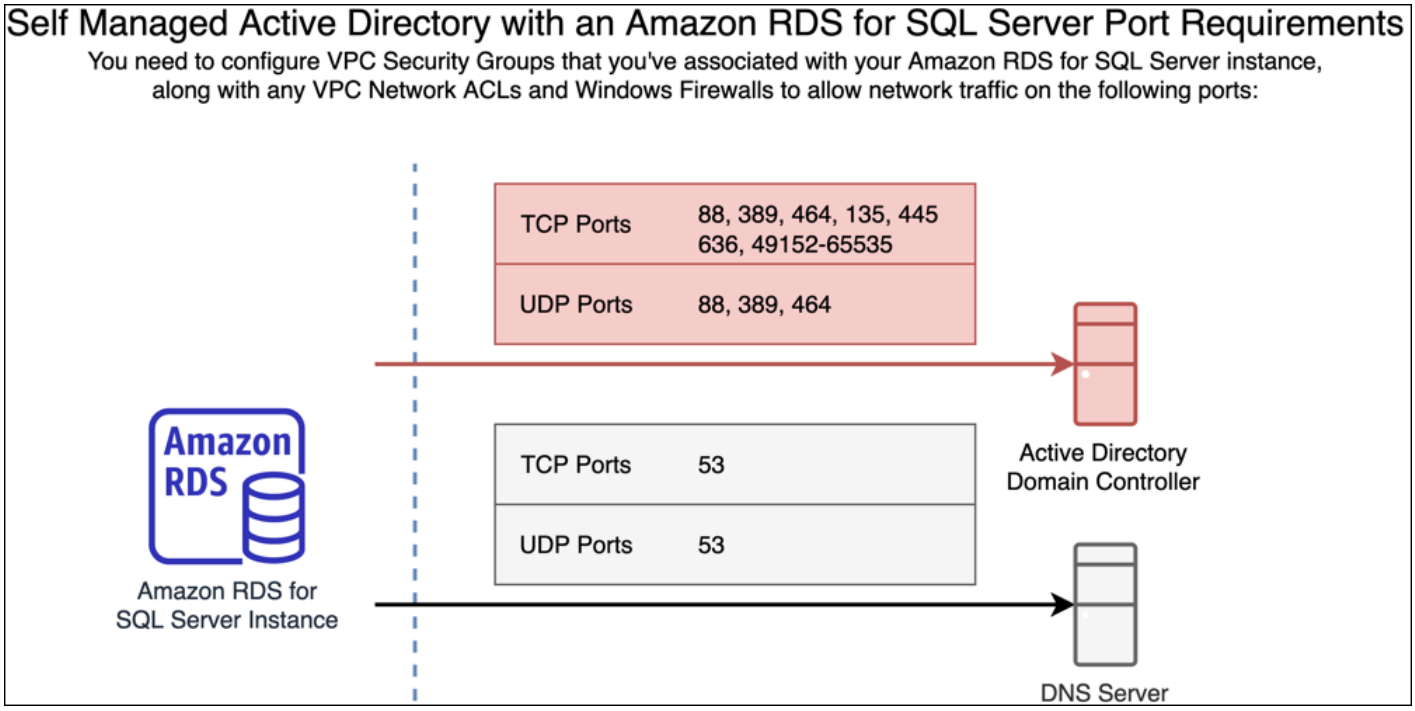
Assicurati di disporre di un dominio Microsoft AD on-premise o gestito dal cliente a cui aggiungere l'istanza Amazon RDS per SQL Server. Il tuo dominio AD on-premise deve avere la seguente configurazione:

- Se sono stati definiti siti Active Directory, assicurati che le sottoreti nel cloud privato virtuale (VPC) associato all'istanza database RDS per SQL Server siano definite nel sito Active Directory. Verifica che non siano presenti conflitti tra le sottoreti del VPC e le sottoreti degli altri siti AD.
- Il tuo controller di dominio AD ha un livello di funzionalità di dominio di Windows Server 2008 R2 o superiore.
- Il nome di dominio AD non può essere nel formato Single Label Domain (SLD). RDS per SQL Server non supporta i domini SLD.
- Il nome di dominio completo (FQDN) per il tuo AD non può superare i 64 caratteri.

Configurazione della connettività di rete

Assicurati di soddisfare le seguenti configurazioni di rete:

- Connettività configurata tra Amazon VPC in cui desideri creare l'istanza database RDS per SQL Server e Active Directory gestito dal cliente. È possibile configurare la connettività utilizzando AWS Direct Connect, AWS VPN, peering VPC o Transit Gateway AWS .
- Per i gruppi di sicurezza VPC, il gruppo di sicurezza predefinito per Amazon VPC predefinito è già aggiunto all'istanza database RDS per SQL Server nella console. Assicurati che il gruppo di sicurezza e le liste di controllo degli accessi (ACL) della rete VPC per le sottoreti in cui stai creando l'istanza database RDS per SQL Server consentano il traffico sulle porte e nelle direzioni mostrate nel diagramma seguente.



Nella tabella seguente è indicato il ruolo di ciascuna porta.

Protocollo	Porte	Ruolo
TCP/UDP	53	Domain Name System (DNS)
TCP/UDP	88	Autenticazione Kerberos
TCP/UDP	464	Modifica/reimpostazione della password
TCP/UDP	389	Lightweight Directory Access Protocol (LDAP)
TCP	135	Distributed Computing Environment/End Point Mapper (DCE/EPMAP)
TCP	445	Condivisione di file SMB di Servizi directory

Protocollo	Porte	Ruolo
TCP	636	Lightweight Directory Access Protocol su TLS/SSL (LDAPS)
TCP	49152 - 65535	Porte effimere per RPC

- In genere, i server DNS del dominio si trovano nei controller di dominio AD. Non è necessario configurare il set di opzioni DHCP nel VPC per utilizzare questa funzionalità. Per ulteriori informazioni, consulta Set di opzioni DHCP nella Guida per l'utente di Amazon VPC.

> **Important**
>
> Se utilizzi le liste di controllo degli accessi (ACL) della rete VPC, devi anche consentire il traffico in uscita sulle porte dinamiche (49152-65535) dall'istanza database RDS per SQL Server. Assicurati che queste regole relative al traffico siano implementate anche nei firewall validi per ciascuno dei controller di dominio AD, per i server DNS e per le istanze database RDS per SQL Server.
>
> Mentre i gruppi di sicurezza VPC richiedono che le porte siano aperte solo nella direzione in cui viene avviato il traffico di rete, la maggior parte dei firewall Windows e delle liste di controllo degli accessi della rete VPC richiedono che le porte siano aperte in entrambe le direzioni.

Configurazione dell'account del servizio di dominio AD

Assicurati di soddisfare i seguenti requisiti per un account del servizio di dominio AD:

- Assicurati di disporre di un account del servizio nel tuo dominio AD gestito dal cliente con autorizzazioni delegate per aggiungere computer al dominio. Un account del servizio di dominio è un account utente nel tuo dominio AD gestito dal cliente a cui è stata delegata l'autorizzazione per eseguire determinate attività.
- All'account del servizio di dominio devono essere delegate le seguenti autorizzazioni nell'unità organizzativa (UO) a cui stai aggiungendo l'istanza database RDS per SQL Server:
  - Capacità convalidata di scrivere sul nome host DNS
  - Capacità convalidata di scrivere sul nome principale del servizio

- Creazione ed eliminazione degli oggetti computer

Rappresentano il set minimo di autorizzazioni necessarie per aggiungere oggetti computer al dominio Active Directory gestito dal cliente. Per ulteriori informazioni, consulta l'argomento relativo agli errori durante il tentativo di aggiungere computer a un dominio nella documentazione di Microsoft Windows Server.

> **Important**
>
> Non spostare gli oggetti computer creati da RDS per SQL Server nell'unità organizzativa dopo la creazione dell'istanza database. Lo spostamento degli oggetti associati creerà una configurazione errata dell'istanza database RDS per SQL Server. Se devi spostare gli oggetti computer creati da Amazon RDS, utilizza l'operazione API RDS ModifyDBInstance per modificare i parametri del dominio in base alla posizione desiderata degli oggetti computer.

Limitazioni

Le seguenti limitazioni si applicano ad AD gestito dal cliente per SQL Server.

- NTLM è l'unico tipo di autenticazione supportato. L'autenticazione Kerberos non è supportata. Se è necessario utilizzare l'autenticazione Kerberos, è possibile utilizzare AWS Managed AD anziché AD autogestito.
- Il servizio Microsoft Distributed Transaction Coordinator (MSDTC) non è supportato, in quanto richiede l'autenticazione Kerberos.
- Le istanze database RDS per SQL Server non utilizzano il server Network Time Protocol (NTP) del dominio AD gestito dal cliente. Utilizzano invece un servizio AWS NTP.
- I server collegati a SQL Server devono utilizzare l'autenticazione SQL per connettersi ad altre istanze database RDS per SQL Server aggiunte al dominio AD gestito dal cliente.
- Le impostazioni Microsoft degli oggetti Criteri di gruppo (GPO) del dominio AD gestito dal cliente non vengono applicate alle istanze database RDS per SQL Server.

Panoramica della configurazione di Active Directory gestito dal cliente

Per configurare un dominio AD gestito dal cliente per un'istanza database RDS per SQL Server, segui i passaggi seguenti, illustrati più dettagliatamente in Configurazione di Active Directory gestito dal cliente:

Nel dominio AD:

- Crea un'unità organizzativa (UO).
- Crea un utente di dominio AD.
- Delega il controllo all'utente di dominio AD.

Dalla AWS Management Console o API:

- Crea una AWS KMS chiave.
- Crea un segreto usando AWS Secrets Manager.
- Crea o modifica un'istanza database RDS per SQL Server e aggiungila al dominio AD gestito dal cliente.

Configurazione di Active Directory gestito dal cliente

Per configurare AD gestito dal cliente, procedi nel seguente modo.

Argomenti

Fase 1: creazione di un'unità organizzativa in AD

> **Important**
>
> È consigliabile creare un'unità organizzativa dedicata e una credenziale di servizio riferita a tale unità organizzativa per qualsiasi AWS account che possiede un'istanza DB RDS per SQL Server aggiunta al dominio AD autogestito. Dedicando un'unità organizzativa e le credenziali di servizio, puoi evitare autorizzazioni in conflitto e seguire il principio del privilegio minimo.

Creazione di un'unità organizzativa in AD

1. Stabilisci una connessione al dominio AD come amministratore del dominio.
2. Apri Utenti e computer di Active Directory e seleziona il dominio in cui desideri creare l'unità organizzativa.
3. Fai clic con il pulsante destro del mouse sul dominio e scegli Nuovo, quindi Unità organizzativa.
4. Inserisci un nome per l'unità operativa.
5. Mantieni selezionata la casella Proteggi il container dall'eliminazione accidentale.
6. Fai clic su OK. La nuova unità organizzativa apparirà sotto il dominio.

Fase 2: creazione un utente di dominio AD in AD

Le credenziali dell'utente del dominio verranno utilizzate per il segreto in AWS Secrets Manager.

Creazione di un utente di dominio AD in AD

1. Apri Utenti e computer di Active Directory e seleziona il dominio e l'unità organizzativa in cui desideri creare l'utente.
2. Fai clic con il pulsante destro del mouse sull'oggetto Utenti, scegli Nuovo, quindi Utente.
3. Immetti il nome, il cognome e il nome di accesso per l'utente. Fai clic su Next (Successivo).
4. Immetti una password per l'utente. Non selezionare "L'utente deve cambiare la password al prossimo accesso". Non selezionare "L'account è disabilitato". Fai clic su Next (Successivo).
5. Fai clic su OK. Il nuovo utente apparirà sotto il dominio.

Fase 3: delega del controllo all'utente AD

Delega del controllo all'utente del dominio AD nel dominio

1. Apri lo snap-in MMC Utenti e computer di Active Directory e seleziona il dominio e l'unità organizzativa in cui desideri creare l'utente.
2. Fai clic con il pulsante destro del mouse sull'unità organizzativa creata in precedenza e scegli Controllo delegato.
3. Nella pagina Delega guidata del controllo, fai clic su Avanti.
4. Nella sezione Utenti o gruppi, fai clic su Aggiungi.
5. Nella sezione Seleziona utenti, computer o gruppi, inserisci l'utente AD creato in precedenza e fai clic su Controlla nomi. Se il controllo degli utenti AD ha esito positivo, fai clic su OK.
6. Nella sezione Utenti o gruppi, conferma che l'utente AD è stato aggiunto e fai clic su Avanti.
7. Nella pagina Operazioni da delegare, seleziona Crea un'operazione personalizzata da delegare, quindi scegli Avanti.
8. Nella sezione Tipo di oggetto Active Directory:
    a. Seleziona Solo i seguenti oggetti contenuti nella cartella.
    b. Seleziona Oggetti del computer.
    c. Seleziona Crea oggetti selezionati in questa cartella.
    d. Seleziona Elimina gli oggetti selezionati in questa cartella e fai clic su Avanti.
9. Nella sezione Autorizzazioni:
    a. Mantieni selezionata l'opzione Generale.
    b. Seleziona Scrittura convalidata in nome host DNS.
    c. Seleziona Scrittura convalidata in nome principale servizio e fai clic su Avanti.
10. Per completare la procedura guidata di delega del controllo, rivedi e conferma le impostazioni e fai clic su Fine.

Fase 4: Creare una AWS KMS chiave

La chiave KMS viene utilizzata per crittografare il tuo AWS segreto.

Per creare una chiave AWS KMS

> **Note**
>
> Per la chiave di crittografia, non utilizzare la chiave KMS AWS predefinita. Assicurati di creare la AWS KMS chiave nello stesso AWS account che contiene l'istanza DB di RDS per SQL Server a cui desideri aggiungere al tuo AD autogestito.

1. Nella AWS KMS console, scegli Crea chiave.
2. In Tipo di chiave, scegli Simmetrica.
3. In Utilizzo delle chiavi, scegli Crittografa e decrittografa.
4. In Advanced options (Opzioni avanzate):
   a. In Origine materiale chiave, scegli KMS.
   b. In Regionalità, scegli Chiave a regione singola e fai clic su Avanti.
5. In Alias, fornisci un nome per la chiave KMS.
6. (Facoltativo) In Descrizione, immetti una descrizione per la chiave KMS.
7. (Facoltativo) In Tag, inserisci un tag come chiave KMS e fai clic su Avanti.
8. In Amministratori delle chiavi, fornisci il nome di un utente IAM e selezionalo.
9. In Eliminazione chiave, mantieni selezionata la casella Consenti agli amministratori delle chiavi di eliminare questa chiave e fai clic su Avanti.
10. In Utenti delle chiavi, fornisci lo stesso utente IAM della fase precedente e selezionalo. Fai clic su Next (Successivo).
11. Riesamina la configurazione.
12. In Policy delle chiavi, includi quanto segue nel campo Dichiarazione associato alla policy:

```
{
    "Sid": "Allow use of the KMS key on behalf of RDS",
    "Effect": "Allow",
    "Principal": {
        "Service": [
            "rds.amazonaws.com"
        ]
    },
    "Action": "kms:Decrypt",
    "Resource": "*"
```

```
}
```

13. Fare clic su Fine.

Passaggio 5: crea un AWS segreto

Per creare un segreto

> **Note**
>
> Assicurati di creare il segreto nello stesso AWS account che contiene l'istanza DB di RDS per SQL Server a cui desideri aggiungere al tuo AD autogestito.

1. In AWS Secrets Manager, scegli Memorizza un nuovo segreto.
2. Per Secret type (Tipo di segreto), scegli Other type of secret (Altro tipo di segreto).
3. In Coppie chiave/valore, aggiungi le due chiavi:
   a. Per la prima chiave, immetti `CUSTOMER_MANAGED_ACTIVE_DIRECTORY_USERNAME`.
   b. Per il valore della prima chiave, immetti il nome dell'utente AD creato nel dominio in una delle fasi precedenti.
   c. Per la seconda chiave, immetti `CUSTOMER_MANAGED_ACTIVE_DIRECTORY_PASSWORD`.
   d. Per il valore della seconda chiave, immetti la password creata per l'utente AD nel dominio.
4. In Chiave di crittografia, inserisci la chiave KMS creata in una delle fasi precedenti e fai clic su Avanti.
5. In Nome del segreto, inserisci un nome descrittivo che semplifichi l'individuazione del segreto in un secondo momento.
6. (Facoltativo) In Descrizione, inserisci una descrizione per il nome del segreto.
7. In Autorizzazioni a livello di risorsa, fai clic su Modifica.
8. Aggiungi la seguente policy alla policy dell'autorizzazione:

> **Note**
>
> Si consiglia di utilizzare le condizione `aws:sourceAccount` e `aws:sourceArn` nella policy per evitare problemi di tipo confused deputy. Usa il tuo Account AWS nome `aws:sourceAccount` e l'`aws:sourceArn`ARN dell'istanza DB di RDS per SQL Server.

Per ulteriori informazioni, consulta Prevenzione del problema "confused deputy" tra servizi.

```
{
    "Version": "2012-10-17",
    "Statement":
    [
        {
            "Effect": "Allow",
            "Principal":
            {
                "Service": "rds.amazonaws.com"
            },
            "Action": "secretsmanager:GetSecretValue",
            "Resource": "*",
            "Condition":
            {
                "StringEquals":
                {
                    "aws:sourceAccount": "123456789012"
                },
                "ArnLike":
                {
                    "aws:sourceArn": "arn:aws:rds:us-west-2:123456789012:db:*"
                }
            }
        }
    ]
}
```

9. Fai clic su Salva, quindi su Avanti.
10. In Configura impostazioni di rotazione, non modificare i valori predefiniti e scegli Avanti.
11. Controlla le impostazioni relative al segreto e fai clic su Archivio.
12. Scegli il segreto creato e copia il valore in ARN secreto. Questa informazione verrà utilizzata nella fase successiva per configurare Active Directory gestito dal cliente.

Fase 6: creazione o modifica di un'istanza database SQL Server

È possibile utilizzare la console, l'interfaccia della linea di comando o l'API RDS per associare un'istanza database RDS per SQL Server a un dominio AD gestito dal cliente. Questa operazione può essere eseguita in uno dei seguenti modi:

- Crea una nuova istanza DB di SQL Server utilizzando la console, il comando create-db-instanceCLI o l'operazione API CreateDBInstance RDS.

  Per istruzioni, consulta Creazione di un'istanza database Amazon RDS.
- Modifica un'istanza DB di SQL Server esistente utilizzando la console, il comando modify-db-instanceCLI o l'operazione API ModifyDBInstance RDS.

  Per istruzioni, consulta Modifica di un'istanza database Amazon RDS.
- Ripristina un'istanza DB di SQL Server da uno snapshot DB utilizzando la console, il comando CLI restore-db-instance-from-db-snapshot o l'operazione API RestoreDB DBSnapshot RDS. InstanceFrom

  Per istruzioni, consulta Ripristino da uno snapshot database.
- Ripristina un'istanza DB di SQL Server point-in-time utilizzando la console, il comando restore-db-instance-to- point-in-time CLI o l'operazione dell'API RestoreDB InstanceToPointInTime RDS.

  Per istruzioni, consulta Ripristino a un'ora specifica per un'istanza database.

Quando si utilizza il AWS CLI, sono necessari i seguenti parametri affinché l'istanza DB possa utilizzare il dominio Active Directory autogestito che è stato creato:

- Per il parametro `--domain-fqdn`, usa il nome di dominio completo (FQDN) del dominio Active Directory gestito dal cliente.
- Per il parametro `--domain-ou`, utilizza l'unità organizzativa creata nel dominio AD gestito dal cliente.
- Per il parametro `--domain-auth-secret-arn`, utilizza il valore riportato nel campo ARN segreto definito in una delle fasi precedenti.
- Per il parametro `--domain-dns-ips`, utilizza gli indirizzi IPv4 primari e secondari dei server DNS per il dominio AD gestito dal cliente. Se non disponi di un indirizzo IP secondario del server DNS, inserisci l'indirizzo IP primario due volte.

I seguenti comandi CLI di esempio mostrano come creare, modificare e rimuovere un'istanza database RDS per SQL Server con un dominio AD gestito dal cliente.

> **Important**
>
> Se modifichi un'istanza database per aggiungerla o rimuoverla da un dominio AD gestito dal cliente, è necessario riavviare l'istanza database affinché la modifica abbia effetto. Puoi scegliere di applicare le modifiche subito o attendere fino alla prossima finestra di manutenzione. La selezione dell'opzione Applica immediatamente causerà tempi di inattività per le istanze database Single-AZ. Un'istanza database Multi-AZ eseguirà un failover prima di completare il riavvio. Per ulteriori informazioni, consulta Impostazione delle modifiche alla pianificazione.

Il seguente comando CLI crea una nuova istanza database RDS per SQL Server e la aggiunge a un dominio AD gestito dal cliente.

Per LinuxmacOS, oUnix:

```
aws rds create-db-instance \
    --db-instance-identifier my-DB-instance \
    --db-instance-class db.m5.xlarge \
    --allocated-storage 50 \
    --engine sqlserver-se \
    --engine-version 15.00.4043.16.v1 \
    --license-model license-included \
    --master-username my-master-username \
    --master-user-password my-master-password \
    --domain-fqdn my_AD_domain.my_AD.my_domain \
    --domain-ou OU=my-AD-test-OU,DC=my-AD-test,DC=my-AD,DC=my-domain \
    --domain-auth-secret-arn "arn:aws:secretsmanager:region:account-number:secret:my-AD-test-secret-123456" \
    --domain-dns-ips "10.11.12.13" "10.11.12.14"
```

Per Windows:

```
aws rds create-db-instance ^
    --db-instance-identifier my-DB-instance ^
    --db-instance-class db.m5.xlarge ^
    --allocated-storage 50 ^
    --engine sqlserver-se ^
```

```
    --engine-version 15.00.4043.16.v1 ^
    --license-model license-included ^
    --master-username my-master-username ^
    --master-user-password my-master-password ^
    --domain-fqdn my-AD-test.my-AD.mydomain ^
    --domain-ou OU=my-AD-test-OU,DC=my-AD-test,DC=my-AD,DC=my-domain ^
    --domain-auth-secret-arn "arn:aws:secretsmanager:region:account-number:secret:my-
AD-test-secret-123456" \ ^
    --domain-dns-ips "10.11.12.13" "10.11.12.14"
```

Il seguente comando CLI modifica un'istanza database RDS per SQL Server esistente in modo che utilizzi un dominio Active Directory gestito dal cliente.

Per LinuxmacOS, oUnix:

```
aws rds modify-db-instance \
    --db-instance-identifier my-DB-instance \
    --domain-fqdn my_AD_domain.my_AD.my_domain \
    --domain-ou OU=my-AD-test-OU,DC=my-AD-test,DC=my-AD,DC=my-domain \
    --domain-auth-secret-arn "arn:aws:secretsmanager:region:account-number:secret:my-
AD-test-secret-123456" \
    --domain-dns-ips "10.11.12.13" "10.11.12.14"
```

Per Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier my-DBinstance ^
    --domain-fqdn my_AD_domain.my_AD.my_domain ^
    --domain-ou OU=my-AD-test-OU,DC=my-AD-test,DC=my-AD,DC=my-domain ^
    --domain-auth-secret-arn "arn:aws:secretsmanager:region:account-number:secret:my-
AD-test-secret-123456" ^
    --domain-dns-ips "10.11.12.13" "10.11.12.14"
```

Il seguente comando CLI rimuove un'istanza database RDS per SQL Server da un dominio Active Directory gestito dal cliente.

Per LinuxmacOS, oUnix:

```
aws rds modify-db-instance \
    --db-instance-identifier my-DB-instance \
    --disable-domain
```

Per Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier my-DB-instance ^
    --disable-domain
```

Fase 7: creazione di accessi SQL Server per l'autenticazione di Windows

Utilizza le credenziali dell'utente master Amazon RDS per eseguire la connessione all'istanza database SQL Server analogamente a quanto avviene con qualsiasi altra istanza database. Poiché l'istanza database viene aggiunta al dominio AD gestito dal cliente, puoi eseguire il provisioning degli account di accesso e degli utenti di SQL Server. Puoi eseguire questa operazione dall'utilità Utenti e gruppi AD nel dominio AD gestito dal cliente. Le autorizzazioni per il database vengono gestite tramite le autorizzazioni standard di SQL Server concesse e revocate in base a questi account di accesso Windows.

Affinché un utente di Active Directory possa eseguire l'autenticazione su SQL Server, deve essere disponibile un accesso Windows SQL Server per l'utente o per un gruppo AD gestito dal cliente di cui l'utente è membro. Il controllo granulare degli accessi viene gestito assegnando o revocando le autorizzazioni per questi login di SQL Server. Un utente AD gestito dal cliente che non dispone di un accesso SQL Server o non appartiene a un gruppo AD gestito dal cliente con tale accesso, non può accedere all'istanza database SQL Server.

È necessaria l'autorizzazione ALTER ANY LOGIN per creare un accesso AD gestito dal cliente per SQL Server. Se non hai ancora creato un accesso con questa autorizzazione, esegui la connessione come utente principale dell'istanza database utilizzando l'autenticazione di SQL Server e quindi crea gli accessi AD gestiti dal cliente per SQL Server nel contesto dell'utente principale.

Esegui il comando DDL (Data Definition Language) seguente per creare un accesso per SQL Server per un utente o un gruppo Active Directory gestito dal cliente.

> **Note**
>
> Specifica utenti o gruppi utilizzando il nome di accesso precedente a Windows 2000 nel formato *my_AD_domain**my_AD_domain_user*. Non puoi utilizzare un UPN (User Principle Name) nel formato *my_AD_domain_user*@*my_AD_domain*.

```
USE [master]
```

```
GO
CREATE LOGIN [my_AD_domain\my_AD_domain_user] FROM WINDOWS WITH DEFAULT_DATABASE =
 [master], DEFAULT_LANGUAGE = [us_english];
GO
```

Per maggiori informazioni, consulta CREATE LOGIN (Transact-SQL) nella documentazione di Microsoft Developer Network.

Gli utenti (persone e applicazioni) del dominio possono ora connettersi all'istanza RDS per SQL Server da un computer client associato al dominio AD gestito dal cliente utilizzando l'autenticazione Windows.

Gestione di un'istanza database in un dominio Active Directory gestito dal cliente

Puoi utilizzare la console o l'API Amazon RDS per gestire la tua istanza DB e la sua relazione con il tuo dominio AD autogestito. AWS CLI Ad esempio, puoi spostare l'istanza database all'interno, al di fuori o tra domini.

Ad esempio, puoi utilizzare l'API Amazon RDS per effettuare quanto segue:

- Per tentare nuovamente l'accesso a un dominio gestito dal cliente a causa di un'iscrizione non riuscita, utilizza l'operazione API ModifyDBInstance e specifica lo stesso set di parametri:
  - `--domain-fqdn`
  - `--domain-dns-ips`
  - `--domain-ou`
  - `--domain-auth-secret-arn`
- Per rimuovere un'istanza database da un dominio gestito dal cliente, utilizza l'operazione API `ModifyDBInstance` e specifica `--disable-domain` come parametro del dominio.
- Per spostare un'istanza database da un dominio gestito dal cliente a un altro, utilizza l'operazione API `ModifyDBInstance` e specifica i parametri di dominio per il nuovo.
  - `--domain-fqdn`
  - `--domain-dns-ips`
  - `--domain-ou`
  - `--domain-auth-secret-arn`
- Per elencare l'iscrizione al dominio AD gestito dal cliente per ciascuna istanza database, utilizza l'operazione API DescribeDBInstances.

Informazioni sull'iscrizione al dominio Active Directory gestito dal cliente

Quando l'istanza database viene creata o modificata, diventa membro del dominio AD gestito dal cliente. La AWS console indica lo stato dell'appartenenza al dominio Active Directory autogestito per l'istanza DB. Lo stato dell'istanza di database può essere uno dei seguenti:

- Collegato: l'istanza è un membro del dominio AD.
- Collegamento in corso: l'istanza sta diventando un membro del dominio.
- pending-join (associazione in sospeso) – L'associazione dell'istanza è in sospeso.
- pending-maintenance-join— AWS tenterà di rendere l'istanza un membro del dominio AD durante la successiva finestra di manutenzione programmata.
- In attesa di rimozione: la rimozione dell'istanza dal dominio AD è in sospeso.
- pending-maintenance-removal— AWS tenterà di rimuovere l'istanza dal dominio AD durante la successiva finestra di manutenzione programmata.
- Non riuscito: un problema di configurazione ha impedito il collegamento dell'istanza al dominio AD. Verifica e correggi la configurazione prima di eseguire nuovamente il comando di modifica dell'istanza.
- Rimozione in corso: è in corso la rimozione dell'istanza dal dominio AD gestito dal cliente.

Una richiesta di collegamento a un dominio AD gestito dal cliente potrebbe non riuscire a causa di un problema di connettività di rete. Ad esempio, è possibile che venga creata un'istanza database o modificata un'istanza esistente senza però che questa diventi un membro di un dominio AD gestito dal cliente. In questo caso, devi emettere nuovamente il comando per creare o modificare l'istanza database o modificare l'istanza appena creata per aggiungerla al dominio AD gestito dal cliente.

Risoluzione dei problemi di Active Directory gestito dal cliente

Di seguito sono riportati i problemi che potresti riscontrare quando configuri o modifichi AD gestito dal cliente.

Codice di errore	Descrizione	Cause comuni	Suggerimenti sulla risoluzione dei problemi
Errore 2/0x2	Il sistema non trova il file	Il formato o la posizione dell'unità organizzativa (UO) specificati con il	Rivedi il parametro –`domain-ou` . Assicurati che l'account del servizio

Codice di errore	Descrizione	Cause comuni	Suggerimenti sulla risoluzione dei problemi
	specificato.	parametro –domain-ou non è valido. L'account del servizio di dominio specificato tramite AWS Secrets Manager non dispone delle autorizzazioni necessarie per accedere all'unità organizzativa.	di dominio disponga delle autorizzazioni corrette per l'unità organizzativa. Per ulteriori informazioni, consulta Configurazione dell'account del servizio di dominio AD.
Errore 5/0x5	Accesso negato.	Nel dominio esistono già autorizzazioni non configurate correttamente per l'account del servizio di dominio o per l'account del computer.	Controlla le autorizzazioni dell'account del servizio di dominio nel dominio e verifica che l'account del computer RDS non sia duplicato nel dominio. È possibile verificare il nome dell'account del computer RDS eseguendo SELECT @@SERVERNAME sull'istanza database RDS per SQL Server. Se utilizzi l'opzione Multi-AZ, prova a riavviare con il failover, quindi verifica nuovamente l'account del computer RDS. Per ulteriori informazioni, consulta Riavvio di un'istanza database.

Codice di errore	Descrizione	Cause comuni	Suggerimenti sulla risoluzione dei problemi
Errore 87/0x57	Il parametro non è corretto.	L'account del servizio di dominio specificato tramite AWS Secrets Manager non dispone delle autorizzazioni corrette. È anche possibile che il profilo utente sia danneggiato.	Verifica i requisiti per l'account del servizio di dominio. Per ulteriori informazioni, consulta Configurazione dell'account del servizio di dominio AD.
Errore 234/0xEA	L'unità organizzativa (UO) specificata non esiste.	L'unità organizzativa specificata con il parametro `–domain-ou` non esiste nel dominio AD gestito dal cliente.	Rivedi il parametro `–domain-ou` e assicurati che l'unità organizzativa specificata esista in AD gestito dal cliente.
Errore 1326/0x52E	Il nome utente o la password sono errati.	Le credenziali dell'account del servizio di dominio fornite in AWS Secrets Manager contengono un nome utente sconosciuto o una password errata. L'account di dominio può anche essere disabilitato in AD gestito dal cliente.	Assicurati che le credenziali fornite in AWS Secrets Manager siano corrette e che l'account di dominio sia abilitato nella tua Active Directory autogestita.

Codice di errore	Descrizione	Cause comuni	Suggerimenti sulla risoluzione dei problemi
Errore 1355/0x54B	Il nome di dominio specificato non esiste o non può essere contattato.	Il dominio è inattivo, il set di IP DNS specificato non è raggiungibile o il nome di dominio completo specificato non è raggiungibile.	Controlla i parametri `–domain-dns-ips` e `–domain-fqdn` per assicurarti che siano corretti. Verifica la configurazione di rete dell'istanza database RDS per SQL Server e assicurati che il dominio AD gestito dal cliente sia raggiungibile. Per ulteriori informazioni, consulta Configurazione della connettività di rete.
Errore 1722/0x6BA	Il server RPC non è disponibile.	Si è verificato un problema durante la connessione al servizio RPC del dominio AD. Questo errore potrebbe essere causato da un problema di rete.	Verifica che il servizio RPC sia in esecuzione sui controller di dominio e che le porte TCP `135` e `49152-65535` siano raggiungibili sul dominio dall'istanza database RDS per SQL Server.
Errore 2224/0x8B0	L'account utente esiste già.	L'account del computer che sta tentando di essere aggiunto al dominio AD gestito dal cliente esiste già.	Identifica l'account del computer eseguendo `SELECT @@SERVERNAME` sull'istanza database RDS per SQL Server, quindi rimuovilo con attenzione da dominio AD gestito dal cliente.

Codice di errore	Descrizione	Cause comuni	Suggerimenti sulla risoluzione dei problemi
Errore 2242/0x8c2	La password di questo utente è scaduta.	La password per l'account del servizio di dominio specificato tramite AWS Secrets Manager è scaduta.	Aggiorna la password per l'account del servizio di dominio utilizzato per aggiungere l'istanza database RDS per SQL Server al dominio AD gestito dal cliente.

Ripristino e aggiunta di un'istanza di database SQL Server a un dominio Active Directory gestito dal cliente

È possibile ripristinare uno snapshot DB o eseguire il point-in-time ripristino (PITR) per un'istanza DB di SQL Server e quindi aggiungerla a un dominio Active Directory autogestito. Dopo aver ripristinato l'istanza database, modificala utilizzando il processo illustrato in Fase 6: creazione o modifica di un'istanza database SQL Server per aggiungere l'istanza a un dominio Active Directory gestito dal cliente.

Utilizzo di Active Directory gestito da AWS con RDS per SQL Server

Puoi utilizzare AWS Managed Microsoft AD per autenticare gli utenti con l'autenticazione di Windows quando si connettono all'istanza database RDS per SQL Server. L'istanza database funziona con AWS Directory Service for Microsoft Active Directory, anche noto come AWS Managed Microsoft AD, per permettere l'autenticazione di Windows. Quando gli utenti eseguono l'autenticazione su un'istanza di database di SQL Server unita al dominio trusting, le richieste di autenticazione vengono inviate alla directory di dominio create con AWS Directory Service.

Disponibilità di regioni e versioni

RDS supporta l'utilizzo solo di AWS Managed Microsoft AD per l'autenticazione Windows. RDS non supporta l'utilizzo di AD Connector. Per ulteriori informazioni, consulta gli argomenti seguenti:

- Policy di compatibilità delle applicazioni per AWS Managed Microsoft AD
- Policy di compatibilità delle applicazioni per AD connector

Per ulteriori informazioni sulla disponibilità di versioni e regioni, consulta Autenticazione Kerberos con Amazon RDS per SQL Server.

Panoramica sulla configurazione dell'autenticazione di Windows

Amazon RDS utilizza la modalità mista per l'autenticazione Windows. In base a questo approccio, l'utente master (il nome e la password utilizzati per creare l'istanza di database SQL Server) utilizza l'autenticazione SQL. Poiché l'account utente master dispone di credenziali privilegiate, è necessario limitare l'accesso a tale account.

Per ottenere l'autenticazione di Windows utilizzando un Microsoft Active Directory in locale o autogestito, crea un trust tra foreste. La fiducia può essere a senso unico o bidirezionale. Per ulteriori informazioni sulla configurazione di trust tra foreste tramite AWS Directory Service, consulta Quando creare una relazione di trust nella Guida di amministrazione di AWS Directory Service.

Per configurare l'autenticazione di Windows per un'istanza database SQL Server, esegui la procedura riportata di seguito, illustrata in maggiore dettaglio in Configurazione dell'autenticazione di Windows per le istanze di database di SQL Server:

1. Utilizza AWS Managed Microsoft AD, dalla AWS Management Console o dall'API di AWS Directory Service, per creare una directory AWS Managed Microsoft AD.

2. Se utilizzi AWS CLI o l'API di Amazon RDS per creare l'istanza database di SQL Server, crea un ruolo AWS Identity and Access Management (IAM). Questo ruolo utilizza la policy IAM gestita `AmazonRDSDirectoryServiceAccess` e consente a Amazon RDS di effettuare chiamate alla directory. Se utilizzi la console per creare l'istanza database di SQL Server, AWS crea automaticamente il ruolo IAM.

   Affinché il ruolo possa permettere l'accesso, l'endpoint AWS Security Token Service (AWS STS) deve essere attivato nella regione AWS per l'account AWS. Gli endpoint AWS STS sono attivi per impostazione predefinita in tutte le regioni AWS e puoi utilizzarli senza ulteriori interventi. Per ulteriori informazioni, consulta la sezione Gestione di AWS STS in una Regione AWS nella Guida per l'utente di IAM.
3. Crea e configura utenti e gruppi nella directory AWS Managed Microsoft AD utilizzando gli strumenti di Microsoft Active Directory. Per ulteriori informazioni sulla creazione di utenti in Active Directory, consulta Gestione di utenti e gruppi in AWS Managed Microsoft AD nella Guida all'amministrazione di AWS Directory Service.
4. Se prevedi di individuare la directory e l'istanza database in VPC diversi, abilita il traffico tra VPC.
5. Utilizza Amazon RDS per creare una nuova istanza database di SQL Server dalla console, dalla AWS CLI o dall'API Amazon RDS. Nella richiesta di creazione, fornisci l'identificatore di dominio ("`d-*`") generato durante la creazione della directory e il nome del ruolo creato. Puoi inoltre modificare un'istanza database di SQL Server per utilizzare l'autenticazione di Windows impostando i parametri dominio e ruolo IAM per l'istanza database.
6. Utilizza le credenziali dell'utente master Amazon RDS per eseguire la connessione all'istanza database di SQL Server analogamente a quanto avviene con qualsiasi altra istanza database. Poiché l'istanza database è aggiunta al dominio AWS Managed Microsoft AD, puoi effettuare il provisioning di account di accesso e utenti di SQL Server da utenti e gruppi di Active Directory nel loro dominio (noti come account di accesso "Windows" di SQL Server) Le autorizzazioni per il database vengono gestite tramite le autorizzazioni standard di SQL Server concesse e revocate in base a questi account di accesso Windows.

Creazione di un endpoint per l'autenticazione Kerberos

L'autenticazione basata su Kerberos richiede che l'endpoint includa il nome host specificato dal cliente, un punto e quindi il nome di dominio completo (FQDN). Il seguente esempio illustra un endpoint che può essere utilizzato con l'autenticazione basata su Kerberos. In questo caso, il nome host dell'istanza database di SQL Server è `ad-test` e il nome di dominio è `corp-ad.company.com`:

```
ad-test.corp-ad.company.com
```

Per accertarti che la connessione utilizzi Kerberos, esegui la seguente query:

```
SELECT net_transport, auth_scheme
  FROM sys.dm_exec_connections
 WHERE session_id = @@SPID;
```

Configurazione dell'autenticazione di Windows per le istanze di database di SQL Server

È possibile usare AWS Directory Service for Microsoft Active Directory, noto anche come AWS Managed Microsoft AD, per configurare l'autenticazione di Windows per un'istanza database di SQL Server. Per configurare l'autenticazione di Windows, attieniti alla seguente procedura.

Fase 1: creazione di una directory utilizzando AWS Directory Service for Microsoft Active Directory

AWS Directory Service crea una Microsoft Active Directory completamente gestita in AWS Cloud. Quando crei una directory AWS Managed Microsoft AD, AWS Directory Service crea automaticamente due controller di dominio e i server Domain Name Service (DNS). I server di directory vengono creati in due sottoreti in due diverse zone di disponibilità all'interno di un VPC. Questa ridondanza assicura che la directory rimanga accessibile anche se si verifica un errore.

Quando crei una directory AWS Managed Microsoft AD, AWS Directory Service esegue le seguenti operazioni:

- Configura una Microsoft Active Directory nel VPC.
- Crea un account amministratore della directory con nome utente Admin e la password specificata. Puoi utilizzare questo account per gestire le directory.

  > **Note**
  >
  > Assicurati di salvare la password. AWS Directory Service non memorizza questa password, che quindi non può essere recuperata o reimpostata.

- Crea un gruppo di sicurezza per i controller della directory.

Quando avvii un AWS Directory Service for Microsoft Active Directory, AWS crea un'unità organizzativa (OU) che contiene tutti gli oggetti della directory. Questa unità organizzativa, che ha

lo stesso nome NetBIOS che hai digitato al momento della creazione della directory, si trova nella radice del dominio. La radice del dominio è di proprietà di ed è gestita da AWS.

L'account admin creato con la directory AWS Managed Microsoft AD dispone delle autorizzazioni per le attività amministrative più comuni per l'unità organizzativa:

- Creazione, aggiornamento o eliminazione di utenti, gruppi e computer.
- Aggiunta di risorse al tuo dominio, come file o server di stampa, quindi assegnazione delle autorizzazioni per tali risorse a utenti e gruppi dell'UO;
- creazione di UO aggiuntive e container;
- delega dell'autorità;
- creazione e collegamento policy di gruppo;
- ripristino degli oggetti eliminati dal cestino riciclaggio di Active Directory;
- Esegui PowerShell moduli Windows AD e DNS sul servizio Web Active Directory.

L'account admin dispone inoltre dei diritti per eseguire le seguenti attività estese a tutto il dominio:

- gestione delle configurazioni DNS (aggiunta, eliminazione o aggiornamento di record, zone e server d'inoltro);
- visualizzazione di log di eventi DNS;
- visualizzazione di log di eventi di sicurezza.

Per creare una directory con AWS Managed Microsoft AD

1. Nel riquadro di navigazione della console AWS Directory Service, scegliere Directories (Directory), quindi selezionare Set up directory (Configurazione della directory).
2. Scegliere AWS Managed Microsoft AD. Attualmente questa è la sola opzione supportata per l'uso con Amazon RDS.
3. Seleziona Avanti.
4. Nella pagina Enter directory information (Inserisci le informazioni sulla directory) inserisci le seguenti informazioni:

   Edizione

   Scegliere l'edizione più adatta alle proprie esigenze.

Nome DNS directory

Il nome completo della directory, ad esempio `corp.example.com`. I nomi più lunghi di 47 caratteri non sono supportati da SQL Server.

Nome NetBIOS della directory

Nome breve opzionale della directory, ad esempio `CORP`.

Descrizione della directory

Descrizione opzionale della directory.

Password amministratore

La password dell'amministratore della directory. Con il processo di creazione della directory viene generato un account amministratore con il nome utente Admin e questa password.

La password dell'amministratore della directory non può includere il termine `admin`. La password distingue tra maiuscole e minuscole e la lunghezza deve essere compresa tra 8 e 64 caratteri. Deve anche contenere un carattere di almeno tre delle seguenti quattro categorie:

- Lettere minuscole (a-z)
- Lettere maiuscole (A-Z)
- Numeri (0-9)
- Caratteri non alfanumerici (~!@#$%^&*_-+=`|\(){}[]:;"'<>,.?/)

Confirm password (Conferma password)

Digitare di nuovo la password dell'amministratore.

5. Seleziona Avanti.
6. Nella pagina Choose VPC and subnets (Scegli VPC e sottoreti) fornire le seguenti informazioni:

VPC

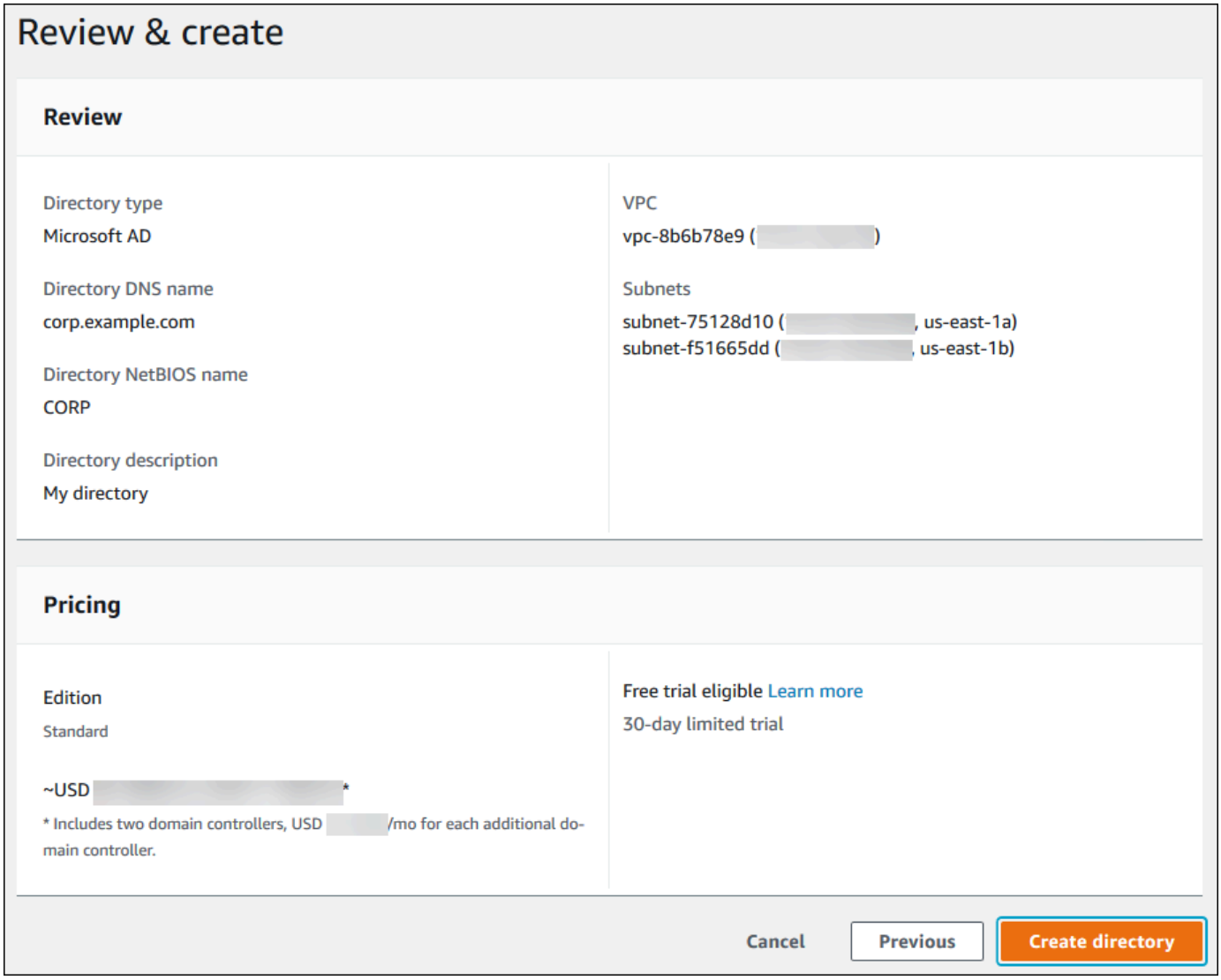
Scegliere il VPC per la directory.

> **Note**
>
> Puoi individuare la directory e l'istanza database in VPC diversi, ma in tal caso, assicurati di abilitare il traffico tra VPC. Per ulteriori informazioni, consulta Fase 4: abilitazione del traffico tra VPC tra la directory e l'istanza database.

Sottoreti

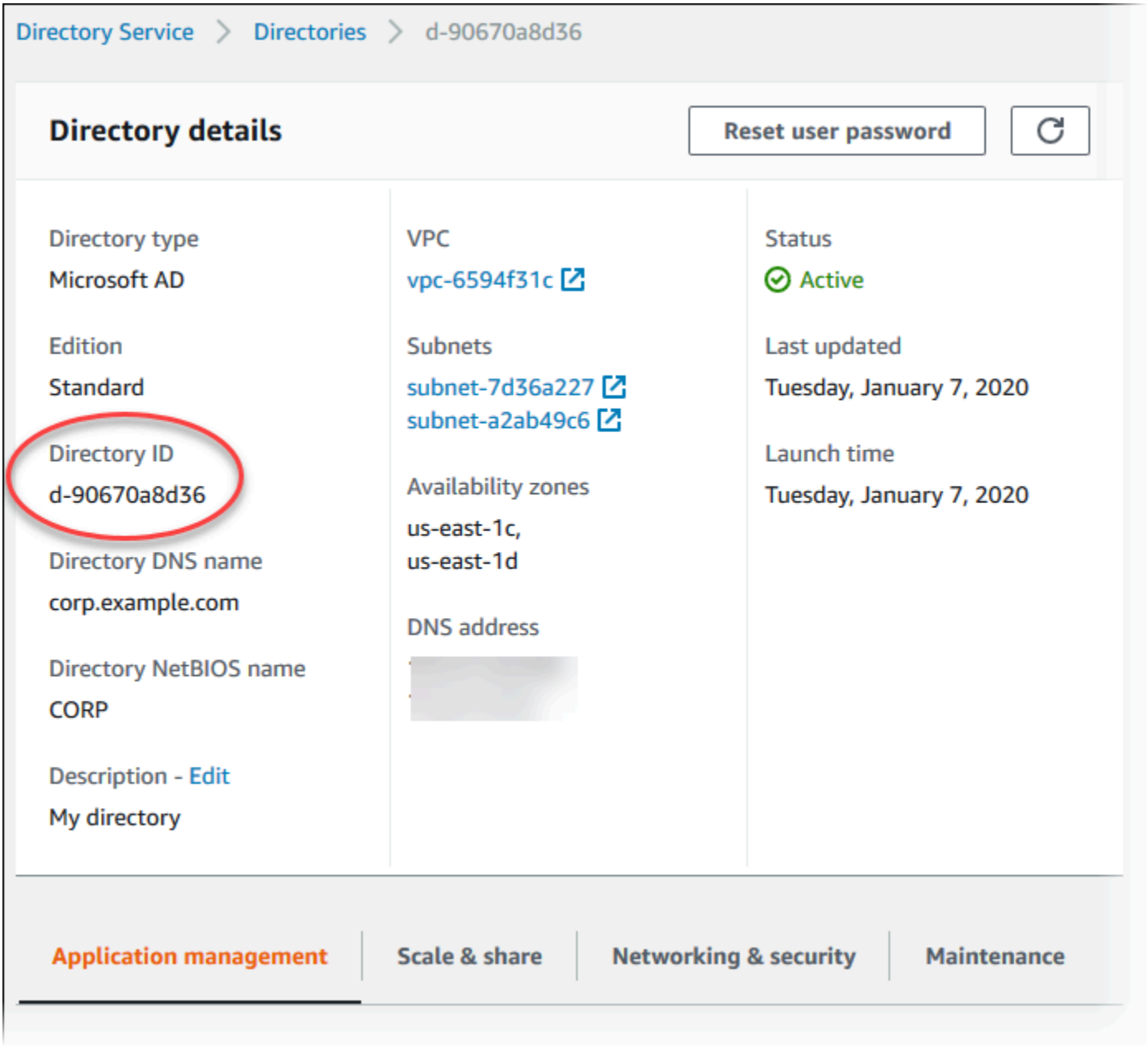
Seleziona le sottoreti per i server di directory. Le due sottoreti devono trovarsi in diverse zone di disponibilità.

7. Seleziona Avanti.
8. Verificare le informazioni della directory Se sono necessarie modifiche, selezionare Previous (Precedente). Quando le informazioni sono corrette, scegli Create Directory (Crea directory).



Per creare la directory sono necessari alcuni minuti. Una volta creata correttamente la directory, il valore Status (Stato) viene modificato in Active (Attivo).

Per consultare le informazioni sulla directory, selezionare l'ID della directory nell'elenco di directory. Prendere nota del valore Directory ID (ID directory). Questo valore è necessario per creare o modificare l'istanza database SQL Server.



Fase 2: creazione del ruolo IAM per l'utilizzo da parte di Amazon RDS

Se utilizzi la console per creare l'istanza database SQL Server, puoi saltare questa fase. Se utilizzi CLI o l'API RDS per creare l'istanza database SQL Server, devi creare un ruolo IAM che utilizza la policy IAM gestita `AmazonRDSDirectoryServiceAccess`. Questo ruolo permette ad Amazon RDS di effettuare chiamate ad AWS Directory Service.

Se utilizzi una policy personalizzata per l'aggiunta di un dominio, anziché utilizzare la policy AWS gestita da `AmazonRDSDirectoryServiceAccess`, assicurati di consentire l'operazione

`ds:GetAuthorizedApplicationDetails`. Questo requisito è efficace a partire da luglio 2019, a causa di una modifica all'API di AWS Directory Service.

La seguente policy IAM, `AmazonRDSDirectoryServiceAccess`, fornisce l'accesso ad AWS Directory Service.

Example Policy IAM per l'accesso ad AWS Directory Service

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
            "ds:DescribeDirectories",
            "ds:AuthorizeApplication",
            "ds:UnauthorizeApplication",
            "ds:GetAuthorizedApplicationDetails"
        ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

Si consiglia di utilizzare le chiavi di contesto delle condizioni globali `aws:SourceArn` e `aws:SourceAccount` nelle relazioni di trust basate sulle risorse per limitare le autorizzazioni del servizio relative a una risorsa specifica. Questo è il modo più efficace per proteggersi dal problema di deputy confused.

Puoi usare le chiavi di contesto delle condizioni globali e avere il valore `aws:SourceArn` che contiene l'ID dell'account. In questo caso, il valore `aws:SourceAccount` e l'account nel valore `aws:SourceArn` deve utilizzare lo stesso ID account quando viene utilizzato nella stessa istruzione.

- Utilizzare `aws:SourceArn` se si desidera un accesso cross-service per una singola risorsa.
- Utilizzare `aws:SourceAccount` se si desidera consentire l'associazione di qualsiasi risorsa in tale account all'uso cross-service.

Nella relazione di trust, assicurati di utilizzare la chiave di contesto della condizione globale `aws:SourceArn` con l'Amazon Resource Name (ARN) completo delle risorse che accedono al

ruolo. Per l'autenticazione Windows, assicurati di includere le istanze database, come mostrato nell'esempio seguente.

Example relazione di affidabilità con la chiave di contesto delle condizioni globali per l'autenticazione di Windows

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "rds.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceArn": [
                        "arn:aws:rds:Region:my_account_ID:db:db_instance_identifier"
                    ]
                }
            }
        }
    ]
}
```

Crea un ruolo IAM utilizzando questa policy IAM e una relazione di affidabilità. Per ulteriori informazioni sulla creazione di ruoli IAM, consulta Creazione di policy gestite dal cliente nella Guida per l'utente di IAM.

Fase 3: creazione e configurazione di utenti e gruppi

Puoi creare utenti e gruppi con lo strumento Utenti Active Directory e computer. Questo è uno degli strumenti Active Directory Domain Services e Active Directory Lightweight Directory Services. Gli utenti possono essere individui singoli o entità che hanno accesso alla tua directory. I gruppi sono molto utili per concedere o negare privilegi ai gruppi di utenti, piuttosto che dover applicare tali privilegi a ogni singolo utente.

Per creare utenti e gruppi in una directory AWS Directory Service, devi essere connesso a un'istanza EC2 Windows che è un membro della directory AWS Directory Service. Devi anche essere connesso come un utente che dispone dei privilegi per creare utenti e gruppi. Per ulteriori informazioni,

consulta Aggiunta di utenti e gruppi (Simple AD e AWS Managed Microsoft AD) nella Guida all'amministrazione di AWS Directory Service.

Fase 4: abilitazione del traffico tra VPC tra la directory e l'istanza database

Se prevedi di individuare la directory e l'istanza database nello stesso VPC, ignora questa fase e passa a Fase 5: creazione o modifica di un'istanza database SQL Server.

Se prevedi di individuare la directory e l'istanza database in VPC differenti, configura il traffico tra VPC utilizzando il peering di VPC o AWS Transit Gateway.

La procedura seguente abilita il traffico tra VPC utilizzando il peering di VPC. Segui le istruzioni in Che cos'è il peering di VPC? nella Amazon Virtual Private Cloud Peering Guide.

Per abilitare il traffico tra VPC utilizzando il peering di VPC

1. Configurare le regole di routing VPC appropriate per garantire che il traffico di rete possa scorrere in entrambe le direzioni.
2. Assicurarsi che il gruppo di protezione dell'istanza database possa ricevere traffico in entrata dal gruppo di sicurezza della directory.
3. Assicurati che non sia presente una regola della lista di controllo accessi (ACL) di rete per bloccare il traffico.

Se la directory appartiene a un account AWS diverso, è necessario condividerla.

Per condividere la directory tra account AWS

1. Avviare la condivisione della directory con l'account AWS in cui verrà creata l'istanza database seguendo le istruzioni in Esercitazione: Condivisione della directory AWS Managed Microsoft AD per unione dominio EC2 agevole nella Guida all'amministrazione di AWS Directory Service.
2. Accedere alla console AWS Directory Service utilizzando l'account per l'istanza database e assicurarsi che lo stato del dominio sia `SHARED` prima di continuare.
3. Dopo aver effettuato l'accesso alla console AWS Directory Service utilizzando l'account per l'istanza database, prendere nota del valore Directory ID (ID directory). Utilizzare questo ID directory per aggiungere l'istanza database al dominio.

Fase 5: creazione o modifica di un'istanza database SQL Server

Crea o modifica un'istanza database SQL Server per l'utilizzo con la directory. Puoi utilizzare la console, CLI o l'API RDS per associare un'istanza database a una directory. Questa operazione può essere eseguita in uno dei seguenti modi:

- Crea una nuova istanza DB di SQL Server utilizzando la console, il comando create-db-instanceCLI o l'operazione API CreateDBInstance RDS.

  Per istruzioni, consulta Creazione di un'istanza database Amazon RDS.
- Modifica un'istanza DB di SQL Server esistente utilizzando la console, il comando modify-db-instanceCLI o l'operazione API ModifyDBInstance RDS.

  Per istruzioni, consulta Modifica di un'istanza database Amazon RDS.
- Ripristina un'istanza DB di SQL Server da uno snapshot DB utilizzando la console, il comando CLI restore-db-instance-from-db-snapshot o l'operazione API RestoreDB DBSnapshot RDS. InstanceFrom

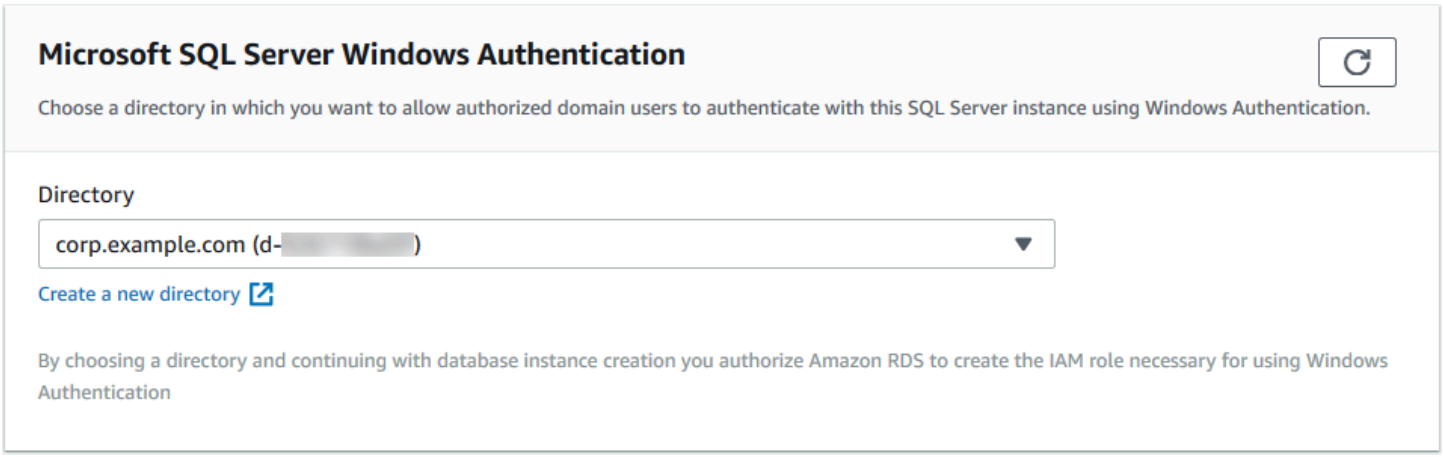
  Per istruzioni, consulta Ripristino da uno snapshot database.
- Ripristina un'istanza DB di SQL Server point-in-time utilizzando la console, il comando restore-db-instance-to- point-in-time CLI o l'operazione RestoreDB RDS API InstanceToPointInTime.

  Per istruzioni, consulta Ripristino a un'ora specifica per un'istanza database.

L'autenticazione di Windows è supportata solo per istanze database di SQL Server in un VPC.

Affinché l'istanza database sia in grado di utilizzare la directory del dominio creata, è richiesto quanto segue:

- Per Directory, devi scegliere l'identificatore di dominio (`d-`*`ID`*) generato durante la creazione della directory.
- Assicurati che il gruppo di protezione VPC disponga di una regola in uscita che consente all'istanza DB di comunicare con la directory.



Quando utilizzi AWS CLI, sono necessari i seguenti parametri per consentire all'istanza database di utilizzare la directory che hai creato:

- Per il parametro `--domain`, utilizza l'identificatore di dominio (d-*ID*) generato durante la creazione della directory.
- Per il parametro `--domain-iam-role-name`, utilizza il ruolo creato che utilizza la policy IAM gestita `AmazonRDSDirectoryServiceAccess`.

Ad esempio, il comando CLI seguente modifica un'istanza database per utilizzare una directory.

PerLinux, o: macOS Unix

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --domain d-ID \
    --domain-iam-role-name role-name
```

Per Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --domain d-ID ^
    --domain-iam-role-name role-name
```

> **Important**
>
> Se modifichi un'istanza database per abilitare l'autenticazione Kerberos, riavvia l'istanza database dopo aver apportato la modifica.

Fase 6: creazione di account di accesso di SQL Server per l'autenticazione di Windows

Utilizza le credenziali dell'utente master Amazon RDS per eseguire la connessione all'istanza database di SQL Server analogamente a quanto avviene con qualsiasi altra istanza database. Poiché l'istanza database viene aggiunta al dominio AWS Managed Microsoft AD, puoi eseguire il provisioning degli account di accesso e degli utenti di SQL Server. Questa operazione viene eseguita dagli utenti e dai gruppi Active Directory nel dominio. Le autorizzazioni per il database vengono gestite tramite le autorizzazioni standard di SQL Server concesse e revocate in base a questi account di accesso Windows.

Affinché un utente di Active Directory possa autenticarsi su SQL Server, deve essere disponibile un account di accesso Windows SQL Server per l'utente o un gruppo di cui l'utente è membro. Il controllo granulare degli accessi viene gestito assegnando o revocando le autorizzazioni per questi login di SQL Server. Un utente che non dispone di un account di accesso SQL Server o appartiene a un gruppo con tale account di accesso, non può accedere all'istanza database SQL Server.

È necessaria l'autorizzazione ALTER ANY LOGIN per creare una connessione SQL Server Active Directory. Se non hai ancora creato account di accesso con questa autorizzazione, esegui la connessione come utente principale dell'istanza database utilizzando l'autenticazione di SQL Server.

Esegui il comando DDL (Data Definition Language) seguente per creare un accesso per SQL Server per un utente o un gruppo di Active Directory.

> **Note**
>
> Specifica utenti o gruppi utilizzando il nome di accesso precedente a Windows 2000 nel formato *domainName**login_name*. Non puoi utilizzare un UPN (User Principle Name) nel formato *login_name*@*DomainName*.

```
USE [master]
GO
```

```
CREATE LOGIN [mydomain\myuser] FROM WINDOWS WITH DEFAULT_DATABASE = [master],
 DEFAULT_LANGUAGE = [us_english];
GO
```

Per maggiori informazioni, consulta CREATE LOGIN (Transact-SQL) nella documentazione di Microsoft Developer Network.

Gli utenti (persone e applicazioni) del tuo dominio possono ora connettersi all'istanza RDS for SQL Server da un computer client associato al dominio utilizzando l'autenticazione Windows.

Gestione di un'istanza database in un dominio

Puoi utilizzare la console, la AWS CLI o l'API Amazon RDS per gestire l'istanza database e la relativa relazione con il dominio. Ad esempio, puoi spostare l'istanza database all'interno, al di fuori o tra domini.

Ad esempio, puoi utilizzare l'API Amazon RDS per effettuare quanto segue:

- Per provare ad associare nuovamente i domini per un'appartenenza non riuscita, utilizza l'operazione API ModifyDBInstance e specifica l'ID della directory dell'appartenenza corrente.
- Per aggiornare il nome del ruolo IAM dell'appartenenza, utilizza l'operazione API `ModifyDBInstance` e specifica l'ID della directory dell'appartenenza corrente e il nuovo ruolo IAM.
- Per rimuovere un'istanza database da un dominio, utilizza l'operazione API `ModifyDBInstance` e specifica `none` come il parametro del dominio.
- Per spostare un'istanza database da un dominio a un altro, utilizza l'operazione API `ModifyDBInstance` e specifica l'identificatore di dominio del nuovo dominio come parametro del dominio.
- Per elencare l'appartenenza per ciascuna istanza database, utilizza l'operazione API DescribeDBInstances.

Appartenenza al dominio

Quando l'istanza di database viene creata o modificata diventa membro del dominio. La console AWS indica lo stato di appartenenza al dominio dell'istanza di database. Lo stato dell'istanza di database può essere uno dei seguenti:

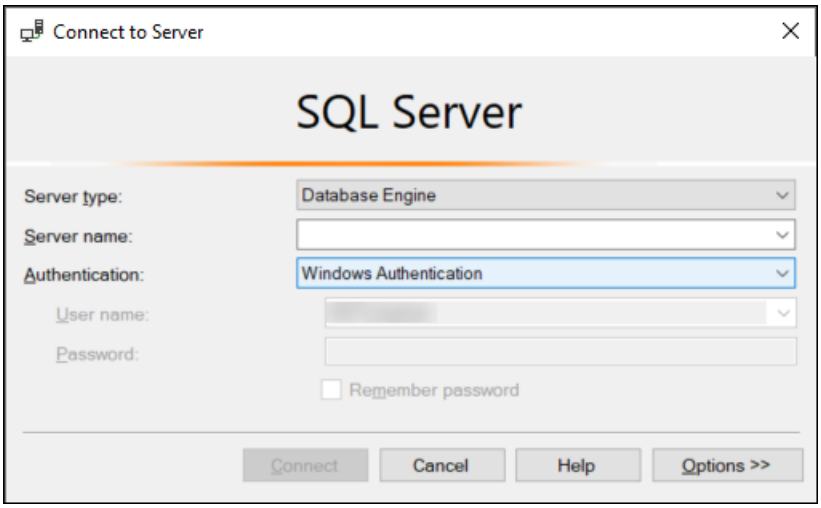
- joined (associata) – L'istanza è membro del dominio.

- joining (in fase di associazione) – L'istanza sta diventando membro del dominio.
- pending-join (associazione in sospeso) – L'associazione dell'istanza è in sospeso.
- pending-maintenance-join— AWS tenterà di rendere l'istanza un membro del dominio durante la successiva finestra di manutenzione programmata.
- pending-removal (rimozione in sospeso) – La rimozione dell'istanza dal dominio è in sospeso.
- pending-maintenance-removal— AWS tenterà di rimuovere l'istanza dal dominio durante la successiva finestra di manutenzione programmata.
- failed (non riuscita) – Un problema di configurazione ha impedito l'associazione dell'istanza al dominio. Verifica e correggi la configurazione prima di eseguire nuovamente il comando di modifica dell'istanza.
- removing (rimozione) – È in corso la rimozione dell'istanza dal dominio.

Una richiesta di associazione a un dominio potrebbe non riuscire a causa di un problema di connettività di rete o di un ruolo IAM non corretto. Ad esempio, è possibile che venga creata un'istanza database o modificata un'istanza esistente senza però che questa diventi un membro di un dominio. In questo caso, emettere nuovamente il comando per creare o modificare l'istanza database o modificare l'istanza appena creata per aggiungerla al dominio.

Connessione a SQL Server mediante autenticazione di Windows

Per eseguire la connessione a SQL Server tramite l'autenticazione di Windows, devi aver effettuato il login al computer che è stato associato al dominio come utente di dominio. Dopo aver avviato SQL Server Management Studio, scegli il tipo di autenticazione Windows Authentication (Autenticazione di Windows) come mostrato di seguito.



Ripristino di un'istanza di database di SQL Server e aggiunta a un dominio

È possibile ripristinare un'istantanea del DB o eseguire point-in-time il ripristino (PITR) per un'istanza DB di SQL Server e quindi aggiungerla a un dominio. Dopo aver ripristinato l'istanza di database, modificala utilizzando il processo illustrato in Fase 5: creazione o modifica di un'istanza database SQL Server per aggiungere l'istanza a un dominio.

Aggiornamento delle applicazioni per la connessione a istanze di database Microsoft SQL Server utilizzando nuovi certificati SSL/TLS

A partire dal 13 gennaio 2023, Amazon RDS ha pubblicato nuovi certificati dell'autorità di certificazione (CA) per la connessione alle istanze database RDS utilizzando Secure Socket Layer o Transport Layer Security (SSL/TLS). Di seguito sono disponibili le informazioni sull'aggiornamento delle applicazioni per utilizzare i nuovi certificati.

Questo argomento aiuta a determinare se le applicazioni client utilizzano SSL/TLS per connettersi alle istanze database. In caso affermativo, puoi determinare anche se le applicazioni richiedono la verifica del certificato per la connessione.

> **Note**
>
> Alcune applicazioni sono configurate per connettersi alle istanze database SQL Server solo se sono in grado di verificare correttamente il certificato sul server.
> Per queste applicazioni, è necessario aggiornare gli archivi di trust delle applicazioni client per includere i nuovi certificati CA.

Dopo aver aggiornato i certificati CA negli archivi di trust delle applicazioni client, puoi ruotare i certificati nelle istanze database. Consigliamo vivamente di testare queste procedure in un ambiente di sviluppo o di gestione temporanea prima di implementarle negli ambienti di produzione.

Per ulteriori informazioni sulla rotazione dei certificati, consulta Rotazione del certificato SSL/TLS. Per ulteriori informazioni sul download, consulta . Per informazioni sull'utilizzo di SSL/TLS con le istanze database Microsoft SQL Server, consulta Utilizzo di SSL con un'istanza database Microsoft SQL Server.

Argomenti

- Determinare se un'applicazione si connette all'istanza database Microsoft SQL Server mediante SSL
- Determinare se un client richiede la verifica del certificato per la connessione
- Aggiornare l'archivio di trust delle applicazioni

Determinare se un'applicazione si connette all'istanza database Microsoft SQL Server mediante SSL

Verifica la configurazione delle istanze database per il valore del parametro `rds.force_ssl`. Per impostazione predefinita, il parametro `rds.force_ssl` è impostato su 0 (off). Se il parametro`rds.force_ssl` è impostato su1 (attivato), i client devono utilizzare SSL/TLS per le connessioni. Per ulteriori informazioni sui gruppi di parametri, consultare Utilizzo di gruppi di parametri.

Eseguire la seguente query per ottenere l'opzione di crittografia corrente per tutte le connessioni aperte a un'istanza database. Se la connessione è crittografata, la colonna `ENCRYPT_OPTION` restituisce `TRUE`.

```
select SESSION_ID,
    ENCRYPT_OPTION,
    NET_TRANSPORT,
    AUTH_SCHEME
    from SYS.DM_EXEC_CONNECTIONS
```

La query mostra solo le connessioni correnti. Non indica se le applicazioni collegate e scollegate in precedenza hanno utilizzato SSL.

Determinare se un client richiede la verifica del certificato per la connessione

Puoi verificare se i diversi tipi di client richiedono la verifica del certificato per la connessione.

> **Note**
>
> Se si utilizzano connettori diversi da quelli elencati, consultare la documentazione specifica del connettore per le informazioni relative a come vengono applicate le connessioni crittografate. Per ulteriori informazioni, consultare Moduli di connessione per i database Microsoft SQL nella documentazione di Microsoft SQL Server.

Avviare SQL Server Management Studio

Verificare se è attivata la crittografia per le connessioni di SQL Server Management Studio:

1. Avviare SQL Server Management Studio.
2. Per Connect to server (Connetti al server) digitare le informazioni del server, il nome utente di accesso e la password.
3. Scegli Opzioni.
4. Verificare se è selezionata Encrypt connection (Connessione crittografata) nella pagina di connessione.

Per ulteriori informazioni su SQL Server Management Studio, consulta Utilizzare SQL Server Management Studio.

Sqlcmd

Il seguente esempio con il client `sqlcmd` illustra come verificare la connessione SQL Server di uno script per determinare se le connessioni riuscite richiedono un certificato valido. Per ulteriori informazioni, consulta la sezione relativa alla connessione con sqlcmd nella documentazione di Microsoft SQL Server.

Quando utilizzi `sqlcmd`, la connessione SSL richiede la verifica del certificato CA del server se utilizzi l'argomento del comando `-N` per crittografare le connessioni come nell'esempio seguente.

```
$ sqlcmd -N -S dbinstance.rds.amazon.com -d ExampleDB
```

> **Note**
>
> Se `sqlcmd` viene richiamato con l'opzione `-C`, considera attendibile il certificato del server, anche se non corrisponde all'archivio store di trust lato client.

ADO.NET

Nel seguente esempio, l'applicazione di collega utilizzando SSL e il certificato del server deve essere verificato.

```
using SQLC = Microsoft.Data.SqlClient;

...

    static public void Main()
    {
        using (var connection = new SQLC.SqlConnection(
            "Server=tcp:dbinstance.rds.amazon.com;" +
            "Database=ExampleDB;User ID=LOGIN_NAME;" +
            "Password=YOUR_PASSWORD;" +
            "Encrypt=True;TrustServerCertificate=False;"
            ))
        {
            connection.Open();
            ...
        }
```

Java

Nel seguente esempio, l'applicazione di collega utilizzando SSL e il certificato del server deve essere verificato.

```
String connectionUrl =
    "jdbc:sqlserver://dbinstance.rds.amazon.com;" +
    "databaseName=ExampleDB;integratedSecurity=true;" +
    "encrypt=true;trustServerCertificate=false";
```

Per abilitare la crittografia SSL per i client che si connettono utilizzando JDBC, può essere necessario aggiungere il certificato Amazon RDS all'archivio di certificati CA Java. Per istruzioni, consultare Configurazione del client per la crittografia nella documentazione Microsoft SQL Server. È anche possibile fornire direttamente il nome del file del certificato CA attendibile aggiungendo `trustStore=`*`path-to-certificate-trust-store-file`* alla stringa di connessione.

> **Note**
>
> Se si utilizza `TrustServerCertificate=true` (o l'equivalente) nella stringa di connessione, il processo di connessione ignora la convalida della catena di attendibilità.

In questo caso l'applicazione si connette anche se non è possibile verificare il certificato. L'utilizzo di `TrustServerCertificate=false` applica la convalida dei certificati ed è una best practice.

Aggiornare l'archivio di trust delle applicazioni

Puoi aggiornare l'archivio di trust delle applicazioni che utilizzano Microsoft SQL Server. Per istruzioni, consulta Crittografia di connessioni specifiche. Consultare anche Configurazione del client per la crittografia nella documentazione Microsoft SQL Server.

Se si sta utilizzando un sistema operativo diverso da Microsoft Windows, consultare la documentazione relativa alla distribuzione del software per l'implementazione SSL/TLS per informazioni relative all'aggiunta di un nuovo certificato CA root. Ad esempio, OpenSSL e GnuTLS sono le opzioni più comunemente usate. Utilizzare il metodo di implementazione per aggiungere trust al certificato CA root RDS. Microsoft fornisce le istruzioni per la configurazione dei certificati su alcuni sistemi.

Per ulteriori informazioni sul download del certificato root, consulta .

Per gli script di esempio che importano i certificati, consulta Script di esempio per l'importazione di certificati nel tuo archivio di trust.

> **Note**
>
> Quando aggiorni l'archivio di trust puoi conservare i certificati meno recenti oltre ad aggiungere i nuovi certificati.

Aggiornamento del motore di database Microsoft SQL Server

Quando Amazon RDS supporta una nuova versione di un motore di database, puoi effettuare l'aggiornamento delle istanze database alla nuova versione. Sono disponibili due tipi di aggiornamenti per le istanze database SQL Server: per la versione principale e per la versione secondaria.

Gli aggiornamenti di versione principali possono contenere modifiche al database non compatibili con le versioni precedenti delle applicazioni esistenti. Ne risulta che è necessario eseguire manualmente gli aggiornamenti della versione principale per le proprie istanze database. Puoi avviare manualmente un aggiornamento principale a una versione modificando l'istanza. Tuttavia, prima di eseguire un aggiornamento alla versione principale, è consigliabile testarlo seguendo i passaggi descritti in Verifica di un aggiornamento.

Al contrario, gli aggiornamento secondari a una versione includono solo modifiche compatibili con le versioni precedenti delle applicazioni esistenti. Puoi avviare un aggiornamento a una versione secondaria manualmente modificando la tua istanza database.

Nell'esempio seguente, il comando CLI restituisce una risposta con `AutoUpgrade` che indica true, ovvero che gli aggiornamenti sono automatici.

```
...

"ValidUpgradeTarget": [
    {
        "Engine": "sqlserver-se",
        "EngineVersion": "14.00.3281.6.v1",
        "Description": "SQL Server 2017 14.00.3281.6.v1",
        "AutoUpgrade": true,
        "IsMajorVersionUpgrade": false
    }

...
```

Per ulteriori informazioni sull'esecuzione degli aggiornamenti, consulta Aggiornamento di un'istanza database SQL Server. Per informazioni sulle versioni di SQL Server disponibili in Amazon RDS, consulta Amazon RDS for Microsoft SQL Server.

Argomenti

- Panoramica dell'aggiornamento

- Aggiornamenti di una versione principale
- Considerazioni su Multi-AZ e sull'ottimizzazione in memoria
- Considerazioni sulle repliche di lettura
- Considerazioni su gruppi di opzioni
- Considerazioni sui gruppi di parametri
- Verifica di un aggiornamento
- Aggiornamento di un'istanza database SQL Server
- Aggiornamento di istanze database obsolete prima del termine del supporto

Panoramica dell'aggiornamento

Durante il processo di aggiornamento, Amazon RDS acquisisce due snapshot DB. Il primo snapshot DB è relativo all'istanza database prima delle modifiche legate all'aggiornamento. Il secondo snapshot DB viene acquisito al termine dell'aggiornamento.

> **Note**
>
> Amazon RDS acquisisce gli snapshot DB solo se hai impostato il periodo di retention dei backup per l'istanza database su un valore maggiore di 0. Per cambiare il periodo di retention dei backup, consulta Modifica di un'istanza database Amazon RDS.

Al completamento di un aggiornamento, non puoi ripristinare la versione precedente del motore di database. Se desideri tornare alla versione precedente, ripristina dallo snapshot DB acquisito prima dell'aggiornamento per creare una nuova istanza database.

Durante un aggiornamento di una versione principale o secondaria di SQL Server, per i parametri Free Storage Space (Spazio di storage libero) e Disk Queue Depth (Profondità coda disco) viene visualizzato -1. Al completamento dell'aggiornamento, entrambi i parametri vengono ripristinati al valore normale.

Aggiornamenti di una versione principale

Amazon RDS attualmente supporta gli aggiornamenti delle versioni principali seguenti per un'istanza database Microsoft SQL Server.

Puoi ora aggiornare l'istanza database esistente a SQL Server 2017 o 2019 da qualsiasi versione, ad eccezione di SQL Server 2008. Per eseguire l'aggiornamento da SQL Server 2008, prima di tutto aggiorna l'istanza a una delle altre versioni.

Versione corrente	Versioni supportate per l'aggiornamento
SQL Server 2019	SQL Server 2022
SQL Server 2017	SQL Server 2022 SQL Server 2019
SQL Server 2016	SQL Server 2022 SQL Server 2019 SQL Server 2017
SQL Server 2014	SQL Server 2022 SQL Server 2019 SQL Server 2017 SQL Server 2016
SQL Server 2012 (fine del supporto)	SQL Server 2022 SQL Server 2019 SQL Server 2017 SQL Server 2016 SQL Server 2014
SQL Server 2008 R2 (fine del supporto)	SQL Server 2016 SQL Server 2014 SQL Server 2012

È possibile utilizzare una AWS CLI query, come nell'esempio seguente, per trovare gli aggiornamenti disponibili per una particolare versione del motore di database.

Example

Per LinuxmacOS, oUnix:

```
aws rds describe-db-engine-versions \
    --engine sqlserver-se \
    --engine-version 14.00.3281.6.v1 \
    --query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" \
    --output table
```

Per Windows:

```
aws rds describe-db-engine-versions ^
    --engine sqlserver-se ^
    --engine-version 14.00.3281.6.v1 ^
    --query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" ^
    --output table
```

L'output mostra che è possibile aggiornare la versione 14.00.3281.6 alle versioni più recente di SQL Server 2017 o 2019.

```
-------------------------
|DescribeDBEngineVersions|
+------------------------+
|      EngineVersion     |
+------------------------+
|  14.00.3294.2.v1       |
|  14.00.3356.20.v1      |
|  14.00.3381.3.v1       |
|  14.00.3401.7.v1       |
|  14.00.3421.10.v1      |
|  14.00.3451.2.v1       |
|  15.00.4043.16.v1      |
|  15.00.4073.23.v1      |
|  15.00.4153.1.v1       |
|  15.00.4198.2.v1       |
|  15.00.4236.7.v1       |
+------------------------+
```

Livello di compatibilità del database

È possibile utilizzare i livelli di compatibilità del database Microsoft SQL Server per modificare alcuni comportamenti del database in modo da emulare versioni precedenti di SQL Server. Per ulteriori informazioni, consulta Livello di compatibilità nella documentazione Microsoft.

Quando aggiorni l'istanza database, tutti i database esistenti rimangono impostati sul livello di compatibilità originale. Se, ad esempio, esegui l'aggiornamento da SQL Server 2014 a SQL Server 2016, tutti i database esistenti hanno un livello di compatibilità di 120. I nuovi database creati dopo l'aggiornamento hanno il livello di compatibilità 130.

È possibile modificare il livello di compatibilità di un database tramite il comando ALTER DATABASE. Per modificare, ad esempio, un database denominato `customeracct` in modo che sia compatibile con SQL Server 2014, utilizza il comando seguente:

```
ALTER DATABASE customeracct SET COMPATIBILITY_LEVEL = 120
```

Considerazioni su Multi-AZ e sull'ottimizzazione in memoria

Amazon RDS supporta le implementazioni Multi-AZ per le istanze database che eseguono Microsoft SQL Server utilizzando i gruppi di disponibilità Always On (AG) o il mirroring del database (DBM) di SQL Server. Per ulteriori informazioni, consulta Implementazioni Multi-AZ per Amazon RDS for Microsoft SQL Server.

Se l'istanza database è in un'implementazione Multi-AZ, vengono aggiornate sia l'istanza database principale che quella di standby. Amazon RDS esegue il rolling degli aggiornamenti. Si verifica un'interruzione solo per la durata di un failover.

SQL Server dal 2014 al 2019 Enterprise Edition supporta l'ottimizzazione in memoria.

Considerazioni sulle repliche di lettura

Durante l'aggiornamento della versione del database, Amazon RDS aggiorna anche tutte le repliche di lettura insieme all'istanza database primaria. Amazon RDS non supporta gli aggiornamenti della versione del database sulle repliche di lettura separatamente. Per ulteriori informazioni sulle repliche di lettura, consultare Utilizzo di repliche di lettura per Microsoft SQL Server in Amazon RDS.

Quando aggiorni la versione del database dell'istanza database primaria, tutte le relative repliche di lettura vengono aggiornate automaticamente. Amazon RDS aggiornerà tutte le repliche di lettura

contemporaneamente prima di aggiornare l'istanza database di origine. Le repliche di lettura potrebbero non essere disponibili fino al completamento dell'aggiornamento della versione del database sull'istanza DB primaria.

Considerazioni su gruppi di opzioni

Se l'istanza database utilizza un gruppo DB di opzioni personalizzato, in alcuni casi Amazon RDS non può assegnare automaticamente all'istanza database un nuovo gruppo di opzioni. Ad esempio, quando esegui l'aggiornamento a una nuova versione principale, devi specificare un nuovo gruppo di opzioni. Ti consigliamo di creare un nuovo gruppo di opzioni e di aggiungere le stesse opzioni presenti nel gruppo di opzioni personalizzato esistente.

Per ulteriori informazioni, consulta Creazione di un gruppo di opzioni o Copia di un gruppo di opzioni.

Considerazioni sui gruppi di parametri

Se l'istanza database utilizza un gruppo di parametri del database personalizzato:

- Amazon RDS riavvia automaticamente l'istanza database dopo un aggiornamento.
- In alcuni casi, RDS non è in grado di assegnare automaticamente un nuovo gruppo di parametri all'istanza database.

  Ad esempio, quando esegui l'aggiornamento a una nuova versione principale, devi specificare un nuovo gruppo di parametri. Ti consigliamo di creare un nuovo gruppo di parametri e di configurare i parametri in modo analogo al gruppo di parametri personalizzato esistente.

Per ulteriori informazioni, consulta Creazione di un gruppo di parametri del database o Copia di un gruppo di parametri database.

Verifica di un aggiornamento

Prima di eseguire l'aggiornamento di una versione principale nell'istanza database, testa a fondo il database e tutte le applicazioni che accedono a esso per verificarne la compatibilità con la nuova versione. È consigliabile utilizzare la procedura seguente.

Per testare un aggiornamento di una versione principale

1. Esaminare Upgrade del Server SQL nella documentazione Microsoft per la nuova versione del motore di database per verificare se vi sono problemi di compatibilità che potrebbero interessare il database o le applicazioni:

2. Se l'istanza database utilizza un gruppo di opzioni personalizzato, creare un nuovo gruppo di opzioni compatibile con la nuova versione a cui si sta eseguendo l'aggiornamento. Per ulteriori informazioni, consulta Considerazioni su gruppi di opzioni.
3. Se l'istanza database utilizza un gruppo di parametri personalizzato, creare un nuovo gruppo di parametri compatibile con la nuova versione a cui si sta eseguendo l'aggiornamento. Per ulteriori informazioni, consulta Considerazioni sui gruppi di parametri.
4. Creare uno snapshot DB dell'istanza database da aggiornare. Per ulteriori informazioni, consulta Creazione di uno snapshot DB per un'istanza DB Single-AZ.
5. Ripristinare lo snapshot DB per creare una nuova istanza database di test. Per ulteriori informazioni, consulta Ripristino da uno snapshot database.
6. Modificare la nuova istanza database di test per aggiornarla alla nuova versione, utilizzando uno dei metodi seguenti:

   - Console
   - AWS CLI
   - API RDS
7. Valutare lo storage utilizzato dall'istanza aggiornata per determinare se l'aggiornamento richiede storage aggiuntivo.
8. Eseguire quanti più test di controllo qualità possibili per l'istanza database aggiornata come necessario per assicurare che il database e l'applicazione funzionino correttamente con la nuova versione. Implementare qualsiasi nuovo test necessario per valutare l'impatto dei problemi di compatibilità identificati nella fase 1. Testare tutte le stored procedure e le funzioni. Indirizzare le versioni di test delle applicazioni all'istanza database aggiornata.
9. Se tutti i test vengono superati, eseguire l'aggiornamento nell'istanza database di produzione. È consigliabile non permettere le operazioni di scrittura nell'istanza database fino a quando non si è certi che tutto funzioni correttamente.

Aggiornamento di un'istanza database SQL Server

Per informazioni sull'aggiornamento manuale o automatico di un'istanza database di SQL Server, consulta quanto segue:

- Aggiornamento della versione del motore di un'istanza database
- Best practice per l'aggiornamento di SQL Server 2008 R2 a SQL Server 2016 su Amazon RDS for SQL Server

> **Important**
>
> Se disponi di istantanee crittografate utilizzando AWS KMS, ti consigliamo di avviare un aggiornamento prima della fine del supporto.

Aggiornamento di istanze database obsolete prima del termine del supporto

Quando una versione principale diventa obsoleta, non puoi installarla su nuove istanze database. RDS proverà ad aggiornare automaticamente tutte le istanze database esistenti.

Se devi ripristinare un'istanza DB obsoleta, puoi eseguire il point-in-time ripristino (PITR) o ripristinare un'istantanea. In questo modo avrai temporaneamente accesso a un'istanza database che utilizza la versione diventata obsoleta. Tuttavia, quando una versione principale diventa totalmente obsoleta, anche queste istanze database saranno automaticamente aggiornate a una versione supportata.

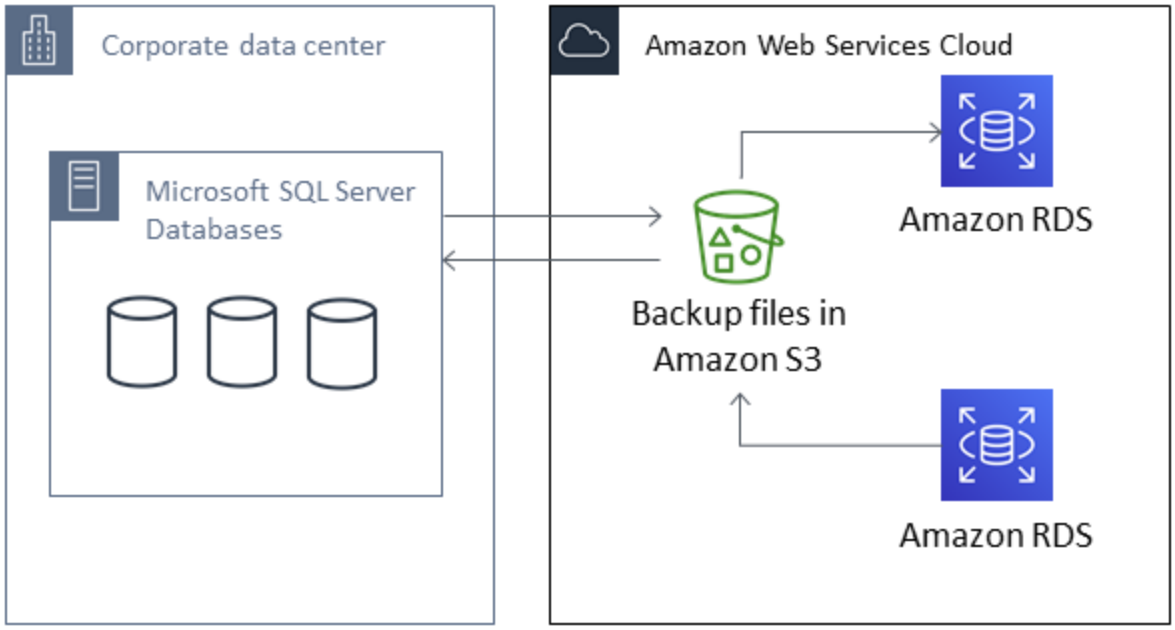
Importazione ed esportazione di database SQL Server mediante backup e ripristino nativi

Amazon RDS supporta il backup nativo e il ripristino dei database Microsoft SQL Server tramite file di backup completi (file .bak). Quanto utilizzi RDS, puoi accedere ai file archiviati in Amazon S3 invece di utilizzare il file system locale sul server del database.

Ad esempio, puoi creare un backup completo dal server locale, archiviarlo in S3 e quindi ripristinarlo in un'istanza database Amazon RDS esistente. Puoi anche eseguire i backup da RDS, archiviarli in S3 e quindi ripristinarli quando lo desideri.

Il backup e il ripristino nativi sono disponibili in tutte le AWS regioni per le istanze DB Single-AZ e Multi-AZ, incluse le istanze DB Multi-AZ con repliche di lettura. Backup e ripristino nativi sono disponibili per tutte le edizioni di Microsoft SQL Server supportate su Amazon RDS.

Il seguente schema mostra gli scenari supportati.



L'utilizzo di file .bak nativi per effettuare il backup e il ripristino di database è in genere il modo più veloce per compiere queste operazioni. Vi sono molteplici vantaggi aggiuntivi con l'utilizzo di backup e ripristino nativi. Ad esempio, puoi eseguire le operazioni seguenti:

- Migrare i database a o da Amazon RDS.
- Spostare i database tra le istanze database RDS for SQL Server.
- Migrare dati, schemi, procedure archiviate, trigger e altri codici di database in file .bak.

- Eseguire backup e ripristino di singoli database invece che di intere istanze database.
- Creare copie di database per sviluppo, attività di test, formazione e dimostrazioni.
- Archiviare e trasferire i file di backup con Amazon S3, per un ulteriore livello di protezione per il disaster recovery.
- Creare backup nativi di database con Transparent Data Encryption (TDE) attivato e ripristinare tali backup nei database on-premise. Per ulteriori informazioni, consulta Supporto per Transparent Data Encryption in SQL Server.
- Ripristinare i backup nativi dei database on-premise con TDE attivato in istanze database di RDS per SQL Server. Per ulteriori informazioni, consulta Supporto per Transparent Data Encryption in SQL Server.

Indice

Limitazioni e consigli

Le limitazioni all'utilizzo di backup e ripristino nativi sono le seguenti:

- Non puoi eseguire il backup o il ripristino da un bucket Amazon S3 in una AWS regione diversa dalla tua istanza database Amazon RDS.
- Non è possibile ripristinare un database se esiste già un database con lo stesso nome. I nomi dei database sono univoci.
- Consigliamo vivamente di non ripristinare backup da un fuso orario in un fuso orario diverso. Se ripristini backup da un fuso orario in un fuso orario diverso, devi controllare le query e le applicazioni per verificare gli effetti del cambiamento di fuso orario.
- Amazon S3 ha un limite di dimensione di 5 TB per file. Per i backup nativi di database di grandi dimensioni, è possibile utilizzare il backup con più file.
- La dimensione massima del database di cui è possibile eseguire il backup su S3 dipende dalla memoria, dalla CPU, dall'I/O e dalle risorse di rete disponibili nell'istanza del DB. Più grande è il database, maggiore è la quantità di memoria utilizzata dall'agente di backup. I nostri test mostrano

che è possibile eseguire un backup compresso di un database da 16 TB sui tipi di istanza di nuova generazione, a partire dalle dimensioni delle istanze `2xlarge` e più grandi, date le risorse di sistema sufficienti.

- Non puoi eseguire il backup o il ripristino da più di 10 file di backup alla volta.
- Il backup differenziale si basa sull'ultimo backup completo. Per completare i backup differenziali non puoi effettuare una snapshot tra l'ultimo backup completo e il backup differenziale. Se si desidera effettuare un backup differenziale, ma esiste uno snapshot, effettua un altro backup completo prima di procedere con il backup differenziale.
- I ripristini differenziali e di log non sono supportati per i database con file in cui il relativo file_guid (identificatore univoco) è impostato su `NULL`.
- Puoi eseguire fino a due attività di backup o ripristino contemporaneamente.
- Non è possibile eseguire backup dei log nativi da SQL Server Amazon RDS.
- RDS supporta ripristini nativi di database fino a 16 TB. I ripristini nativi di database su SQL Server Express Edition sono limitati a 10 GB.
- Non puoi effettuare un backup nativo durante la finestra di manutenzione o quando Amazon RDS è impegnato ad acquisire uno snapshot del database. Se un'attività di backup nativa si sovrappone alla finestra di backup giornaliero di Servizi RDS, l'attività di backup nativa viene annullata.
- Nelle istanze database Multi-AZ è possibile ripristinare in modo nativo solo i database con backup nel modello di ripristino "Full" (Completo).
- Non è supportato il ripristino da backup differenziali su istanze Multi-AZ.
- Non è supportata la chiamata delle procedure RDS per backup e ripristino nativi in una transazione.
- Utilizza una crittografia AWS KMS key simmetrica per crittografare i backup. Amazon RDS non supporta le chiavi KMS asimmetriche. Per ulteriori informazioni, consulta Creazione di chiavi KMS di crittografia simmetrica nella Guida per gli sviluppatori di AWS Key Management Service .
- I file di backup nativi sono crittografati con la chiave KMS specifica utilizzando la modalità di crittografia "Solo crittografia". Quando vengono ripristinati i file di backup crittografati, tieni presente che sono crittografati con modalità di crittografia "Solo crittografia".
- Non è possibile ripristinare un database che contiene un gruppo di file FILESTREAM.

Ti consigliamo di utilizzare backup e ripristino nativi per migrare il tuo database in RDS se il database può essere offline quando il file di backup viene creato, copiato e ripristinato. Se il database locale non può essere offline, ti consigliamo di utilizzare il AWS Database Migration Service per migrare

il database su Amazon RDS. Per ulteriori informazioni, consulta Cos'è? AWS Database Migration Service

Backup e ripristino nativi non hanno lo scopo di sostituire le funzionalità di ripristino dei dati della funzione di copia di snapshot tra regioni. Ti consigliamo di utilizzare la copia istantanea per copiare lo snapshot del database in un'altra AWS regione per il disaster recovery tra regioni in Amazon RDS. Per ulteriori informazioni, consulta Copia di una snapshot DB..

Configurazione di backup e ripristino nativi

Per configurare backup e ripristino nativi, sono necessari tre componenti:

1. Un bucket Amazon S3 per archiviare i file di backup.

   Devi disporre di un bucket S3 per utilizzare i file di backup e quindi caricare i backup che desideri migrare su RDS. Se disponi già di un bucket Amazon S3, puoi utilizzarlo. Se non hai già un bucket, puoi crearne uno nuovo. In alternativa, puoi scegliere che venga creato automaticamente un nuovo bucket quando aggiungi l'opzione `SQLSERVER_BACKUP_RESTORE` usando la AWS Management Console.

   Per ulteriori informazioni, consulta la Guida per l'utente di Amazon Simple Storage Service.

2. Un ruolo AWS Identity and Access Management (IAM) per accedere al bucket.

   Se disponi già di un ruolo IAM, puoi utilizzarlo. In alternativa, puoi scegliere che venga creato automaticamente un nuovo ruolo IAM quando aggiungi l'opzione `SQLSERVER_BACKUP_RESTORE` usando la AWS Management Console. In alternativa, è possibile crearne uno nuovo manualmente.

   Se vuoi creare un nuovo ruolo IAM manualmente, segui l'approccio illustrato nella sezione successiva. Eseguire la stessa operazione se si desidera associare le relazioni di trust e i criteri di autorizzazione a un ruolo IAM esistente.

3. L'opzione `SQLSERVER_BACKUP_RESTORE` aggiunta a un gruppo di opzioni nella tua istanza database.

   Per abilitare backup e ripristino nativi sulla tua istanza database, devi aggiungere l'opzione `SQLSERVER_BACKUP_RESTORE` a un gruppo di opzioni sulla tua istanza database. Per ulteriori informazioni e istruzioni, consulta Supporto per backup nativo e ripristino in SQL Server.

Creazione manuale di un ruolo IAM per backup e ripristino nativi

Se lo desideri, puoi creare manualmente un nuovo ruolo IAM da utilizzare con il backup e il ripristino nativi. In tal caso, crea un ruolo da cui delegare le autorizzazioni del servizio Amazon RDS al bucket Amazon S3. Quando si crea un ruolo IAM, si allegano una relazione di trust e un criterio di autorizzazioni. La policy di attendibilità consente a Servizi RDS di assumere questo ruolo. La policy di autorizzazione definisce le operazioni che questo ruolo può eseguire. Per ulteriori informazioni sulla creazione del ruolo, consulta la pagina Creazione di un ruolo per delegare le autorizzazioni a un servizio AWS.

Per la funzione di backup e ripristino nativi, utilizza le policy di attendibilità e autorizzazione simili agli esempi presenti in questa sezione. Nell'esempio che segue utilizziamo il nome del servizio `rds.amazonaws.com` come alias per tutti gli account di servizio. Negli altri esempi specifichiamo un Amazon Resource Name (ARN) per identificare un altro account, un altro utente o un altro ruolo al quale concediamo l'accesso nella policy di attendibilità.

Si consiglia di utilizzare le chiavi di contesto delle condizioni globali `aws:SourceArn` e `aws:SourceAccount` nelle relazioni di trust basate sulle risorse per limitare le autorizzazioni del servizio relative a una risorsa specifica. Questo è il modo più efficace per proteggersi dal problema di deputy confused.

Puoi usare le chiavi di contesto delle condizioni globali e avere il valore `aws:SourceArn` che contiene l'ID dell'account. In questo caso, il valore `aws:SourceAccount` e l'account nel valore `aws:SourceArn` deve utilizzare lo stesso ID account quando viene utilizzato nella stessa istruzione.

- Utilizzare `aws:SourceArn` se si desidera un accesso cross-service per una singola risorsa.
- Utilizzare `aws:SourceAccount` se si desidera consentire l'associazione di qualsiasi risorsa in tale account all'uso cross-service.

Nella relazione di attendibilità, assicurati di utilizzare la chiave di contesto della condizione globale `aws:SourceArn` con l'ARN completo delle risorse che accedono al ruolo. Per il backup e il ripristino nativi, assicurati di includere sia il gruppo di opzioni database che le istanze database, come illustrato nell'esempio seguente.

Example relazione di attendibilità con chiave di contesto delle condizioni globali per backup e ripristino nativi

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "rds.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceArn": [
                        "arn:aws:rds:Region:my_account_ID:db:db_instance_identifier",
                        "arn:aws:rds:Region:my_account_ID:og:option_group_name"
                    ]
                }
            }
        }
    ]
}
```

Nell'esempio che segue viene utilizzato un ARN per specificare la risorsa. Per ulteriori informazioni, consulta la pagina Amazon Resource Names (ARN).

Example Policy di autorizzazione per backup e ripristino nativi senza supporto della crittografia

```
{
    "Version": "2012-10-17",
    "Statement":
    [
        {
        "Effect": "Allow",
        "Action":
            [
                "s3:ListBucket",
                "s3:GetBucketLocation"
            ],
        "Resource": "arn:aws:s3:::bucket_name"
        },
        {
        "Effect": "Allow",
        "Action":
            [
                "s3:GetObjectAttributes",
```

```
                "s3:GetObject",
                "s3:PutObject",
                "s3:ListMultipartUploadParts",
                "s3:AbortMultipartUpload"
            ],
        "Resource": "arn:aws:s3:::bucket_name/*"
        }
    ]
}
```

Example Policy di autorizzazione per backup e ripristino nativi con supporto della crittografia

Se desideri crittografare i tuoi file di backup, includi una chiave di crittografia nella policy di autorizzazione. Per ulteriori informazioni sulle chiavi di crittografia, consulta Nozioni di base nella Guida per gli sviluppatori di AWS Key Management Service .

> **Note**
>
> Per crittografare i backup è necessario utilizzare una chiave KMS di crittografia simmetrica. Amazon RDS non supporta le chiavi KMS asimmetriche. Per ulteriori informazioni, consulta Creazione di chiavi KMS di crittografia simmetrica nella Guida per gli sviluppatori di AWS Key Management Service .
> Il ruolo IAM deve inoltre essere un utente chiave e un amministratore chiave per la chiave KMS, ovvero deve essere specificato nel criterio chiave. Per ulteriori informazioni, consulta Creazione di chiavi KMS di crittografia simmetrica nella Guida per gli sviluppatori di AWS Key Management Service .

```
{
    "Version": "2012-10-17",
    "Statement":
    [
        {
        "Effect": "Allow",
        "Action":
            [
                "kms:DescribeKey",
                "kms:GenerateDataKey",
                "kms:Encrypt",
                "kms:Decrypt"
            ],
```

```
        "Resource": "arn:aws:kms:region:account-id:key/key-id"
        },
        {
        "Effect": "Allow",
        "Action":
            [
                "s3:ListBucket",
                "s3:GetBucketLocation"
            ],
        "Resource": "arn:aws:s3:::bucket_name"
        },
        {
        "Effect": "Allow",
        "Action":
            [
                "s3:GetObjectAttributes",
                "s3:GetObject",
                "s3:PutObject",
                "s3:ListMultipartUploadParts",
                "s3:AbortMultipartUpload"
            ],
        "Resource": "arn:aws:s3:::bucket_name/*"
        }
    ]
}
```

Uso di backup e ripristino nativi

Dopo aver abilitato e configurato backup e ripristino nativi, puoi iniziare a utilizzarli. Per farlo, devi innanzitutto connetterti al database Microsoft SQL Server e poi chiamare una stored procedure Amazon RDS per eseguire il lavoro. Per istruzioni sulla connessione al tuo database, consulta Connessione a un'istanza database che esegua il motore di database di Microsoft SQL Server.

Alcune delle stored procedure richiedono di fornire un Amazon Resource Name (ARN) al tuo file e bucket Amazon S3. Il formato per l'ARN è `arn:aws:s3:::bucket_name/file_name.extension`. Amazon S3 non richiede un numero di account o una AWS regione negli ARN.

Se fornisci anche una chiave KMS opzionale, il formato per l'ARN della chiave è `arn:aws:kms:region:account-id:key/key-id`. Per ulteriori informazioni, consulta Amazon Resource Names (ARN) e AWS service namespace. Per crittografare i backup è necessario utilizzare una chiave KMS di crittografia simmetrica. Amazon RDS non supporta le chiavi KMS asimmetriche. Per ulteriori informazioni, consulta Creazione di chiavi KMS di crittografia simmetrica nella Guida per gli sviluppatori di AWS Key Management Service .

> **Note**
>
> Indipendentemente dal fatto che si utilizzi o meno una chiave KMS, le attività native di backup e ripristino abilitano per impostazione predefinita la crittografia AES (Advanced Encryption Standard) a 256 bit sul lato server per i file caricati su S3.

Per istruzioni su come chiamare ciascuna stored procedure, consulta i seguenti argomenti:

- Backup di un database
- Ripristino di un database
- Ripristino di un log
- Completamento di un ripristino del database
- Utilizzo di database parzialmente ripristinati
- Annullamento di un'attività
- Monitoraggio dello stato delle attività

Backup di un database

Per eseguire il backup del tuo database, utilizza la stored procedure `rds_backup_database`.

> **Note**
>
> Non puoi effettuare il backup di un database durante la finestra di manutenzione o quando Amazon RDS è impegnato ad acquisire uno snapshot.

Utilizzo

```
exec msdb.dbo.rds_backup_database
 @source_db_name='database_name',
 @s3_arn_to_backup_to='arn:aws:s3:::bucket_name/file_name.extension',
 [@kms_master_key_arn='arn:aws:kms:region:account-id:key/key-id'],
 [@overwrite_s3_backup_file=0|1],
 [@type='DIFFERENTIAL|FULL'],
 [@number_of_files=n];
```

I parametri seguenti sono obbligatori:

- `@source_db_name` – Nome del database di cui eseguire il backup.
- `@s3_arn_to_backup_to` – L'ARN che indica il bucket Amazon S3 da utilizzare per il backup con il nome del file di backup.

  Il file può avere qualsiasi estensione, ma di norma è `.bak`.

I parametri seguenti sono facoltativi:

- `@kms_master_key_arn`: l'ARN per la chiave KMS di crittografia simmetrica da utilizzare per crittografare l'elemento.
  - Non è possibile utilizzare la chiave di crittografia predefinita. Se si utilizza la chiave predefinita, non verrà eseguito il backup del database.
  - Se non si specifica un identificatore della chiave KMS, il file di backup non verrà crittografato. Per ulteriori informazioni, consulta Crittografia delle risorse Amazon RDS.
  - Quando si specifica una chiave KMS, viene utilizzata la crittografia lato client.

  - Amazon RDS non supporta le chiavi KMS asimmetriche. Per ulteriori informazioni, consulta Creazione di chiavi KMS di crittografia simmetrica nella Guida per gli sviluppatori di AWS Key Management Service .
- `@overwrite_s3_backup_file` – Un valore che indica se sovrascrivere un file di backup esistente.
  - `0` – Il file esistente non viene sovrascritto. Questo è il valore predefinito.

    L'impostazione di `@overwrite_s3_backup_file` su 0 restituisce un errore se il file esiste già.
  - `1` – Il file esistente con il nome specificato viene sovrascritto, anche se non è un file di backup.
- `@type` – Tipo di backup.
  - `DIFFERENTIAL` – Viene eseguito un backup differenziale.
  - `FULL` – Viene eseguito un backup completo. Questo è il valore predefinito.

  Il backup differenziale si basa sull'ultimo backup completo. Per completare i backup differenziali non puoi effettuare una snapshot tra l'ultimo backup completo e il backup differenziale. Per effettuare un backup differenziale ed esiste uno snapshot, effettua un altro backup completo prima di procedere con il backup differenziale.

  Puoi cercare l'ultimo backup completo o snapshot utilizzando la seguente query SQL di esempio.

```
select top 1
database_name
,  backup_start_date
,  backup_finish_date
from    msdb.dbo.backupset
where   database_name='mydatabase'
and     type = 'D'
order by backup_start_date desc;
```

- `@number_of_files` – Numero di file in cui verrà diviso il backup (blocchi). Il numero massimo è 10.
  - Il backup a più file è supportato per backup completi e differenziali.
  - Se immetti un valore pari a 1 o ometti il parametro, viene creato un singolo file di backup.

  Fornisci il prefisso in comune dei file, quindi il suffisso con un asterisco (*). L'asterisco può trovarsi ovunque nella parte *`nome_file`* dell'ARN S3. L'asterisco viene sostituito da una serie di stringhe alfanumeriche nei file generati, a partire da `1-of-`*`number_of_files`*.

Ad esempio, se i nomi dei file nell'ARN S3 sono `backup*.bak` e imposti `@number_of_files=4`, i file di backup generati sono `backup1-of-4.bak`, `backup2-of-4.bak`, `backup3-of-4.bak` e `backup4-of-4.bak`.

- Se uno dei nomi di file esiste già e `@overwrite_s3_backup_file` è 0, viene restituito un errore.
- I backup a più file possono avere un solo asterisco nella parte *`nome_file`* dell'ARN S3.
- I backup a file singolo possono avere un numero qualsiasi di asterischi nella parte *`nome_file`* dell'ARN S3. Gli asterischi non vengono rimossi dal nome file generato.

Esempi

Example di backup differenziale

```
exec msdb.dbo.rds_backup_database
@source_db_name='mydatabase',
@s3_arn_to_backup_to='arn:aws:s3:::mybucket/backup1.bak',
@overwrite_s3_backup_file=1,
@type='DIFFERENTIAL';
```

Example di backup completo con crittografia

```
exec msdb.dbo.rds_backup_database
@source_db_name='mydatabase',
@s3_arn_to_backup_to='arn:aws:s3:::mybucket/backup1.bak',
@kms_master_key_arn='arn:aws:kms:us-east-1:123456789012:key/AKIAIOSFODNN7EXAMPLE',
@overwrite_s3_backup_file=1,
@type='FULL';
```

Example di backup di più file

```
exec msdb.dbo.rds_backup_database
@source_db_name='mydatabase',
@s3_arn_to_backup_to='arn:aws:s3:::mybucket/backup*.bak',
@number_of_files=4;
```

Example di backup differenziale di più file

```
exec msdb.dbo.rds_backup_database
```

```
@source_db_name='mydatabase',
@s3_arn_to_backup_to='arn:aws:s3:::mybucket/backup*.bak',
@type='DIFFERENTIAL',
@number_of_files=4;
```

Example di backup di più file con crittografia

```
exec msdb.dbo.rds_backup_database
@source_db_name='mydatabase',
@s3_arn_to_backup_to='arn:aws:s3:::mybucket/backup*.bak',
@kms_master_key_arn='arn:aws:kms:us-east-1:123456789012:key/AKIAIOSFODNN7EXAMPLE',
@number_of_files=4;
```

Example di backup di più file con sovrascrittura S3

```
exec msdb.dbo.rds_backup_database
@source_db_name='mydatabase',
@s3_arn_to_backup_to='arn:aws:s3:::mybucket/backup*.bak',
@overwrite_s3_backup_file=1,
@number_of_files=4;
```

Example di backup di file singolo con il parametro @number_of_files

Questo esempio genera un file di backup denominato `backup*.bak`.

```
exec msdb.dbo.rds_backup_database
@source_db_name='mydatabase',
@s3_arn_to_backup_to='arn:aws:s3:::mybucket/backup*.bak',
@number_of_files=1;
```

Ripristino di un database

Per eseguire il ripristino del tuo database devi chiamare la stored procedure `rds_restore_database`. Amazon RDS crea uno snapshot iniziale del database dopo che l'attività di ripristino è stata completata e il database è aperto.

Utilizzo

```
exec msdb.dbo.rds_restore_database
 @restore_db_name='database_name',
```

```
@s3_arn_to_restore_from='arn:aws:s3:::bucket_name/file_name.extension',
@with_norecovery=0|1,
[@kms_master_key_arn='arn:aws:kms:region:account-id:key/key-id'],
[@type='DIFFERENTIAL|FULL'];
```

I parametri seguenti sono obbligatori:

- `@restore_db_name` – Il nome del database da ripristinare. I nomi dei database sono univoci. Non è possibile ripristinare un database se esiste già un database con lo stesso nome.
- `@s3_arn_to_restore_from` – L'ARN che indica il prefisso e i nomi Amazon S3 dei file di backup utilizzati per ripristinare il database.
  - Per il backup di un file singolo, fornisci l'intero nome file.
  - Per eseguire un backup a più file, fornisci il prefisso in comune dei file, quindi il suffisso con un asterisco (`*`).
  - Se `@s3_arn_to_restore_from` è vuoto, viene restituito il seguente errore: S3 ARN prefix cannot be empty (Il prefisso ARN di S3 non può essere vuoto).

Il seguente parametro è obbligatorio per i ripristini differenziali, ma facoltativo per i ripristini completi:

- `@with_norecovery` – La clausola di ripristino da utilizzare per l'operazione di ripristino.
  - Impostala su `0` per ripristinare con RECOVERY. In questo caso, il database è online dopo il ripristino.
  - Impostala su `1` per ripristinare con NORECOVERY. In questo caso, il database rimane nello stato RESTORING dopo il completamento dell'attività di ripristino. Con questo approccio puoi eseguire ripristini differenziali successivi.
  - Per i ripristini DIFFERENTIAL, specifica `0` o `1`.
  - Per i ripristini `FULL`, per impostazione predefinita questo valore è `0`.

I parametri seguenti sono facoltativi:

- `@kms_master_key_arn`: la chiave KMS da usare per decrittografare il file di backup, se è stato crittografato.

  Quando si specifica una chiave KMS, viene utilizzata la crittografia lato client.
- `@type` – Tipo di ripristino. I tipi validi sono `DIFFERENTIAL` e `FULL`. Il valore predefinito è `FULL`.

> **Note**
>
> Per i ripristini differenziali, il database deve trovarsi nello stato RESTORING oppure deve già esistere un'attività che ripristina con NORECOVERY.
> Non è possibile ripristinare i backup differenziali successivi mentre il database è online.
> Non è possibile inviare un'attività di ripristino per un database che ha già un'attività di ripristino in sospeso con RECOVERY.
> I ripristini completi con NORECOVERY e i ripristini differenziali non sono supportati nelle istanze Multi-AZ.
> Il ripristino di un database in un'istanza Multi-AZ con repliche di lettura è simile al ripristino di un database in un'istanza Multi-AZ. Non è necessario eseguire ulteriori azioni per ripristinare un database in una replica.

Esempi

Example di ripristino di un singolo file

```
exec msdb.dbo.rds_restore_database
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/backup1.bak';
```

Example di ripristino di più file

Per evitare errori durante il ripristino di più file, assicurati che tutti i file di backup abbiano lo stesso prefisso e che nessun altro file utilizzi tale prefisso.

```
exec msdb.dbo.rds_restore_database
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/backup*';
```

Example di ripristino completo del database con RECOVERY

I seguenti tre esempi eseguono la stessa attività di ripristino completo con RECOVERY.

```
exec msdb.dbo.rds_restore_database
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/backup1.bak';
```

```
exec msdb.dbo.rds_restore_database
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/backup1.bak',
[@type='DIFFERENTIAL|FULL'];
```

```
exec msdb.dbo.rds_restore_database
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/backup1.bak',
@type='FULL',
@with_norecovery=0;
```

Example di ripristino completo del database con crittografia

```
exec msdb.dbo.rds_restore_database
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/backup1.bak',
@kms_master_key_arn='arn:aws:kms:us-east-1:123456789012:key/AKIAIOSFODNN7EXAMPLE';
```

Example di ripristino completo del database con NORECOVERY

```
exec msdb.dbo.rds_restore_database
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/backup1.bak',
@type='FULL',
@with_norecovery=1;
```

Example di ripristino differenziale con NORECOVERY

```
exec msdb.dbo.rds_restore_database
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/backup1.bak',
@type='DIFFERENTIAL',
@with_norecovery=1;
```

Example di ripristino differenziale con RECOVERY

```
exec msdb.dbo.rds_restore_database
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/backup1.bak',
@type='DIFFERENTIAL',
```

```
@with_norecovery=0;
```

Ripristino di un log

Per ripristinare il log chiama la stored procedure `rds_restore_log`.

Utilizzo

```
exec msdb.dbo.rds_restore_log
 @restore_db_name='database_name',
 @s3_arn_to_restore_from='arn:aws:s3:::bucket_name/log_file_name.extension',
 [@kms_master_key_arn='arn:aws:kms:region:account-id:key/key-id'],
 [@with_norecovery=0|1],
 [@stopat='datetime'];
```

I parametri seguenti sono obbligatori:

- `@restore_db_name` – Il nome del database di cui vuoi ripristinare il log.
- `@s3_arn_to_restore_from` – L'ARN che indica il prefisso e i nomi Amazon S3 dei file di log utilizzati per ripristinare il log. Il file può avere qualsiasi estensione, ma di norma è `.trn`.

  Se `@s3_arn_to_restore_from` è vuoto, viene restituito il seguente errore: S3 ARN prefix cannot be empty (Il prefisso ARN di S3 non può essere vuoto).

I parametri seguenti sono facoltativi:

- `@kms_master_key_arn`: la chiave KMS da usare per decrittografare il registro, se è stato crittografato.
- `@with_norecovery` – La clausola di ripristino da utilizzare per l'operazione di ripristino. Il valore predefinito è `1`.
  - Impostala su `0` per ripristinare con RECOVERY. In questo caso, il database è online dopo il ripristino. Non è possibile ripristinare i backup di log successivi mentre il database è online.
  - Impostala su `1` per ripristinare con NORECOVERY. In questo caso, il database rimane nello stato RESTORING dopo il completamento dell'attività di ripristino. Con questo approccio puoi eseguire ripristini di log successivi.
- `@stopat` – Un valore che specifica che lo stato del database viene ripristinato alla data e all'ora specificate (nel formato datetime). Solo i record del log delle transazioni scritti prima della data e dell'ora specificate vengono applicati al database.

Se questo parametro non viene specificato (è NULL), viene ripristinato il log completo.

> **Note**
>
> Per i ripristini di log, il database deve trovarsi nello stato RESTORING oppure deve già esistere un'attività che ripristina con NORECOVERY.
> Non è possibile ripristinare i backup di log mentre il database è online.
> Non è possibile inviare un'attività di ripristino di log per un database che ha già un'attività di ripristino in sospeso con RECOVERY.
> I ripristini di log non sono supportati nelle istanze Multi-AZ.

Esempi

Example di ripristino di log

```
exec msdb.dbo.rds_restore_log
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/mylog.trn';
```

Example di ripristino di log con crittografia

```
exec msdb.dbo.rds_restore_log
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/mylog.trn',
@kms_master_key_arn='arn:aws:kms:us-east-1:123456789012:key/AKIAIOSFODNN7EXAMPLE';
```

Example di ripristino di log con NORECOVERY

I seguenti due esempi eseguono la stessa attività di ripristino di log con NORECOVERY.

```
exec msdb.dbo.rds_restore_log
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/mylog.trn',
@with_norecovery=1;
```

```
exec msdb.dbo.rds_restore_log
@restore_db_name='mydatabase',
```

```
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/mylog.trn';
```

Example di ripristino di log con RECOVERY

```
exec msdb.dbo.rds_restore_log
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/mylog.trn',
@with_norecovery=0;
```

Example di ripristino di log con la clausola STOPAT

```
exec msdb.dbo.rds_restore_log
@restore_db_name='mydatabase',
@s3_arn_to_restore_from='arn:aws:s3:::mybucket/mylog.trn',
@with_norecovery=0,
@stopat='2019-12-01 03:57:09';
```

Completamento di un ripristino del database

Se l'ultima attività di ripristino sul database è stata eseguita utilizzando `@with_norecovery=1`, il database è ora nello stato RESTORING. Apri il database per il normale funzionamento utilizzando la stored procedure `rds_finish_restore`.

Utilizzo

```
exec msdb.dbo.rds_finish_restore @db_name='database_name';
```

> **Note**
>
> Per utilizzare questo approccio, il database deve trovarsi nello stato RESTORING senza attività di ripristino in sospeso.
> La procedura `rds_finish_restore` non è supportata nelle istanze Multi-AZ.
> Per completare il ripristino del database, utilizza l'accesso master. In alternativa, utilizza l'accesso utente che ha ripristinato più di recente il database o il log con NORECOVERY.

Utilizzo di database parzialmente ripristinati

Rimozione di un database parzialmente ripristinato

Per rimuovere un database parzialmente ripristinato (lasciato nello stato RESTORING), utilizza la stored procedure `rds_drop_database`.

```
exec msdb.dbo.rds_drop_database @db_name='database_name';
```

> **Note**
>
> Non è possibile inviare una richiesta DROP per un database che ha già un'attività di ripristino in sospeso o in completamento.
> Per rimuovere il database, utilizza l'accesso master. In alternativa, utilizza l'accesso utente che ha ripristinato più di recente il database o il log con NORECOVERY.

Comportamento di ripristino e point-in-time ripristino delle istantanee per database parzialmente ripristinati

I database parzialmente ripristinati nell'istanza di origine (lasciati nello stato RESTORING) vengono eliminati dall'istanza di destinazione durante il ripristino e il ripristino delle istantanee. point-in-time

Annullamento di un'attività

Per annullare un'attività di backup o ripristino, chiama la stored procedure `rds_cancel_task`.

> **Note**
>
> Non puoi annullare un'attività FINISH_RESTORE.

Utilizzo

```
exec msdb.dbo.rds_cancel_task @task_id=ID_number;
```

Il parametro seguente è obbligatorio:

- `@task_id` – L'ID dell'attività da annullare. L'ID attività si ottiene chiamando `rds_task_status`.

Monitoraggio dello stato delle attività

Per monitorare lo stato delle tue attività di backup e ripristino, chiama la stored procedure `rds_task_status`. Se non fornisci alcun parametro, la stored procedure restituisce lo stato di tutte le attività. Lo stato delle attività viene aggiornato all'incirca ogni due minuti. La cronologia delle operazioni viene conservata per 36 giorni.

Utilizzo

```
exec msdb.dbo.rds_task_status
 [@db_name='database_name'],
 [@task_id=ID_number];
```

I parametri seguenti sono facoltativi:

- `@db_name` – Il nome del database per il quale visualizzare lo stato dell'attività.
- `@task_id` – L'ID dell'attività per la quale visualizzare lo stato.

Esempi

Example di elenco dello stato per un'attività specifica

```
exec msdb.dbo.rds_task_status @task_id=5;
```

Example di elenco dello stato per un'attività e un database specifici

```
exec msdb.dbo.rds_task_status
@db_name='my_database',
@task_id=5;
```

Example di elenco di tutte le attività e relativi stati per un database specifico

```
exec msdb.dbo.rds_task_status @db_name='my_database';
```

Example di elenco di tutte le attività e relativi stati per l'istanza corrente

```
exec msdb.dbo.rds_task_status;
```

Risposta

La stored procedure `rds_task_status` restituisce le seguenti colonne.

Colonna	Descrizione
`task_id`	L'ID dell'attività.
`task_type`	Tipo di attività in base ai parametri di input, come segue: • Per le attività di backup: • BACKUP_DB – Backup di database completo • BACKUP_DB_DIFFERENTIAL – Backup di database differenziale • Per le attività di ripristino: • RESTORE_DB – Ripristino completo del database con RECOVERY • RESTORE_DB_NORECOVERY – Ripristino completo del database con NORECOVERY • RESTORE_DB_DIFFERENTIAL – Ripristino differenziale del database con RECOVERY • RESTORE_DB_DIFFERENTIAL_NORECOVERY – Ripristino differenziale del database con NORECOVERY • RESTORE_DB_LOG – Ripristino del log con RECOVERY • RESTORE_DB_LOG_NORECOVERY – Ripristino del log con NORECOVERY • Per le attività di completamento del ripristino: • FINISH_RESTORE – Completa il ripristino e apre il database

Colonna	Descrizione
	Amazon RDS crea uno snapshot iniziale del database dopo che è stato aperto al completamento delle seguenti attività di ripristino: • RESTORE_DB • RESTORE_DB_DIFFERENTIAL • RESTORE_DB_LOG • FINISH_RESTORE
`database_name`	Il nome del database al quale l'attività è associata.
`% complete`	L'avanzamento dell'attività espresso in percentuale.
`duration (mins)`	La quantità di tempo dedicato all'attività, in minuti.

Colonna	Descrizione
`lifecycle`	Lo stato dell'attività. I possibili stati sono i seguenti: • `CREATED` – Quando chiami `rds_backup_database` o `rds_restore_database`, viene creata un'attività il cui stato viene impostato su `CREATED`. • `IN_PROGRESS` – Dopo l'avvio di un'attività di backup o ripristino, lo stato viene impostato su `IN_PROGRESS`. Possono essere necessari fino a 5 minuti perché lo stato cambi da `CREATED` in `IN_PROGRESS`. • `SUCCESS` – Dopo il completamento di un'attività di backup o ripristino, lo stato viene impostato su `SUCCESS`. • `ERROR` – In caso di errore di un'attività di backup o ripristino, lo stato viene impostato su `ERROR`. Per ulteriori informazioni sull'errore, consulta la colonna `task_info`. • `CANCEL_REQUESTED` – Quando chiami `rds_cancel_task`, lo stato dell'attività viene impostato su `CANCEL_REQUESTED`. • `CANCELLED` – Dopo che un'attività è stata annullata, lo stato dell'attività viene impostato su `CANCELLED`.
`task_info`	Ulteriori informazioni sull'attività. Se si verifica un errore durante il backup o il ripristino di un database, questa colonna contiene informazioni sull'errore. Per l'elenco dei possibili errori e le strategie di mitigazione, consulta Risoluzione dei problemi.
`last_updated`	La data e l'ora dell'ultimo aggiornamento dello stato dell'attività. Lo stato viene aggiornato dopo ogni 5% di avanzamento.
`created_at`	La data e l'ora di creazione dell'attività.
`S3_object_arn`	L'ARN che indica il prefisso e il nome Amazon S3 del file di cui viene eseguito il backup o il ripristino.

Colonna	Descrizione
`overwrite_s3_backup_file`	Il valore del parametro `@overwrite_s3_backup_file` specificato quando chiami un'attività di backup. Per ulteriori informazioni, consulta Backup di un database.
`KMS_master_key_arn`	L'ARN per la chiave KMS utilizzata per la crittografia (per il backup) e la decrittografia (per il ripristino).
`filepath`	Non applicabile ad attività di backup e ripristino nativi.
`overwrite_file`	Non applicabile ad attività di backup e ripristino nativi.

Compressione dei file di backup

Per risparmiare spazio nel tuo bucket Amazon S3 puoi comprimere i tuoi file di backup. Per ulteriori informazioni sulla compressione dei file di backup, consulta Compressione backup nella documentazione di Microsoft.

La compressione dei file di backup è supportata per le seguenti edizioni dei database:

- Microsoft SQL Server Enterprise Edition
- Microsoft SQL Server Standard Edition

Per attivare la compressione per i file di backup è necessario eseguire il seguente codice:

```
exec rdsadmin.dbo.rds_set_configuration 'S3 backup compression', 'true';
```

Per disattivare la compressione per i file di backup è necessario eseguire il seguente codice:

```
exec rdsadmin.dbo.rds_set_configuration 'S3 backup compression', 'false';
```

Risoluzione dei problemi

Di seguito sono elencati i problemi che si potrebbero riscontrare quando si utilizzano backup e ripristino nativi.

Problema	Suggerimenti sulla risoluzione dei problemi
L'opzione di backup/ripristino del database non è ancora abilitata o è in fase di abilitazione. Please try again later. (Per favore, riprova più tardi)	Accertarsi di aver aggiunto l'opzione `SQLSERVER_BACKUP_RESTORE` al gruppo di opzioni DB associato all'istanza database. Per ulteriori informazioni, consulta Aggiunta dell'opzione Native Backup and Restore (Backup nativo e ripristino).
Accesso negato	Il processo di backup o ripristino non può accedere al file di backup. Ciò è in genere dovuto a problemi di questo tipo: • Riferimento al bucket non coretto. Riferimento al bucket utilizzando un formato non coretto. Riferimento a un nome file senza utilizzare l'ARN. • Autorizzazioni non corrette sul file del bucket. Ad esempio, se viene creato da un account diverso che sta tentando di accedervi ora, aggiungere le autorizzazioni corrette. • Una policy IAM incorretta o incompleta. Il ruolo IAM deve includere tutti gli elementi necessari, come, ad esempio, la versione corretta. Il tutto è descritto in Importazione ed esportazione di database SQL Server mediante backup e ripristino nativi.
BACKUP DATABASE WITH COMPRESSION (BACKUP di DATABASE CON COMPRESSIONE) non è supportato nell'edizione <edition_name>	La compressione dei file di backup è supportata solo per Microsoft SQL Server Enterprise Edition e Standard Edition. Per ulteriori informazioni, consulta Compressione dei file di backup.
La chiave <ARN> non esiste	Hai tentato di ripristinare un backup crittografato ma non hai fornito una chiave di crittografia valida. Controlla la chiave di crittografia e riprova. Per ulteriori informazioni, consulta Ripristino di un database.

Problema	Suggerimenti sulla risoluzione dei problemi
Riemetti l'attività con il tipo corretto e la proprietà di sovrascrivere	Se tenti di effettuare il backup del tuo database e indichi il nome di un file che già esiste, ma imposti la proprietà di sovrascrittura su falso, l'operazione di salvataggio non viene completata. Pre risolvere l'errore, indica il nome di un file che non esiste oppure imposta la proprietà di sovrascrittura su vero. Per ulteriori informazioni, consulta Backup di un database. È anche possibile che, nonostante volessi ripristinare il tuo database, hai chiamato la stored procedure `rds_backup_database` per errore. In questo caso, chiama piuttosto la stored procedure `rds_restore_database` . Per ulteriori informazioni, consulta Ripristino di un database. Se avevi l'intenzione di ripristinare il tuo database e hai chiamato la stored procedure `rds_restore_database` , verifica di aver indicato il nome di un file di backup valido. Per ulteriori informazioni, consulta Uso di backup e ripristino nativi.
Specifica un bucket che si trova nella stessa regione dell'istanza di RDS	Non puoi eseguire il backup o il ripristino da un bucket Amazon S3 in una AWS regione diversa dalla tua istanza database Amazon RDS. Puoi utilizzare la replica di Amazon S3 per copiare il file di backup nella regione corretta. AWS Per ulteriori informazioni, consulta la sezione Replica tra regioni nella documentazione di Amazon S3.
Il bucket specificato non esiste.	Verifica di aver indicato l'ARN corretto per il bucket e il file, nel formato corretto. Per ulteriori informazioni, consulta Uso di backup e ripristino nativi.

Problema	Suggerimenti sulla risoluzione dei problemi
L'utente <ARN> non è autorizzato ad eseguire <kms action> sulla risorsa <ARN>	Hai richiesto un'operazione crittografata, ma non hai fornito le autorizzazioni corrette AWS KMS . Verifica di disporre delle autorizzazioni corrette o aggiungile. Per ulteriori informazioni, consulta Configurazione di backup e ripristino nativi.
L'attività di ripristino non è in grado di eseguire il ripristino da più di 10 file di backup). Ridurre il numero di file corrispondenti e riprovare.	Ridurre il numero di file di cui stai provando a effettuare il ripristino. Se necessario, puoi rendere ogni singolo file più grande.
Il database '*nome_database* ' esiste già. Non sono consentiti due database che differiscono solo per maiuscole e minuscole o accento. Scegli un nome di database diverso.	Non è possibile ripristinare un database se esiste già un database con lo stesso nome. I nomi dei database sono univoci.

Importazione ed esportazione di dati SQL Server mediante altri metodi

Di seguito, è possibile trovare informazioni relative all'utilizzo di snapshot per importare i dati Microsoft SQL Server in Amazon RDS. È anche possibile trovare informazioni relative all'utilizzo di snapshot per esportare i dati da un'istanza database RDS che esegue SQL Server.

Se lo scenario lo consente, è più semplice spostare i dati da e in Amazon RDS utilizzando la funzionalità di backup e di ripristino nativa. Per ulteriori informazioni, consulta Importazione ed esportazione di database SQL Server mediante backup e ripristino nativi.

> **Note**
>
> Amazon RDS per Microsoft SQL Server non supporta l'importazione di dati nel database `msdb`.

Importazione di dati in Amazon RDS per SQL Server utilizzando uno snapshot

Per importare dati in un'istanza database SQL Server utilizzando uno snapshot

1. Creare un'Istanza database. Per ulteriori informazioni, consulta Creazione di un'istanza database Amazon RDS.
2. Negare alle applicazioni l'accesso all'istanza database di destinazione.

   Se si impedisce l'accesso all'istanza database durante l'importazione di dati, il trasferimento di dati sarà più veloce. Inoltre, non ci si dovrà preoccupare di eventuali conflitti durante il caricamento dei dati se altre applicazioni non possono scrivere sull'istanza database nello stesso momento. Se si verificano dei problemi ed è necessario eseguire il rollback a uno snapshot di database precedente, le sole modifiche che si perderanno sono i dati importati. Sarà possibile importare nuovamente tali dati dopo la risoluzione del problema.

   Per informazioni sul controllo accessi all'istanza database, consulta Controllo dell'accesso con i gruppi di sicurezza.
3. Creare uno snapshot del database di destinazione.

   Se il database di destinazione è già popolato con dati, è consigliabile creare uno snapshot del database prima di importare i dati. Se si verificano dei problemi nell'importazione dei dati o si desidera eliminare le modifiche, è possibile ripristinare lo stato precedente del database

utilizzando lo snapshot. Per informazioni sugli snapshot di database, consulta Creazione di uno snapshot DB per un'istanza DB Single-AZ.

> **Note**
>
> Quando si crea uno snapshot di database, le operazioni I/O per il database vengono sospese per un momento (millisecondi) durante l'esecuzione del backup.

4. Disabilitare i backup automatici sul database di destinazione.

   La disabilitazione dei backup automatici sull'istanza database di destinazione migliora le prestazioni durante l'importazione dei dati in quanto Amazon RDS non registra le transazioni se i backup automatici sono disabilitati. È tuttavia necessario considerare vari aspetti. I backup automatici sono necessari per eseguire un ripristino point-in-time. Pertanto, non è possibile ripristinare il database a uno specifico point-in-time durante l'importazione dei dati. Inoltre, qualsiasi backup automatico creato sull'istanza database viene eliminato, a meno che non si scelga di conservarlo.

   La scelta di conservare i backup automatici può contribuire a proteggere dall'eliminazione accidentale dei dati. Amazon RDS insieme a ciascun backup automatico salva anche le proprietà dell'istanza database per facilitarne il recupero. L'utilizzo di questa opzione consente di ripristinare un'istanza database eliminata in un momento specifico del periodo di retention dei backup anche dopo averla eliminata. I backup automatici vengono automaticamente eliminati alla fine della finestra di backup specificata, così come accade per un'istanza database attiva.

   È possibile anche utilizzare snapshot precedenti per ripristinare il database e tutti gli snapshot creati restano disponibili. Per informazioni sui backup automatici, consulta Introduzione ai backup.

5. Disabilitare i vincoli di chiave esterna, se applicabile.

   Se è necessario disabilitare tali vincoli, è possibile farlo con lo script seguente.

```
--Disable foreign keys on all tables
    DECLARE @table_name SYSNAME;
    DECLARE @cmd NVARCHAR(MAX);
    DECLARE table_cursor CURSOR FOR SELECT name FROM sys.tables;

    OPEN table_cursor;
    FETCH NEXT FROM table_cursor INTO @table_name;
```

```
    WHILE @@FETCH_STATUS = 0 BEGIN
      SELECT @cmd = 'ALTER TABLE '+QUOTENAME(@table_name)+' NOCHECK CONSTRAINT
 ALL';
      EXEC (@cmd);
      FETCH NEXT FROM table_cursor INTO @table_name;
    END

    CLOSE table_cursor;
    DEALLOCATE table_cursor;

    GO
```

6. Eliminare gli indici, se applicabile.
7. Disabilitare i trigger, se applicabile.

   Se è necessario disabilitare i trigger, è possibile farlo con lo script seguente.

```
--Disable triggers on all tables
    DECLARE @enable BIT = 0;
    DECLARE @trigger SYSNAME;
    DECLARE @table SYSNAME;
    DECLARE @cmd NVARCHAR(MAX);
    DECLARE trigger_cursor CURSOR FOR SELECT trigger_object.name trigger_name,
     table_object.name table_name
    FROM sysobjects trigger_object
    JOIN sysobjects table_object ON trigger_object.parent_obj = table_object.id
    WHERE trigger_object.type = 'TR';

    OPEN trigger_cursor;
    FETCH NEXT FROM trigger_cursor INTO @trigger, @table;

    WHILE @@FETCH_STATUS = 0 BEGIN
      IF @enable = 1
         SET @cmd = 'ENABLE ';
      ELSE
         SET @cmd = 'DISABLE ';

      SET @cmd = @cmd + ' TRIGGER dbo.'+QUOTENAME(@trigger)+' ON
 dbo.'+QUOTENAME(@table)+' ';
      EXEC (@cmd);
      FETCH NEXT FROM trigger_cursor INTO @trigger, @table;
    END
```

```
    CLOSE trigger_cursor;
    DEALLOCATE trigger_cursor;

    GO
```

8. Eseguire una query sull'istanza SQL Server di origine per tutti gli account di accesso che si intende importare nell'istanza database di destinazione.

   SQL Server archivia account di accesso e password nel database `master`. Poiché Amazon RDS non concede l'accesso al database `master`, non è possibile importare direttamente account di accesso e password nell'istanza database di destinazione. È necessario invece eseguire una query sul database `master` sull'istanza SQL Server di origine per generare un file DDL (Data Definition Language). Il file deve includere tutti gli accessi e le password che si desidera aggiungere all'istanza database di destinazione. Il file deve includere anche le appartenenze ai ruoli e le autorizzazioni che si desidera trasferire.

   Per informazioni sull'esecuzione di query sul database `master`, consulta Trasferimento di accessi e password tra istanze di SQL Server 2005 e SQL Server 2008 nella Knowledge Base di Microsoft.

   L'output dello script è un altro script che è possibile eseguire sull'istanza database di destinazione. Il codice dello script nell'articolo della Knowledge Base è il seguente:

   ```
   p.type IN
   ```

   Sostituire ogni occorrenza di `p.type` con il codice seguente:

   ```
   p.type = 'S'
   ```

9. Importare i dati utilizzando il metodo in Importazione dei dati.
10. Concedere alle applicazioni l'accesso all'istanza database di destinazione.

    Al termine dell'importazione dei dati, è possibile concedere l'accesso all'istanza database alle applicazioni bloccate durante l'importazione. Per informazioni sul controllo accessi all'istanza database, consulta Controllo dell'accesso con i gruppi di sicurezza.
11. Abilitare i backup automatici sull'istanza database di destinazione.

    Per informazioni sui backup automatici, consulta Introduzione ai backup.

12. Abilitare i vincoli di chiave esterna.

    Se i vincoli di chiave esterna sono stati disabilitati in precedenza, è possibile abilitarli con lo script seguente.

```
--Enable foreign keys on all tables
    DECLARE @table_name SYSNAME;
    DECLARE @cmd NVARCHAR(MAX);
    DECLARE table_cursor CURSOR FOR SELECT name FROM sys.tables;

    OPEN table_cursor;
    FETCH NEXT FROM table_cursor INTO @table_name;

    WHILE @@FETCH_STATUS = 0 BEGIN
      SELECT @cmd = 'ALTER TABLE '+QUOTENAME(@table_name)+' CHECK CONSTRAINT ALL';
      EXEC (@cmd);
      FETCH NEXT FROM table_cursor INTO @table_name;
    END

    CLOSE table_cursor;
    DEALLOCATE table_cursor;
```

13. Abilitare gli indici, se applicabile.
14. Abilitare i trigger, se applicabile.

    Se i trigger sono stati disabilitati in precedenza, è possibile abilitarli con lo script seguente.

```
--Enable triggers on all tables
    DECLARE @enable BIT = 1;
    DECLARE @trigger SYSNAME;
    DECLARE @table SYSNAME;
    DECLARE @cmd NVARCHAR(MAX);
    DECLARE trigger_cursor CURSOR FOR SELECT trigger_object.name trigger_name,
     table_object.name table_name
    FROM sysobjects trigger_object
    JOIN sysobjects table_object ON trigger_object.parent_obj = table_object.id
    WHERE trigger_object.type = 'TR';

    OPEN trigger_cursor;
    FETCH NEXT FROM trigger_cursor INTO @trigger, @table;

    WHILE @@FETCH_STATUS = 0 BEGIN
      IF @enable = 1
```

```
         SET @cmd = 'ENABLE ';
      ELSE
         SET @cmd = 'DISABLE ';

      SET @cmd = @cmd + ' TRIGGER dbo.'+QUOTENAME(@trigger)+' ON
dbo.'+QUOTENAME(@table)+' ';
      EXEC (@cmd);
      FETCH NEXT FROM trigger_cursor INTO @trigger, @table;
   END

   CLOSE trigger_cursor;
   DEALLOCATE trigger_cursor;
```

Importazione dei dati

Microsoft SQL Server Management Studio è un client SQL Server grafico incluso in tutte le edizioni di Microsoft SQL Server ad eccezione di Express Edition. SQL Server Management Studio Express è fornito da Microsoft come download gratuito. Per trovare questo download, consulta il sito Web di Microsoft.

> **Note**
>
> SQL Server Management Studio è disponibile solo come applicazione basata su Windows.

SQL Server Management Studio include i seguenti strumenti, utili nell'importazione di dati in un'istanza database SQL Server:

- Procedura guidata Genera e pubblica script
- Importazione/Esportazione guidata
- Copia bulk

Procedura guidata Genera e pubblica script

La procedura guidata Genera e pubblica script crea uno script che contiene lo schema di un database, i dati o entrambi. È possibile generare uno script per un database nella distribuzione SQL Server locale. È quindi possibile eseguire lo script per trasferire le informazioni in esso contenute in un'istanza database Amazon RDS.

> **Note**
>
> Per database di dimensioni di 1 GiB o superiori, è più efficace eseguire lo script del solo schema di database. Si utilizza quindi l'Importazione/Esportazione guidata della caratteristica copia bulk di SQL Server per il trasferimento dei dati.

Per informazioni dettagliate sulla procedura guidata Genera e pubblica script, consulta la documentazione di Microsoft SQL Server.

Nella procedura guidata, presta particolare attenzione alle opzioni avanzate nella pagina Opzioni di creazione script per assicurarti che tutti gli elementi da includere nello script siano selezionati. Ad esempio, per impostazione predefinita, i trigger di database non sono inclusi nello script.

Quando lo script viene generato e salvato, puoi utilizzare SQL Server Management Studio per la connessione all'istanza database e quindi eseguire lo script.

Importazione/Esportazione guidata

Importazione/Esportazione guidata crea un pacchetto Integration Services speciale, che puoi utilizzare per copiare i dati dal database SQL Server locale nell'istanza database di destinazione. La procedura guidata può filtrare le tabelle e persino le tuple in una tabella da copiare nell'istanza database di destinazione.

> **Note**
>
> Importazione/Esportazione guidata è una soluzione appropriata per i set di dati di grandi dimensioni, ma forse non la più rapida per esportare dati a distanza dalla distribuzione locale. Una soluzione più rapida è la funzionalità di copia bulk di SQL Server.

Per ulteriori informazioni su Importazione/Esportazione guidata, consulta la documentazione di Microsoft SQL Server.

Nella pagina Choose a Destination (Seleziona destinazione) della procedura guidata, procedere come segue:

- In Nome server, digitare il nome dell'endpoint per l'istanza database.
- Per la modalità di autenticazione del server, scegliere Usa autenticazione di SQL Server.

- Per Nome utente e Password, digitare le credenziali per l'utente master creato per l'istanza database.

Copia bulk

La funzionalità di copia bulk di SQL Server è una soluzione efficace per copiare dati da un database di origine all'istanza database. La copia bulk scrive i dati specificati in un file di dati, ad esempio un file ASCII. In seguito, puoi eseguire la copia bulk per scrivere il contenuto del file sull'istanza database di destinazione.

Questa sezione utilizza l'utilità bcp inclusa in tutte le edizioni di SQL Server. Per informazioni dettagliate sulle operazioni di importazione ed esportazione bulk, consulta la documentazione di Microsoft SQL Server.

> **Note**
>
> Prima di utilizzare la funzionalità di copia bulk, devi dapprima importare lo schema del database nell'istanza database di destinazione. La procedura guidata Genera e pubblica script, descritta precedentemente in questo argomento, è uno strumento eccellente per questo scopo.

Il comando seguente esegue la connessione all'istanza SQL Server locale. Genera un file delimitato da tabulazioni di una tabella specificata nella directory principale C:\ della distribuzione SQL Server esistente. La tabella è specificata dal relativo nome completo e il file di testo ha lo stesso nome della tabella che viene copiata.

```
bcp dbname.schema_name.table_name out C:\table_name.txt -n -S localhost -U username -
P password -b 10000
```

Il codice precedente include le seguenti opzioni:

- `-n` specifica che la copia bulk utilizza i tipi di dati nativi dei dati da copiare.
- `-S` specifica l'istanza SQL Server a si collega l'utilità bcp.
- `-U` specifica il nome utente dell'account utilizzato per il login all'istanza SQL Server.
- `-P` specifica la password per l'utente specificato da `-U`.
- `-b` specifica il numero di righe per batch di dati importati.

> **Note**
>
> Potrebbero esserci altri parametri importanti per la tua operazione di importazione. Ad esempio, potresti avere bisogno del parametro `-E` relativo ai valori di identità. Per ulteriori informazioni, consulta la descrizione completa della sintassi della riga di comando per l'utilità bcp nella documentazione di Microsoft SQL Server.

Ad esempio, supponiamo che un database denominato `store` che utilizza lo schema predefinito dbo contenga una tabella denominata `customers`. L'account utente `admin`, con la password `insecure`, copia 10.000 righe della tabella `customers` in un file denominato `customers.txt`.

```
bcp store.dbo.customers out C:\customers.txt -n -S localhost -U admin -P insecure -b
 10000
```

Dopo aver generato il file di dati, è possibile caricare i dati nell'istanza database utilizzando un comando simile. Prima di procedere, crea il database e lo schema nell'istanza database di destinazione. Utilizza quindi l'argomento `in` per specificare un file di input anziché `out` per specificare un file di output. Invece di utilizzare localhost per specificare l'istanza SQL Server locale, specifica l'endpoint dell'istanza database. Se utilizzi una porta differente dalla porta 1433, specifica anche quella. Il nome utente e la password sono la password e l'utente master dell'istanza database. La sintassi è esposta di seguito.

```
bcp dbname.schema_name.table_name
     in C:\table_name.txt -n -S endpoint,port -U master_user_name -
P master_user_password -b 10000
```

Per continuare l'esempio precedente, supponiamo che il nome utente master sia `admin` e che la password sia `insecure`. L'endpoint per l'istanza database è `rds.ckz2kqd4qsn1.us-east-1.rds.amazonaws.com` e si utilizza la porta 4080. Il comando è il seguente.

```
bcp store.dbo.customers in C:\customers.txt -n -S rds.ckz2kqd4qsn1.us-
east-1.rds.amazonaws.com,4080 -U admin -P insecure -b 10000
```

> **Note**
>
> Specifica una password diversa dal prompt mostrato qui come best practice per la sicurezza.

Esportazione di dati da Amazon RDS per SQL Server

Puoi scegliere una delle seguenti opzioni per esportare dati da un'istanza database RDS for SQL Server:

- Backup di database nativo con file di backup completo (.bak) – L'utilizzo di file .bak per il backup di database è estremamente ottimizzato ed è in genere il modo più rapido per esportare dati. Per ulteriori informazioni, consulta Importazione ed esportazione di database SQL Server mediante backup e ripristino nativi.
- Procedura guidata di importazione ed esportazione di SQL Server – Per ulteriori informazioni, consulta Importazione/Esportazione guidata di SQL Server.
- Procedura guidata di generazione e pubblicazione di script di SQL Server e utilità bcp – Per ulteriori informazioni, consulta Procedura guidata Genera e pubblica script e utilità bcp di SQL Server.

Importazione/Esportazione guidata di SQL Server

Puoi utilizzare Importazione/Esportazione guidata di SQL Server per copiare una o più tabelle, viste o query dall'istanza database RDS for SQL Server in un altro datastore. Questa scelta è la migliore se il datastore di destinazione non è SQL Server. Per ulteriori informazioni, consulta Importazione ed esportazione guidata di SQL Server nella documentazione di SQL Server.

L'Importazione/Esportazione guidata di SQL Server è disponibile come parte di Microsoft SQL Server Management Studio. Il client SQL Server grafico è incluso in tutte le edizioni di Microsoft SQL Server ad eccezione di Express Edition. SQL Server Management Studio è disponibile solo come applicazione basata su Windows. SQL Server Management Studio Express è fornito da Microsoft come download gratuito. Per trovare questo download, consulta il sito Web di Microsoft.

Per esportare dati con Importazione/Esportazione guidata di SQL Server

1. In SQL Server Management Studio, connettersi all'istanza database RDS for SQL Server. Per informazioni dettagliate su come eseguire questa operazione, consulta Connessione a un'istanza database che esegua il motore di database di Microsoft SQL Server.
2. In Esplora oggetti, espandere Database, aprire il menu contestuale (fare clic con il pulsante destro del mouse) del database di origine, scegliere Attività, quindi scegliere Esporta dati. Viene visualizzata la procedura guidata.
3. Nella pagina Seleziona origine dati, procedere come segue:

    a. Per Origine dati scegliere **`SQL Server Native Client 11.0`**.

b. Verifica che la casella Nome server riporti l'endpoint dell'istanza database RDS for SQL Server.

c. Selezionare Usa autenticazione di SQL Server. In Nome utente e Password specifica il nome utente master e la password dell'istanza database.

d. Verificare che nella casella Database sia visualizzato il database da cui esportare i dati.

e. Seleziona Successivo.

4. Nella pagina Seleziona destinazione, procedere come segue:

a. Per Destinazione scegliere **`SQL Server Native Client 11.0`**.

> **Note**
>
> Sono disponibili altre origini dati di destinazione. Queste includono i provider di dati .NET Framework, i provider OLE DB, SQL Server Native Client e ADO.NET, Microsoft Office Excel, Microsoft Office Access e l'origine del file flat. Se si sceglie come destinazione una di queste origini dati, ignorare la parte restante della fase 4. Per i dettagli delle informazioni di connessione da fornire successivamente, consultare l'argomento relativo alla scelta di una destinazione nella documentazione di SQL Server.

b. In Nome server, digitare il nome di server dell'istanza database SQL Server di destinazione.

c. Scegliere il tipo di autenticazione appropriato. Digitare un nome utente e una password, se necessario.

d. Per Database, scegliere il nome del database di destinazione, oppure scegliere Nuovo per creare un nuovo database per i dati esportati.

Se si sceglie Nuovo, consultare Crea database nella documentazione di SQL Server per dettagli sulle informazioni di database da fornire.

e. Seleziona Successivo.

5. Nella pagina Copia tabella o query, scegliere Copia i dati da una o più tabelle o viste oppure Scrivi una query per specificare i dati da trasferire. Seleziona Successivo.

6. Se si sceglie Scrivi una query per specificare i dati da trasferire, viene visualizzata la pagina Impostazione query di origine. Digitare o incollare una query SQL, quindi scegliere Analizza per verificarla. Dopo la convalida della query, scegliere Avanti.

7. Nella pagina Seleziona tabelle e viste di origine, procedere come segue:

a. Selezionare le tabelle e le viste da esportare oppure verificare che la query fornita sia selezionata.

b. Scegliere Modifica mapping e specificare le informazioni di mapping delle colonne e di database. Per ulteriori informazioni, consulta Mapping colonne nella documentazione di SQL Server.

c. (Facoltativo) Per visualizzare un'anteprima dei dati da esportare, selezionare la tabella, la vista o la query, quindi scegliere Anteprima.

d. Seleziona Successivo.

8. Nella pagina Esegui pacchetto, verificare che Esegui immediatamente sia selezionato. Seleziona Successivo.

9. Nella pagina Completamento della procedura guidata, verificare che le informazioni di esportazione dei dati siano quelle previste. Scegli Finish (Fine).

10. Nella pagina Esecuzione completata, scegliere Chiudi.

Procedura guidata Genera e pubblica script e utilità bcp di SQL Server

Puoi utilizzare la procedura guidata Genera e pubblica script di SQL Server per creare script per un intero database o per soltanto alcuni oggetti selezionati. Puoi eseguire questi script su un'istanza database SQL Server di destinazione per ricreare gli oggetti con script. Successivamente, puoi utilizzare l'utilità bcp per esportare in bulk i dati degli oggetti selezionati nell'istanza database di destinazione. Questa scelta è preferibile se intendi spostare un intero database (con oggetti che non siano tabelle) o grandi quantità di dati tra due istanze database SQL Server. Per una descrizione completa della sintassi della riga di comando di bcp, consulta Utilità bcp nella documentazione di Microsoft SQL Server.

La procedura guidata Genera e pubblica script di SQL Server è disponibile come parte di Microsoft SQL Server Management Studio. Il client SQL Server grafico è incluso in tutte le edizioni di Microsoft SQL Server ad eccezione di Express Edition. SQL Server Management Studio è disponibile solo come applicazione basata su Windows. SQL Server Management Studio Express è fornito da Microsoft come download gratuito.

Per esportare dati mediante la procedura guidata Genera e pubblica script e l'utilità bcp di SQL Server

1. In SQL Server Management Studio, connettersi all'istanza database RDS for SQL Server. Per informazioni dettagliate su come eseguire questa operazione, consulta Connessione a un'istanza database che esegua il motore di database di Microsoft SQL Server.
2. In Esplora oggetti, espandere il nodo Database e selezionare il database di cui si intende creare lo script.
3. Seguire le istruzioni in Procedura guidata Genera e pubblica script nella documentazione di SQL Server per creare un file di script.
4. In SQL Server Management Studio, connettersi all'istanza database SQL Server di destinazione.
5. Con l'istanza database SQL Server di destinazione selezionata in Object Explorer (Esplora oggetti), seleziona Open (Apri) dal menu File, quindi seleziona File e apri il file di script.
6. Se si è eseguito lo script dell'intero database, esaminare l'istruzione CREATE DATABASE nello script. Assicurarsi che il database venga creato nella posizione e con i parametri desiderati. Per ulteriori informazioni, consulta CREATE DATABASE nella documentazione di SQL Server.
7. Se si creano utenti di database nello script, verificare se esistono account di accesso server sull'istanza database di destinazione per quegli utenti. In caso contrario, creare degli account di accesso per tali utenti, altrimenti i comandi con script per la creazione di utenti di database non riescono. Per ulteriori informazioni, consulta Creazione di un account di accesso nella documentazione di SQL Server.
8. Scegliere !Execute nel menu Editor SQL per eseguire il file di script e creare oggetti di database. Al termine dello script, verificare che tutti gli oggetti di database esistano come previsto.
9. Utilizza l'utilità bcp per esportare i dati dall'istanza database RDS for SQL Server in file. Aprire un prompt dei comandi e digitare il comando seguente.

   ```
   bcp database_name.schema_name.table_name out data_file -n -S aws_rds_sql_endpoint -
   U username -P password
   ```

   Il codice precedente include le seguenti opzioni:

   - table_name è il nome di una delle tabelle che è stata ricreata nel database di destinazione e che ora si intende popolare con dati.
   - data_file è il percorso completo e il nome del file di dati da creare.
   - `-n` specifica che la copia bulk utilizza i tipi di dati nativi dei dati da copiare.

- `-S` specifica l'istanza database SQL Server da cui eseguire l'esportazione.
- `-U` specifica il nome utente da utilizzare durante la connessione all'istanza database SQL Server.
- `-P` specifica la password per l'utente specificato da `-U`.

Di seguito viene illustrato un esempio del comando .

```
bcp world.dbo.city out C:\Users\JohnDoe\city.dat -n -S sql-jdoe.1234abcd.us-
west-2.rds.amazonaws.com,1433 -U JohnDoe -P ClearTextPassword
```

Ripetere questo passaggio fino ad avere file di dati per tutte le tabelle da esportare.

10. Preparare l'istanza database di destinazione per l'importazione bulk dei dati seguendo le istruzioni contenute in Prepararsi all'importazione bulk dei dati nella documentazione di SQL Server.
11. Decidere il metodo di importazione bulk da utilizzare dopo aver preso in considerazione le prestazioni e altri problemi descritti in Informazioni sulle operazioni di importazione ed esportazione bulk nella documentazione di SQL Server.
12. Importare in blocco i dati dai file di dati creati utilizzando l'utilità bcp. Per fare ciò, seguire le istruzioni contenute in Importare ed esportare dati bulk con bcp oppure nella sezione relativa all'importazione di dati bulk utilizzando BULK INSERT o OPENROWSET(BULK...) nella documentazione di SQL Server, a seconda del metodo di importazione scelto nella fase 11.

Utilizzo di repliche di lettura per Microsoft SQL Server in Amazon RDS

Generalmente, per configurare la replica tra le istanze database di Amazon RDS si utilizzano repliche di lettura. Per informazioni generali sulle repliche di lettura, consulta Uso delle repliche di lettura dell'istanza database.

Questa sezione contiene informazioni specifiche sull'utilizzo delle repliche di lettura su Amazon RDS per SQL Server.

Argomenti

Configurazione delle repliche di lettura per SQL Server

Prima di poter utilizzare un'istanza database come istanza database di origine per la replica, devi abilitare i backup automatici sull'istanza database di origine. Per farlo devi impostare il periodo di retention dei backup su un valore diverso da zero. L'impostazione di questo tipo di implementazione impone anche l'attivazione dei backup automatici.

La creazione di una replica di lettura SQL Server non richiede alcuna interruzione delle attività per l'istanza database primaria. Amazon RDS imposta i parametri e le autorizzazioni necessari per l’istanza database di origine e la replica di lettura senza interruzione del servizio. Viene acquisito uno snapshot dell'istanza database di origine e tale snapshot diventa la replica di lettura. Quando una replica di lettura viene eliminata non si verifica alcuna interruzione.

È possibile creare fino a 15 repliche di lettura da un'istanza database di origine. Per un efficace funzionamento della replica, raccomandiamo di configurare ciascuna replica di lettura con la stessa quantità di risorse di calcolo e di archiviazione dell'istanza database di origine. Se si dimensiona l'istanza database di origine, si devono dimensionare anche le repliche di lettura.

La versione del motore del database SQL Server dell'istanza database di origine e tutte le relative repliche di lettura devono essere uguali. Amazon RDS aggiorna la primaria immediatamente dopo

l'aggiornamento delle repliche di lettura, a prescindere dalla finestra di manutenzione. Per ulteriori informazioni sull'aggiornamento della versione del motore del database, consultare Aggiornamento del motore di database Microsoft SQL Server.

Una replica di lettura deve disporre di risorse di calcolo e storage sufficienti per ricevere e applicare le modifiche provenienti dall'origine. Se una replica di lettura raggiunge la massima capacità di risorse di calcolo, di rete o di storage, smette di ricevere o applicare modifiche dalla sua origine. Puoi modificare le risorse di storage e CPU di una replica di lettura in modo indipendente dalla sua origine e dalle altre repliche di lettura.

Limitazioni per le repliche di lettura con SQL Server

Le seguenti limitazioni si applicano alle repliche di lettura di SQL Server su Amazon RDS:

- Le repliche di lettura sono disponibili solo sul motore SQL Server Enterprise Edition (EE).
- Le repliche di lettura sono disponibili per le versioni di SQL Server 2016-2022.
- È possibile creare fino a 15 repliche di lettura da un'istanza database di origine. La replica potrebbe subire ritardi quando l'istanza DB di origine ha più di 5 repliche di lettura.
- Le repliche di lettura sono disponibili solo per le istanze database in esecuzione sulle classi di istanze database con quattro o più vCPU.
- Una replica di lettura supporta fino a 100 database a seconda del tipo di classe dell'istanza e della modalità di disponibilità. È necessario creare database sull'istanza DB di origine per replicarli automaticamente nelle repliche di lettura. Non è possibile scegliere singoli database da replicare. Per ulteriori informazioni, consulta Restrizioni per le istanze database di Microsoft SQL Server.
- Non è possibile eliminare un database da una replica di lettura. Per eliminare un database, eliminalo dall'istanza DB di origine con la `rds_drop_database` stored procedure. Per ulteriori informazioni, consulta Rimozione di un database Microsoft SQL Server.
- Se l'istanza DB di origine utilizza Transparent Data Encryption (TDE) per crittografare i dati, anche la replica di lettura configura automaticamente TDE.

  Se l'istanza DB di origine utilizza una chiave KMS per crittografare i dati, le repliche di lettura nella stessa regione utilizzano la stessa chiave KMS. Per le repliche di lettura tra regioni, è necessario specificare una chiave KMS dall'area della replica di lettura al momento della creazione della replica di lettura. Non è possibile modificare la chiave KMS per una replica di lettura.
- Le repliche di lettura hanno lo stesso fuso orario e le stesse regole di confronto dell'istanza DB di origine, indipendentemente dalla zona di disponibilità in cui vengono create.

- Le repliche di lettura sono disponibili solo per le istanze database in esecuzione sulle classi di istanze database con quattro o più vCPU.
- Il supporto per le operazioni seguenti non è disponibile su Amazon RDS per SQL Server:
  - Conservazione di backup delle repliche di lettura
  - oint-in-time Ripristino IP da repliche di lettura
  - Snapshot manuali di repliche di lettura
  - Repliche di lettura AZ multiple
  - Creazione di repliche di lettura di repliche di lettura
  - Sincronizzazione degli accessi degli utenti a repliche di lettura
- Amazon RDS per SQL Server non interviene per attenuare un elevato ritardo di replica tra un'istanza database di origine e le sue repliche di lettura. Assicurati che l'istanza database di origine e le sue repliche di lettura siano dimensionate correttamente, in termini di capacità di calcolo e storage, per adattarsi al loro carico operativo.
- È possibile eseguire la replica tra le regioni AWS GovCloud (Stati Uniti orientali) e AWS GovCloud (Stati Uniti occidentali), ma non all'interno o all'esterno. AWS GovCloud (US) Regions

Considerazioni sulle opzioni per le repliche RDS per SQL Server

Prima di creare una replica RDS per SQL Server, considera i requisiti, le restrizioni e i consigli seguenti:

- Se la replica SQL Server si trova nella stessa regione dell'istanza database di origine, assicurati che appartenga allo stesso gruppo di opzioni dell'istanza database di origine. Le modifiche al gruppo di opzioni di origine o all'appartenenza al gruppo di opzioni di origine si propagano alle repliche. Queste modifiche vengono applicate alle repliche immediatamente dopo l'applicazione all'istanza database di origine, indipendentemente dalla finestra di manutenzione delle repliche.

  Per ulteriori informazioni sui gruppi di opzioni, consulta Uso di gruppi di opzioni.
- Quando crei una replica tra regioni SQL Server, Amazon RDS crea un gruppo di opzioni dedicato.

  Non puoi rimuovere una replica tra regioni SQL Server dal suo gruppo di opzioni dedicato. Nessun'altra istanza database può usare il gruppo di opzioni dedicato per una replica tra regioni SQL Server.

Di seguito sono riportate le opzioni replicate. Per aggiungere opzioni replicate a una replica in più regioni SQL Server, aggiungile al gruppo di opzioni dell'istanza database di origine. L'opzione è installata anche su tutte le repliche dell'istanza database di origine.

- `TDE`

Di seguito sono riportate le opzioni non replicate. È possibile aggiungere o rimuovere le opzioni non replicate da un gruppo di opzioni dedicato.

- `MSDTC`
- `SQLSERVER_AUDIT`
- Per abilitare l'opzione `SQLSERVER_AUDIT` sulla replica di lettura tra regioni, aggiungi l'opzione `SQLSERVER_AUDIT` al gruppo di opzioni dedicato sulla replica di lettura tra regioni e sul gruppo di opzioni dell'istanza di origine. Aggiungendo l'opzione `SQLSERVER_AUDIT` all'istanza di origine della replica di lettura tra regioni SQL Server, è possibile creare l'oggetto di controllo a livello di server e le specifiche di controllo a livello di server su ciascuna delle repliche di lettura tra regioni dell'istanza di origine. Per consentire alle repliche di lettura tra regioni di caricare i log di controllo completati su un bucket Amazon S3, aggiungi l'opzione `SQLSERVER_AUDIT` al gruppo di opzioni dedicato e configura le impostazioni delle opzioni. Il bucket Amazon S3 che stai utilizzando come destinazione dei file del controllo deve trovarsi nella stessa regione della replica di lettura tra regioni. Puoi modificare l'impostazione dell'opzione `SQLSERVER_AUDIT` per ogni replica di lettura tra regioni in modo indipendente affinché ciascuna possa accedere a un bucket Amazon S3 nella rispettiva regione.

Le opzioni seguenti non sono supportate per le repliche di lettura tra regioni.

- `SSRS`
- `SSAS`
- `SSIS`

Le opzioni seguenti sono parzialmente supportate per le repliche di lettura tra regioni.

- `SQLSERVER_BACKUP_RESTORE`
- L'istanza database di origine di una replica tra regioni SQL Server può avere l'opzione `SQLSERVER_BACKUP_RESTORE`, ma non è possibile eseguire ripristini nativi sull'istanza database di origine finché non si eliminano tutte le relative repliche tra regioni. Qualsiasi attività di ripristino nativo esistente verrà annullata durante la creazione di una replica tra regioni. Non puoi aggiungere l'opzione `SQLSERVER_BACKUP_RESTORE` a un gruppo di opzioni dedicato.

Per ulteriori informazioni su backup e ripristino nativi, consulta Importazione ed esportazione di database SQL Server mediante backup e ripristino nativi.

Quando promuovi una replica di lettura tra regioni SQL Server, tale replica si comporta come qualsiasi altra istanza database SQL Server, compresa la gestione delle opzioni. Per ulteriori informazioni sui gruppi di opzioni, consulta Uso di gruppi di opzioni.

Sincronizzazione degli utenti e degli oggetti del database con una replica di lettura SQL Server

Tutti gli accessi, i ruoli server personalizzati, i processi di SQL Agent o altri oggetti a livello di server presenti nell'istanza database primaria al momento della creazione di una replica di lettura dovrebbero essere presenti nella replica di lettura appena creata. Tuttavia, tutti gli oggetti a livello di server creati nell'istanza database primaria dopo la creazione della replica di lettura non verranno replicati automaticamente ed è necessario crearli manualmente nella replica di lettura.

Gli utenti del database vengono replicati automaticamente dall'istanza database primaria nella replica di lettura. Poiché il database delle repliche di lettura è in modalità di sola lettura, l'identificatore di sicurezza (SID) dell'utente del database non può essere aggiornato nel database. Pertanto, quando si creano accessi SQL nella replica di lettura, è essenziale assicurarsi che il SID di tale accesso corrisponda al SID dell'accesso SQL corrispondente nell'istanza database primaria. Se non vengono sincronizzati, i SID degli accessi SQL non saranno in grado di accedere al database nella replica di lettura. Gli accessi autenticati di Windows Active Directory (AD) non presentano questo problema perché SQL Server ottiene il SID da Active Directory.

Sincronizzazione di un accesso SQL tra istanza database primaria e replica di lettura

1. Eseguire la connessione all'istanza database primaria.
2. Creare un nuovo accesso SQL nell'istanza database primaria.

```
USE [master]
GO
CREATE LOGIN TestLogin1
WITH PASSWORD = 'REPLACE WITH PASSWORD';
```

> **Note**
>
> Specifica una password diversa dal prompt mostrato qui come best practice per la sicurezza.

3. Creare un nuovo utente del database per l'accesso SQL nel database.

```
USE [REPLACE WITH YOUR DB NAME]
GO
CREATE USER TestLogin1 FOR LOGIN TestLogin1;
GO
```

4. Controllare il SID dell'accesso SQL appena creato nell'istanza database primaria.

```
SELECT name, sid FROM sys.server_principals WHERE name =  TestLogin1;
```

5. Connettersi alla replica di lettura. Creare il nuovo accesso SQL.

```
CREATE LOGIN TestLogin1 WITH PASSWORD = 'REPLACE WITH PASSWORD', SID=[REPLACE WITH
 sid FROM STEP #4];
```

In alternativa, se si dispone dell'accesso al database delle repliche di lettura, è possibile correggere l'utente orfano come segue:

1. Connettersi alla replica di lettura.
2. Individuare gli utenti orfani nel database.

```
USE [REPLACE WITH YOUR DB NAME]
GO
EXEC sp_change_users_login 'Report';
GO
```

3. Creare un nuovo accesso SQL per l'utente orfano del database.

```
CREATE LOGIN TestLogin1 WITH PASSWORD = 'REPLACE WITH PASSWORD', SID=[REPLACE WITH
 sid FROM STEP #2];
```

   Esempio:

```
CREATE LOGIN TestLogin1 WITH PASSWORD = 'TestPa$$word#1',
 SID=[0x1A2B3C4D5E6F7G8H9I0J1K2L3M4N5O6P];
```

> **Note**
>
> Specifica una password diversa dal prompt mostrato qui come best practice per la sicurezza.

Risoluzione dei problemi relativi a una replica di lettura SQL Server

Puoi monitorare il ritardo di replica in Amazon CloudWatch visualizzando la metrica Amazon RDS `ReplicaLag`. Per ulteriori informazioni sui ritardi della replica, consultare Monitoraggio della replica di lettura.

Se il ritardo della replica è eccessivamente lungo, puoi utilizzare la seguenti query per ottenere informazioni sul ritardo.

```
SELECT AR.replica_server_name
     , DB_NAME (ARS.database_id) 'database_name'
     , AR.availability_mode_desc
     , ARS.synchronization_health_desc
     , ARS.last_hardened_lsn
     , ARS.last_redone_lsn
     , ARS.secondary_lag_seconds
FROM sys.dm_hadr_database_replica_states ARS
INNER JOIN sys.availability_replicas AR ON ARS.replica_id = AR.replica_id
--WHERE DB_NAME(ARS.database_id) = 'database_name'
ORDER BY AR.replica_server_name;
```

Implementazioni Multi-AZ per Amazon RDS for Microsoft SQL Server

Le implementazioni Multi-AZ forniscono alle istanze database maggior disponibilità, longevità dei dati e tolleranza ai guasti. In caso di manutenzione pianificata del database o interruzione non pianificata del servizio, Amazon RDS esegue automaticamente il failover sull'istanza DB up-to-date secondaria. Questa funzionalità consente alle operazioni del database di riprendere velocemente senza intervento manuale. Le istanze primarie e di standby usano lo stesso endpoint, il cui indirizzo di rete fisico passa alla replica secondaria come parte del processo di failover. Non è necessario riconfigurare l'applicazione quando si verifica un failover.

Amazon RDS supporta le implementazioni Multi-AZ per Microsoft SQL Server tramite il mirroring del database (DBM) di SQL Server o i gruppi di disponibilità (AG) Always On. Amazon RDS monitora e mantiene lo stato della tua implementazione Multi-AZ. Se si verificano problemi, RDS ripara automaticamente le istanze database non salutari, ristabilisce la sincronizzazione e avvia i failover. Il failover ha luogo solo se le istanze di standby e primarie non sono completamente sincronizzate. Non è necessario gestire nulla.

Quando configuri il Multi-AZ per SQL Server, RDS configura automaticamente tutti i database sull'istanza per utilizzare i gruppi di disponibilità o il mirroring di database. Amazon RDS gestisce l'istanza primaria, l’istanza witness e l'istanza di standby per tuo conto. Poiché la configurazione è automatica, RDS seleziona DBM o Always On AGs in base alla versione di SQL Server distribuita.

Amazon RDS supporta Multi-AZ con Always On AGs per le seguenti versioni ed edizioni di SQL Server:

- SQL Server 2022:
  - Standard Edition
  - Enterprise Edition
- SQL Server 2019:
  - Standard Edition 15.00.4073.23 e successive
  - Enterprise Edition
- SQL Server 2017:
  - Standard Edition 14.00.3401.7 e successive
  - Enterprise Edition 14.00.3049.1 e successive
- SQL Server 2016: Enterprise Edition 13.00.5216.0 e successive

Amazon RDS supporta Multi-AZ con DBM per le seguenti versioni ed edizioni di SQL Server, tranne le versioni di Enterprise Edition annotate in precedenza:

- SQL Server 2019: Standard Edition 15.00.4043.16
- SQL Server 2017: edizioni Standard ed Enterprise
- SQL Server 2016: edizioni Standard ed Enterprise
- SQL Server 2014: edizioni Standard ed Enterprise

Per determinare se l'istanza database di SQL Server è Single-AZ, Multi-AZ con DBM o Multi-AZ con gruppi di disponibilità Always On, puoi utilizzare la seguente query SQL:

```
SELECT CASE WHEN dm.mirroring_state_desc IS NOT NULL THEN 'Multi-AZ (Mirroring)'
    WHEN dhdrs.group_database_id IS NOT NULL THEN 'Multi-AZ (AlwaysOn)'
    ELSE 'Single-AZ'
    END 'high_availability'
FROM sys.databases sd
LEFT JOIN sys.database_mirroring dm ON sd.database_id = dm.database_id
LEFT JOIN sys.dm_hadr_database_replica_states dhdrs ON sd.database_id =
 dhdrs.database_id AND dhdrs.is_local = 1
WHERE DB_NAME(sd.database_id) = 'rdsadmin';
```

L'output è simile a quello riportato di seguito.

```
high_availability
Multi-AZ (AlwaysOn)
```

Aggiunta di Multi-AZ a un'istanza database di Microsoft SQL Server

Quando si crea una nuova istanza DB di SQL Server utilizzando AWS Management Console, è possibile aggiungere Multi-AZ con Database Mirroring (DBM) o Always On AGs. A tale scopo, scegli Yes (Mirroring / Always On) (Sì (Mirroring/Always On)) in Multi-AZ deployment (Implementazione Multi-AZ). Per ulteriori informazioni, consulta Creazione di un'istanza database Amazon RDS.

Quando modifichi un'istanza database di SQL Server esistente usando la console, puoi aggiungere Multi-AZ con i gruppi di disponibilità o il mirroring di database selezionando Yes (Mirroring / Always On) (Sì (Mirroring/Always On)) in Multi-AZ deployment (Implementazione Multi-AZ) nella pagina Modify DB instance (Modifica istanza database). Per ulteriori informazioni, consulta Modifica di un'istanza database Amazon RDS.

> **Note**
>
> Se l'istanza database esegue il mirroring del database, non gruppi di disponibilità Always On (AG), potrebbe essere necessario disabilitare l'ottimizzazione in memoria prima di aggiungere Multi-AZ. Disabilitare l'ottimizzazione in memoria con DBM prima di aggiungere Multi-AZ se l'istanza database esegue SQL Server 2014, 2016 o 2017 Enterprise Edition e l'ottimizzazione in memoria è abilitata.
> Se l'istanza database esegue AGS, non richiede questa fase.

Rimozione di Multi-AZ da un'istanza database Microsoft SQL Server

Quando si modifica un'istanza DB di SQL Server esistente utilizzando AWS Management Console, è possibile rimuovere Multi-AZ con DBM o AGs. A tale scopo, scegli No (Mirroring / Always On) No (mirroring/Always on) in Multi-AZ deployment (Implementazione Multi-AZ) nella pagina Modify DB Instance (Modifica istanza database). Per ulteriori informazioni, consulta Modifica di un'istanza database Amazon RDS.

Limitazioni, note e suggerimenti per l'implementazione di Multi-AZ di Microsoft SQL Server

Di seguito sono riportate alcune limitazioni quando si utilizzano le implementazioni Multi-AZ su istanze database RDS per SQL Server:

- Multi-AZ tra regioni non è supportata.
- L'arresto di un'istanza database RDS per SQL Server in un'implementazione multi-AZ non è supportato.
- Non è possibile configurare l'istanza database secondaria per accettare attività di lettura del database.
- Multi-AZ con i gruppi di disponibilità Always On supporta l'ottimizzazione in memoria.
- Multi-AZ con gruppi di disponibilità (AG) Always On non supporta l'autenticazione Kerberos per il listener del gruppo di disponibilità. Questo perché il listener non dispone di un Service Principal Name (SPN).
- Non è possibile rinominare un database su un'istanza database di SQL Server presente in un'implementazione Multi-AZ di SQL Server. Se è necessario rinominare un database su tale

istanza, disattiva prima Multi-AZ per l'istanza database, quindi rinomina il database. Infine, attiva di nuovo Multi-AZ per l'istanza database.

- Si possono ripristinare solo le istanze database Multi-AZ di cui è stato eseguito il backup con il modello di recupero Full.
- Le implementazioni multi-AZ hanno un limite di 10.000 processi di SQL Server Agent.

  Se hai bisogno di un limite più alto, richiedi un aumento contattando. AWS Support Aprire la pagina del Centro di supporto AWS Support effettuando l'accesso se necessario, quindi selezionare Crea caso. Selezionare Service limit increase (Aumento limiti del servizio). Compilare e inviare il modulo.

Di seguito sono riportate alcune note quando si utilizzano le implementazioni Multi-AZ su istanze database RDS per SQL Server:

- Amazon RDS mostra l'endpoint listener del gruppo di disponibilità Always On AGs. L'endpoint è visibile nella console e viene restituito dall'operazione API `DescribeDBInstances` come una voce nel campo endpoint.
- Amazon RDS supporta i failover di sottoreti multiple del gruppo di disponibilità.
- Per utilizzare la funzionalità Multi-AZ di SQL Server con un'istanza database SQL Server in un cloud privato virtuale (VPC), crea innanzitutto un gruppo di sottoreti DB con sottoreti in almeno due distinte zone di disponibilità. Quindi assegnare il gruppo di sottoreti DB alla replica primaria dell'istanza database di SQL Server.
- Quando un'istanza database viene modificata e impostata come implementazione Multi-AZ, durante la modifica il relativo stato è Modifica in corso. Amazon RDS crea l'istanza standby ed esegue un backup dell'istanza database principale. Dopo aver completato il processo, lo stato dell'istanza database primaria diventa available (disponibile).
- Le implementazioni Multi-AZ gestiscono tutti i database sullo stesso nodo. Se un database nell'host primario esegue il failover, tutti i database di SQL Server eseguono il failover come un'unica unità atomica nell'host di standby. Amazon RDS effettua il provisioning di un nuovo host integro e sostituisce l'host non integro.
- Multi-AZ con DBM o AG supporta una replica di standby singola.
- Utenti, accessi e autorizzazioni vengono automaticamente replicati sulla versione secondaria. Non è necessario ricrearli. I ruoli del server definiti dall'utente vengono replicati solo nelle istanze database che utilizzano i gruppi di Always On per implementazioni Multi-AZ.
- Nelle implementazioni Multi-AZ, RDS per SQL Server crea accessi a SQL Server per consentire Always On AGS o il mirroring del database. RDS crea accessi con lo schema

seguente,, e. db_<dbiResourceId>_node1_login db_<dbiResourceId>_node2_login db_<dbiResourceId>_witness_login

- RDS per SQL Server crea un accesso a SQL Server per consentire l'accesso alla lettura delle repliche. RDS crea un accesso con lo schema seguente,. db_<readreplica_dbiResourceId>_node_login
- Nelle distribuzioni Multi-AZ, i processi di SQL Server Agent vengono replicati dall'host principale all'host secondario quando la funzionalità di replica del processo è attivata. Per ulteriori informazioni, consulta Attivazione della replica di processo SQL Server Agent.
- In un'implementazione standard dell'istanza database in una zona di disponibilità singola, è possibile osservare un aumento della latenza a causa dalle replica sincrona dei dati.
- I tempi di failover sono influenzati dal tempo necessario per completare il processo di recupero. Le transazioni di grandi dimensioni aumentano il tempo di failover.
- Nelle distribuzioni di SQL Server Multi-AZ, riavviare con failover, riavvia solo l'istanza database primaria. Dopo aver eseguito il failover, l'istanza database primaria diventa la nuova istanza database secondaria. I parametri potrebbero non essere aggiornati per istanze Multi-AZ. Per il riavvio senza failover, le istanze database primarie e secondarie vengono riavviate e i parametri vengono aggiornati dopo il riavvio. Se l'istanza database non risponde, si consiglia di riavviare senza failover.

Di seguito sono riportate alcune raccomandazioni quando si utilizzano le implementazioni Multi-AZ su RDS per le istanze database di Microsoft SQL Server:

- Per database utilizzati in produzione o in preproduzione, raccomandiamo quanto segue:
  - Implementazioni Multi-AZ per l'alta disponibilità
  - "Provisioned IOPS" per prestazioni veloci e coerenti
  - "Memoria ottimizzata" anziché "Uso generale"
- Non è possibile selezionare la zona di disponibilità per l'istanza secondaria, quindi tieni questo a mente quando distribuisci gli host delle applicazioni. Il tuo database potrebbe non riuscire su un'altra zona di disponibilità e gli host delle applicazioni potrebbero non essere nella stessa zona di disponibilità del database. Per questo motivo, consigliamo di bilanciare gli host delle applicazioni tra tutte le AZ della regione specificata AWS .
- Per prestazioni ottimali, non abilitare i gruppi di disponibilità Always On o il mirroring di database durante un'operazione di caricamento di dati di grandi dimensioni. Se vuoi caricare i dati il più

rapidamente possibile, termina il caricamento dei dati prima di convertire l'istanza database in un'implementazione Multi-AZ.

- Le applicazioni che accedono ai database SQL Server devono avere una gestione delle eccezioni che rileva gli errori di connessione. Il seguente esempio di codice mostra un blocco try/catch che rileva un errore di comunicazione. In questo esempio, l'istruzione `break` esce dal ciclo `while` se la connessione ha esito positivo, ma ritenta fino a 10 volte se viene generata un'eccezione.

```
int RetryMaxAttempts = 10;
int RetryIntervalPeriodInSeconds = 1;
int iRetryCount = 0;
while (iRetryCount < RetryMaxAttempts)
{
   using (SqlConnection connection = new SqlConnection(DatabaseConnString))
   {
      using (SqlCommand command = connection.CreateCommand())
      {
         command.CommandText = "INSERT INTO SOME_TABLE VALUES ('SomeValue');";
         try
         {
            connection.Open();
            command.ExecuteNonQuery();
            break;
         }
         catch (Exception ex)
         {
            Logger(ex.Message);
            iRetryCount++;
         }
         finally {
            connection.Close();
         }
      }
   }
   Thread.Sleep(RetryIntervalPeriodInSeconds * 1000);
}
```

- Non utilizzare il comando `Set Partner Off` quando lavori con le istanze Multi-AZ. Ad esempio, non effettuare le seguenti operazioni.

```
--Don't do this
ALTER DATABASE db1 SET PARTNER off
```

- Non impostare la modalità di ripristino su `simple`. Ad esempio, non effettuare le seguenti operazioni.

```
--Don't do this
ALTER DATABASE db1 SET RECOVERY simple
```

- Non utilizzare il parametro `DEFAULT_DATABASE` quando crei nuovi accessi sulle istanze database Multi-AZ poiché queste impostazioni non possono essere applicate al mirroring di standby. Ad esempio, non effettuare le seguenti operazioni.

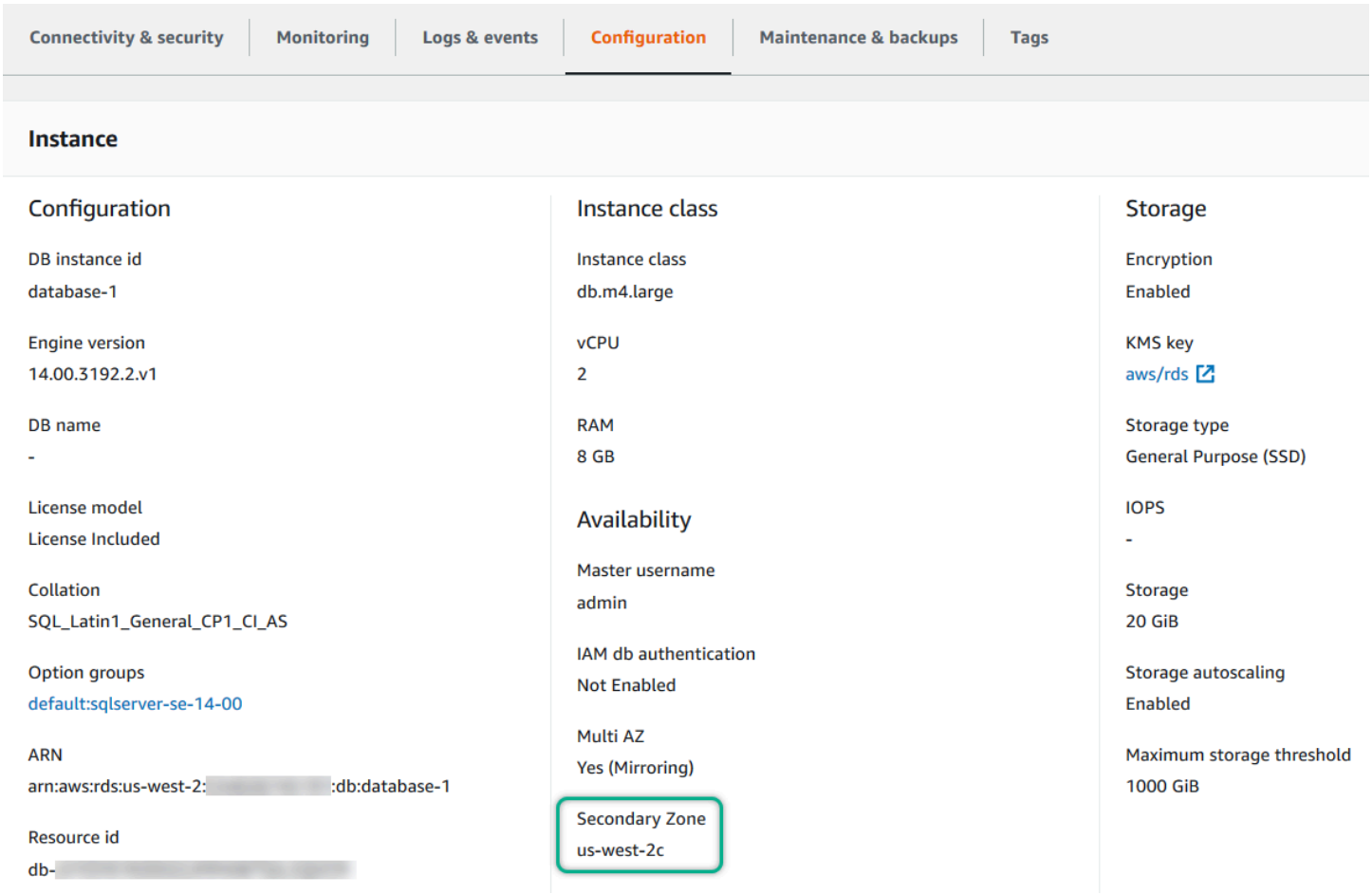
```
--Don't do this
CREATE LOGIN [test_dba] WITH PASSWORD=foo, DEFAULT_DATABASE=[db2]
```

  Inoltre, non fare quanto segue.

```
--Don't do this
ALTER LOGIN [test_dba] SET DEFAULT_DATABASE=[db3]
```

Determinazione della posizione della versione secondaria

Puoi determinare la posizione della replica secondaria usando la AWS Management Console. È necessario conoscere la posizione della versione secondaria se imposti l'istanza database primaria in un VPC.



È inoltre possibile visualizzare la zona di disponibilità del secondario utilizzando il AWS CLI comando `describe-db-instances` o l'operazione API RDS. `DescribeDBInstances` L'output mostra la zona di disponibilità secondaria in cui si trova il mirroring di standby.

Migrazione di mirroring di database a gruppi di disponibilità Always On

Nella versione 14.00.3049.1 di Microsoft SQL Server Enterprise Edition, i gruppi di disponibilità Always On sono abilitati per impostazione predefinita.

Per migrare il mirroring di database a gruppi di disponibilità Always On, verifica innanzitutto la versione di cui disponi. Se utilizzi un'istanza database con una versione precedente a Enterprise Edition 13.00.5216.0, modifica l'istanza per applicarla alla patch 13.00.5216.0 o successive. Se utilizzi un'istanza database con una versione precedente a Enterprise Edition 14.00.3049.1, modifica l'istanza per applicarla alla patch 14.00.3049.1 o successive.

Se desideri aggiornare un'istanza database sottoposta a mirroring per utilizzare i gruppi di disponibilità, esegui prima l'aggiornamento, modifica l'istanza per rimuovere Multi-AZ e quindi

modificala nuovamente per aggiungere Multi-AZ. Ciò converte la tua istanza per usare i gruppi di disponibilità Always On.

Funzionalità opzionali per Microsoft SQL Server su Amazon RDS

Nelle sezioni di seguito sono disponibili informazioni sull'aumento delle istanze Amazon RDS che eseguono il motore database di Microsoft SQL Server.

Argomenti

Utilizzo di SSL con un'istanza database Microsoft SQL Server

Ora puoi utilizzare Secure Sockets Layer (SSL) per crittografare le connessioni tra le applicazioni client e le istanze database Amazon RDS che eseguono Microsoft SQL Server. Il supporto per SSL è disponibile in tutte le regioni AWS per tutte le edizioni di SQL Server supportate.

Quando crei un'istanza database di SQL Server, Amazon RDS crea un certificato SSL per l'istanza. Il certificato SSL include l'endpoint dell'istanza database come nome comune (CN) per il certificato SSL per la protezione contro attacchi di spoofing.

Vi sono due modi per utilizzare SSL per connettersi all'istanza database di SQL Server:

- Forzare SSL per tutte le connessioni — ciò avviene in modo trasparente per il client e il client non deve effettuare alcuna operazione per utilizzare SSL.
- Crittografare connessioni specifiche — viene configurata una connessione SSL da un computer client specifico e devi eseguire alcune operazioni nel client per crittografare le connessioni.

Per informazioni sul supporto di Transport Layer Security (TLS) per SQL Server, consulta l'argomento relativo al supporto di TLS 1.2 per Microsoft SQL Server.

Imposizione dell'utilizzo di SSL per le connessioni all'istanza database

Puoi forzare l'utilizzo di SSL per tutte le connessioni all'istanza database. Se forzi l'utilizzo di SSL per le connessioni, ciò avviene in modo trasparente per il client e il client non deve effettuare alcuna operazione per utilizzare SSL.

Per forzare l'utilizzo di SSL, utilizza il parametro `rds.force_ssl`. Per impostazione predefinita, il parametro `rds.force_ssl` è impostato su `0 (off)`. Imposta il parametro `rds.force_ssl` su `1 (on)` per forzare l'utilizzo di SSL per le connessioni. Il parametro `rds.force_ssl` è statico, quindi dopo aver modificato il valore devi riavviare l'istanza database per rendere effettiva la modifica.

Per forzare l'utilizzo di SSL per tutte le connessioni all'istanza database

1. Determinare il gruppo di parametri collegato all'istanza database:
    a. Accedi alla AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
    b. Nell'angolo in alto a destra della console Amazon RDS, scegliere la regione AWS dell'istanza database.

c. Nel riquadro di navigazione, scegliere Databases (Database) quindi scegliere il nome dell'istanza database per mostrarne i dettagli.

d. Scegliere la scheda Configuration (Configurazione). Individuare il campo Gruppo di parametri nella sezione.

2. Se necessario, creare un nuovo gruppo di parametri. Se l'istanza database utilizza il gruppo di parametri predefinito, è necessario creare un nuovo gruppo di parametri. Se l'istanza database utilizza un gruppo di parametri non predefinito, è possibile scegliere di modificare il gruppo di parametri esistente oppure di creare un nuovo gruppo di parametri. Se si modifica un gruppo di parametri esistente, la modifica interessa tutte le istanze database che utilizzano tale gruppo di parametri.

   Per creare un nuovo gruppo di parametri, seguire le istruzioni in Creazione di un gruppo di parametri del database.

3. Modificare il gruppo di parametri nuovo o esistente per impostare il parametro `rds.force_ssl` su `true`. Per modificare un gruppo di parametri, seguire le istruzioni in Modifica di parametri in un gruppo di parametri del database.

4. Se è stato creato un nuovo gruppo di parametri, modificare l'istanza database per collegare il nuovo gruppo di parametri. Modificare l'impostazione DB Parameter Group (Gruppo di parametri database) dell'istanza database. Per ulteriori informazioni, consulta Modifica di un'istanza database Amazon RDS.

5. Riavviare l'istanza database. Per ulteriori informazioni, consulta Riavvio di un'istanza database.

Crittografia di connessioni specifiche

Puoi forzare l'utilizzo di SSL per tutte le connessioni all'istanza database oppure puoi crittografare le connessioni solo da computer client specifici. Per utilizzare SSL da un client specifico, devi ottenere i certificati per il computer client, importare i certificati nel computer client e quindi crittografare le connessioni dal computer client.

> **Note**
>
> Tutte le istanze di SQL Server create dopo il 5 agosto 2014 utilizzando l'endpoint dell'istanza database nel campo relativo al nome comune del certificato SSL. Prima del 5 agosto 2014, la verifica dei certificati SSL non era disponibile per le istanze di SQL Server basate su VPC. Se disponi di un'istanza database di SQL Server basata su VPC creata prima del 5 agosto 2014 e desideri utilizzare la verifica dei certificati SSL e garantire che l'endpoint

dell'istanza sia incluso come nome comune per il certificato SSL per tale istanza database, rinomina l'istanza. Quando rinomini un'istanza database, viene distribuito un nuovo certificato e l'istanza viene riavviata per abilitare il nuovo certificato.

Recupero di certificati per i computer client

Per crittografare le connessioni da un computer client a un'istanza database Amazon RDS che esegue Microsoft SQL Server, è necessario un certificato nel computer client.

Per ottenere il certificato, scaricalo nel computer client. Puoi scaricare un certificato root che funziona per tutte le regioni. Puoi anche scaricare un bundle di certificati che contiene i certificati root sia nuovi che precedenti. Inoltre, puoi scaricare certificati intermedi specifici della regione. Per ulteriori informazioni sul download, consulta .

Dopo aver scaricato il certificato appropriato, importalo nel sistema operativo Microsoft Windows seguendo la procedura nella sezione successiva.

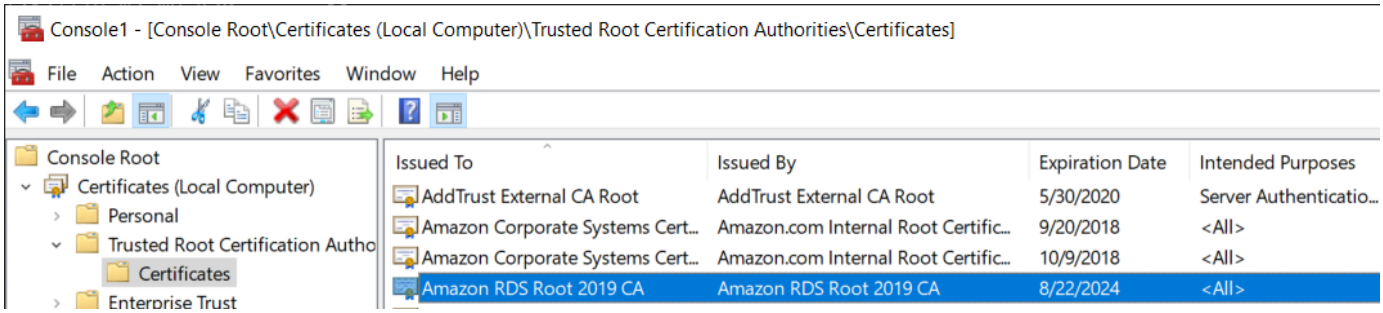
Importazione di certificati nei computer client

Puoi utilizzare la procedura seguente per importare il certificato nel sistema operativo Microsoft Windows nel computer client.

Per importare il certificato nel sistema operativo Windows:

1. Nel menu Start digitare **`Run`** nella casella di ricerca e premere INVIO.
2. Nella casella Apri digitare **`MMC`** e quindi scegliere OK.
3. Nella console MMC scegliere Aggiungi/Rimuovi snap-in dal menu File.
4. Nella finestra di dialogo Aggiungi o rimuovi snap-in, per Snap-in disponibili selezionare **`Certificates`** e quindi scegliere Avanti.
5. Nella finestra di dialogo Snap-in certificati scegliere Account del computer e quindi Avanti.
6. Nella finestra di dialogo Seleziona computer scegliere Fine.
7. Nella finestra di dialogo Aggiungi o rimuovi snap-in scegliere OK.
8. Nella console MMC espandere Certificati, aprire il menu contestuale (clic con il pulsante destro del mouse) per Autorità di certificazione radice attendibili, scegliere Tutte le attività e quindi scegliere Importa.
9. Nella prima pagina dell'Importazione guidata certificati scegliere Avanti.

10. Nella seconda pagina dell'Importazione guidata certificati scegliere Sfoglia. Nella finestra del browser modificare il tipo di file scegliendo Tutti i file (*.*), perché .pem non è un'estensione standard per i certificati. Individuare il file con estensione .pem scaricato in precedenza.
11. Scegliere Apri per selezionare il file del certificato e quindi scegliere Avanti.
12. Nella terza pagina dell'Importazione guidata certificati scegliere Avanti.
13. Nella quarta pagina dell'Importazione guidata certificati scegliere Fine. Viene visualizzata una finestra di dialogo che indica che l'importazione è riuscita.
14. Nella console MMC espandere Certificati, espandere Autorità di certificazione radice attendibili e quindi scegliere Certificati. Individuare il certificato per verificarne l'esistenza, come illustrato di seguito.



Crittografia di connessioni a un'istanza database Amazon RDS che esegue Microsoft SQL Server

Dopo avere importato un certificato nel computer client, è possibile crittografare le connessioni dal computer client a un'istanza database Amazon RDS che esegue Microsoft SQL Server.

Per SQL Server Management Studio, attieniti alla procedura seguente. Per ulteriori informazioni su SQL Server Management Studio, consulta Utilizzare SQL Server Management Studio.

Per crittografare le connessioni da SQL Server Management Studio

1. Avviare SQL Server Management Studio.
2. Per Connetti al server digitare le informazioni del server, il nome utente di accesso e la password.
3. Scegliere Opzioni.
4. Selezionare Crittografa connessione.
5. Scegliere Connetti.
6. Verificare che la connessione sia crittografata eseguendo la query seguente. Verificare che la query restituisca `true` per `encrypt_option`.

```
select ENCRYPT_OPTION from SYS.DM_EXEC_CONNECTIONS where SESSION_ID = @@SPID
```

Per qualsiasi altro client SQL, attieniti alla procedura seguente.

Per crittografare le connessioni da altri client SQL

1. Aggiungere `encrypt=true` alla stringa di connessione. La stringa deve essere disponibile come opzione o come proprietà nella pagina della connessione negli strumenti dell'interfaccia utente grafica.

   > **Note**
   >
   > Per abilitare la crittografia SSL per i client che si connettono utilizzando JDBC, può essere necessario aggiungere il certificato SQL Amazon RDS all'archivio di certificati CA (cacerts) Java. A tale scopo, è possibile utilizzare l'utilità keytool.

2. Verificare che la connessione sia crittografata eseguendo la query seguente. Verificare che la query restituisca `true` per `encrypt_option`.

   ```
   select ENCRYPT_OPTION from SYS.DM_EXEC_CONNECTIONS where SESSION_ID = @@SPID
   ```

Configurazione dei protocolli di protezione e dei cifrari

È possibile attivare e disattivare determinati protocolli di sicurezza e cifrari utilizzando i parametri DB. I parametri di protezione che è possibile configurare (ad eccezione della versione 1.2 di TLS) sono riportati nella tabella seguente.

Parametro DB	Valori consentiti (impostazione predefinita in grassetto)	Descrizione
rds.tls10	Impostazione predefinita, abilitato, disabilitato	TLS 1.0.
rds.tls11	Impostazione predefinita, abilitato, disabilitato	TLS 1.1.
rds.tls12	default	TLS 1.2. Non è possibile modificare questo valore.
rds.fips	0, 1	Quando si imposta il parametro su 1, RDS impone l'uso di moduli conformi allo standard Federal Information Processing Standard (FIPS) 140-2. Per ulteriori informazioni, consulta Use SQL Server 2016 in FIPS 140-2-compliant mode (Utilizza SQL Server 2016 in modalità conforme a FIPS 140-2) nella documenta zione Microsoft.
rds.rc4	Impostazione predefinita, abilitato, disabilitato	Cifratura flusso RC4.
rds.diffie-hellman	Impostazione predefinita, abilitato, disabilitato	Crittografia dello scambio chiavi Diffie-Hellman.

Parametro DB	Valori consentiti (impostazione predefinita in grassetto)	Descrizione
rossi. diffie-hellman-min-key-lunghezza del bit	Impostazione predefinita, 1024, 2048, 4096	Lunghezza minima del bit per le chiavi Diffie-Hellman.
rds.curve25519	Impostazione predefinita, abilitato, disabilitato	Crittografia Curve25519 a curva ellittica. Questo parametro non è supportat o per tutte le versioni del motore.
rds.3des168	Impostazione predefinita, abilitato, disabilitato	Crittografia DES (Triple Data Encryption Standard) con una lunghezza di chiave a 168 bit.

> **Note**
>
> Per le versioni secondarie del motore successive a 16.00.4120.1, 15.00.4365.2, 14.00.3465.1, 13.00.6435.1 e 12.00.6449.1, l'impostazione predefinita per i parametri del DB,, ed è disabilitata. *`rds.tls10` `rds.tls11` `rds.rc4` `rds.curve25519` `rds.3des168`* Altrimenti l'impostazione predefinita è abilitata.
> Per le versioni minori del motore successive a 16.00.4120.1, 15.00.4365.2, 14.00.3465.1, 13.00.6435.1 e 12.00.6449.1, l'impostazione predefinita per è 3072. `rds.diffie-hellman-min-key-bit-length` Altrimenti l'impostazione predefinita è 2048.

Per configurare i protocolli e i cifrari di sicurezza, attenersi alla procedura descritta di seguito.

1. Creare un gruppo di parametri DB personalizzato.
2. Modificare i parametri nel gruppo di parametri.
3. Associare il gruppo di parametri DB all'istanza database.

Per ulteriori informazioni sui gruppi di parametri database, consulta Utilizzo di gruppi di parametri.

Creazione del gruppo di parametri relativi alla sicurezza

Creare un gruppo di parametri per i parametri relativi alla sicurezza che corrisponde all'edizione di SQL Server e alla versione dell'istanza database.

Console

Nella procedura seguente viene creato un gruppo di parametri per SQL Server Standard Edition 2016.

Per creare il gruppo di parametri

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel pannello di navigazione, scegliParameter groups (Gruppi di parametri).
3. Scegliere Create parameter group (Crea gruppo di parametri).
4. Nel riquadro Create parameter group (Crea gruppi di parametri), procedi nel modo seguente:
   a. Per Famiglia del gruppo di parametri, scegliere sqlserver-se-13.0.
   b. Per Group name (Nome gruppo), immettere un identificatore per il gruppo di parametri, ad esempio **`sqlserver-ciphers-se-13`**.
   c. Per Description (Descrizione), immettere **`Parameter group for security protocols and ciphers`**.
5. Scegliere Create (Crea).

CLI

Nella procedura seguente viene creato un gruppo di parametri per SQL Server Standard Edition 2016.

Per creare il gruppo di parametri

- Eseguire uno dei seguenti comandi.

  Example

  Per LinuxmacOS, oUnix:

```
aws rds create-db-parameter-group \
```

```
    --db-parameter-group-name sqlserver-ciphers-se-13 \
    --db-parameter-group-family "sqlserver-se-13.0" \
    --description "Parameter group for security protocols and ciphers"
```

Per Windows:

```
aws rds create-db-parameter-group ^
    --db-parameter-group-name sqlserver-ciphers-se-13 ^
    --db-parameter-group-family "sqlserver-se-13.0" ^
    --description "Parameter group for security protocols and ciphers"
```

Modifica dei parametri relativi alla sicurezza

Modificare i parametri relativi alla sicurezza nel gruppo di parametri che corrisponde all'edizione di SQL Server e alla versione dell'istanza database.

Console

Nella procedura seguente, il gruppo di parametri creato per SQL Server Standard Edition 2016 viene modificato. In questo esempio viene disattivata la versione 1.0 di TLS.

Per modificare il gruppo di parametri

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione scegliere Parameter groups (Gruppi di parametri).
3. Scegli il gruppo di parametri, ad esempio sqlserver-ciphers-se-13.
4. In Parameters (Parametri), filtrare l'elenco dei parametri per **`rds`**.
5. Scegliere Edit parameters (Modifica parametri).
6. Scegliere rds.tls10.
7. Per Values (Valori), scegliere Disabled (Disabilitato).
8. Seleziona Save changes (Salva modifiche).

CLI

Nella procedura seguente, il gruppo di parametri creato per SQL Server Standard Edition 2016 viene modificato. In questo esempio viene disattivata la versione 1.0 di TLS.

Per modificare il gruppo di parametri

- Eseguire uno dei seguenti comandi.

  Example

  Per LinuxmacOS, oUnix:

```
aws rds modify-db-parameter-group \
    --db-parameter-group-name sqlserver-ciphers-se-13 \
    --parameters
 "ParameterName='rds.tls10',ParameterValue='disabled',ApplyMethod=pending-reboot"
```

  Per Windows:

```
aws rds modify-db-parameter-group ^
    --db-parameter-group-name sqlserver-ciphers-se-13 ^
    --parameters
 "ParameterName='rds.tls10',ParameterValue='disabled',ApplyMethod=pending-reboot"
```

Associazione del gruppo di parametri relativi alla sicurezza all'istanza database

Per associare il gruppo di parametri alla tua istanza DB, usa AWS Management Console o il AWS CLI.

Console

È possibile associare il gruppo di parametri a un'istanza database nuova o esistente:

- Per una nuova istanza database, associarli all'avvio dell'istanza. Per ulteriori informazioni, consulta Creazione di un'istanza database Amazon RDS.
- Per un'istanza database esistente, associarli modificando l'istanza. Per ulteriori informazioni, consulta Modifica di un'istanza database Amazon RDS.

CLI

È possibile associare il gruppo di parametri a un'istanza database nuova o esistente:

Per creare un'istanza database con il gruppo di parametri

- Specificare lo stesso tipo di motore di database e la versione principale utilizzati durante la creazione del gruppo di parametri.

  Example

  Per LinuxmacOS, oUnix:

```
aws rds create-db-instance \
    --db-instance-identifier mydbinstance \
    --db-instance-class db.m5.2xlarge \
    --engine sqlserver-se \
    --engine-version 13.00.5426.0.v1 \
    --allocated-storage 100 \
    --master-user-password secret123 \
    --master-username admin \
    --storage-type gp2 \
    --license-model li \
    --db-parameter-group-name sqlserver-ciphers-se-13
```

  Per Windows:

```
aws rds create-db-instance ^
    --db-instance-identifier mydbinstance ^
    --db-instance-class db.m5.2xlarge ^
    --engine sqlserver-se ^
    --engine-version 13.00.5426.0.v1 ^
    --allocated-storage 100 ^
    --master-user-password secret123 ^
    --master-username admin ^
    --storage-type gp2 ^
    --license-model li ^
    --db-parameter-group-name sqlserver-ciphers-se-13
```

  > **Note**
  >
  > Specifica una password diversa dal prompt mostrato qui come best practice per la sicurezza.

Per modificare un'istanza database e associare il gruppo di parametri

- Eseguire uno dei seguenti comandi.

  Example

  Per LinuxmacOS, oUnix:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --db-parameter-group-name sqlserver-ciphers-se-13 \
    --apply-immediately
```

  Per Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --db-parameter-group-name sqlserver-ciphers-se-13 ^
    --apply-immediately
```

Integrazione di un'istanza database Amazon RDS for SQL Server con Amazon S3

Puoi trasferire i file tra un'istanza database che esegue Amazon RDS for SQL Server e un bucket Amazon S3. In questo modo, puoi utilizzare Amazon S3 con le caratteristiche di SQL Server, come BULK INSERT. Ad esempio, puoi scaricare file .csv, .xml, .txt e altri da Amazon S3 nell'host dell'istanza database e importare i dati da `D:\S3\` nel database. Tutti i file vengono archiviati in `D:\S3\` nell'istanza database.

Si applicano le limitazioni seguenti:

- I file nella cartella `D:\S3` vengono eliminati nella replica di standby dopo un failover su istanze Multi-AZ. Per ulteriori informazioni, consulta Limitazioni Multi-AZ per l'integrazione S3.
- L'istanza database e il bucket S3 devono trovarsi nella stessa regione AWS.
- Se si eseguono più attività di integrazione S3 alla volta, le attività vengono eseguite in sequenza, non in parallelo.

> **Note**
>
> Le attività di integrazione S3 condividono la stessa coda delle attività di backup e ripristino native. In questa coda possono essere presenti al massimo due attività in esecuzione in qualsiasi momento. Di conseguenza, due attività di backup e ripristino native in esecuzione bloccheranno tutte le attività di integrazione S3.

- Dovrai riabilitare la caratteristica di integrazione di S3 nelle istanze ripristinate. L'integrazione S3 non viene propagata dall'istanza di origine all'istanza ripristinata. I file in `D:\S3` in un'istanza ripristinata vengono eliminati.
- Il download nell'istanza database è limitato a 100 file. In altre parole, non possono essere presenti più di 100 file in `D:\S3\`.
- Per il download sono supportati solo i file senza estensioni di file o con le seguenti estensioni di file: .abf, .asdatabase, .bcp, .configsettings, .csv, .dat, .deploymentoptions, .deploymenttargets, .fmt, .info, e .xmla.
- Il bucket S3 deve avere lo stesso proprietario del ruolo AWS Identity and Access Management (IAM) correlato. Pertanto, l'integrazione tra account S3 non è supportata.
- Il bucket S3 non può essere aperto al pubblico.
- Le dimensioni dei file per i caricamenti da RDS a S3 sono limitate a 50 GB per file.

- La dimensione dei file per i download da S3 a RDS è limitata al massimo supportato da S3.

Argomenti

Per ulteriori informazioni sull'utilizzo dei file in Amazon S3, consulta Nozioni di base su Amazon Simple Storage Service.

Prerequisiti per l'integrazione di RDS for SQL Server con S3

Prima di iniziare, trova o crea il bucket S3 che desideri utilizzare. Quindi, aggiungi le autorizzazioni in modo che l'istanza database RDS possa accedere al bucket S3. Per configurare questo accesso, crea una policy e un ruolo IAM.

Console

Per creare una policy IAM per accedere ad Amazon S3

1. Nel riquadro di navigazione della console di gestione IAM, scegliere Policies (Policy).
2. Creare una nuova policy e usare la scheda Visual editor (Editor visivo) per le seguenti fasi.
3. Per Service (Servizio), immettere **S3** e scegliere il servizio S3.
4. Per Actions (Operazioni), scegliere le seguenti opzioni per concedere l'accesso richiesto dall'istanza database:

    - `ListAllMyBuckets` – obbligatorio
    - `ListBucket` – obbligatorio
    - `GetBucketACL` – obbligatorio

- `GetBucketLocation` – obbligatorio
- `GetObject` – obbligatorio per il download dei file da S3 a `D:\S3\`
- `PutObject` – obbligatorio per il caricamento dei file da `D:\S3\` a S3
- `ListMultipartUploadParts` – obbligatorio per il caricamento dei file da `D:\S3\` a S3
- `AbortMultipartUpload` – obbligatorio per il caricamento dei file da `D:\S3\` a S3

5. Per Resources (Risorse), le opzioni visualizzate dipendono dalle operazioni scelte nella fase precedente. Potrebbero essere visualizzate le opzioni per bucket, object (oggetto) o entrambi. Per ognuna di queste opzioni, aggiungere l'Amazon Resource Name (ARN) appropriato.

   Per bucket, aggiungere l'ARN per il bucket che si desidera utilizzare. Ad esempio, se il bucket è denominato `example-bucket`, impostare l'ARN su `arn:aws:s3:::example-bucket`.

   Per object (oggetto), immettere l'ARN per il bucket e scegliere una delle opzioni seguenti:

   - Per concedere l'accesso a tutti i file nel bucket specificato, selezionare Any (Qualsiasi) per Bucket name (Nome bucket) e Object name (Nome oggetto).
   - Per concedere l'accesso a cartelle o file specifici del bucket, fornire gli ARN per i bucket e gli oggetti specifici a cui si desidera che SQL Server acceda.

6. Seguire le istruzioni indicate nella console fino al termine della creazione della policy.

   Le precedenti sono indicazioni generali per la creazione di una policy. Per istruzioni più dettagliate sulla creazione delle policy IAM, consultare Creazione di policy IAM nella Guida per l'utente di IAM.

Per creare un ruolo IAM che utilizza la policy IAM della procedura precedente

1. Nel riquadro di navigazione della console di gestione IAM, scegliere Roles (Ruoli).
2. Creare un nuovo ruolo IAM e scegliere le seguenti opzioni quando vengono visualizzate nella console:

   - Servizio AWS
   - RDS
   - RDS – Add Role to Database (RDS – Aggiungi ruolo al database)

   Scegliere Next: Permissions (Successivo: Autorizzazioni) nella parte inferiore dello schermo.

3. Per Attach permissions policies (Collega policy di autorizzazioni), immettere il nome della policy IAM precedentemente creata. Scegliere quindi la policy dall'elenco.
4. Seguire le istruzioni indicate nella console fino al termine della creazione del ruolo.

   Le precedenti sono indicazioni generali per la configurazione di un ruolo. Per istruzioni più dettagliate sulla creazione dei ruoli, consultare Ruoli IAM nella Guida per l'utente di IAM.

AWS CLI

Per concedere ad Amazon RDS l'accesso a un bucket Amazon S3, utilizza la seguente procedura:

1. Creare una policy IAM che conceda ad Amazon RDS l'accesso a un bucket S3.
2. Creare un ruolo IAM che Amazon RDS può assumere per conto dell'utente per accedere ai bucket S3.

   Per ulteriori informazioni, consulta la pagina relativa alla creazione di un ruolo per delegare le autorizzazioni a un utente IAM nella Guida per l'utente IAM.
3. Collegare la policy IAM al ruolo IAM creato.

Co,e creare la policy IAM

Includere le operazioni appropriate per garantire l'accesso richiesto dall'istanza database:

- `ListAllMyBuckets` – obbligatorio
- `ListBucket` – obbligatorio
- `GetBucketACL` – obbligatorio
- `GetBucketLocation` – obbligatorio
- `GetObject` – obbligatorio per il download dei file da S3 a `D:\S3\`
- `PutObject` – obbligatorio per il caricamento dei file da `D:\S3\` a S3
- `ListMultipartUploadParts` – obbligatorio per il caricamento dei file da `D:\S3\` a S3
- `AbortMultipartUpload` – obbligatorio per il caricamento dei file da `D:\S3\` a S3

1. Il seguente comando dell'AWS CLI crea una policy IAM denominata `rds-s3-integration-policy` con queste opzioni. Concede l'accesso a un bucket denominato `bucket_name`.

Example

Per LinuxmacOS, oUnix:

```
aws iam create-policy \
  --policy-name rds-s3-integration-policy \
  --policy-document '{
        "Version": "2012-10-17",
        "Statement": [
            {
                "Effect": "Allow",
                "Action": "s3:ListAllMyBuckets",
                "Resource": "*"
            },
            {
                "Effect": "Allow",
                "Action": [
                    "s3:ListBucket",
                    "s3:GetBucketACL",
                    "s3:GetBucketLocation"
                ],
                "Resource": "arn:aws:s3:::bucket_name"
            },
            {
                "Effect": "Allow",
                "Action": [
                    "s3:GetObject",
                    "s3:PutObject",
                    "s3:ListMultipartUploadParts",
                    "s3:AbortMultipartUpload"
                ],
                "Resource": "arn:aws:s3:::bucket_name/key_prefix/*"
            }
        ]
    }'
```

Per Windows:

Assicurarsi di cambiare i caratteri di fine riga con quelli supportati dall'interfaccia in uso (^ al posto di \). Inoltre, in Windows, è necessario applicare a tutte le doppie virgolette il carattere di

escape \. Per evitare l'uso del carattere di escape per le virgolette in JSON, è possibile salvarlo in un file e passarlo come parametro.

Per prima cosa creare il file `policy.json` con la seguente policy di autorizzazione:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "s3:ListAllMyBuckets",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:ListBucket",
                "s3:GetBucketACL",
                "s3:GetBucketLocation"
            ],
            "Resource": "arn:aws:s3:::bucket_name"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:PutObject",
                "s3:ListMultipartUploadParts",
                "s3:AbortMultipartUpload"
            ],
            "Resource": "arn:aws:s3:::bucket_name/key_prefix/*"
        }
    ]
}
```

Usare il comando seguente per creare la policy:

```
aws iam create-policy ^
     --policy-name rds-s3-integration-policy ^
     --policy-document file://file_path/assume_role_policy.json
```

2. Dopo aver creato la policy, annotarne l'Amazon Resource Name (ARN). L'ARN sarà necessario in una fase successiva.

Per creare il ruolo IAM

- Il seguente comando AWS CLI crea il ruolo IAM `rds-s3-integration-role` a questo proposito.

  Example

  Per LinuxmacOS, oUnix:

```
aws iam create-role \
    --role-name rds-s3-integration-role \
    --assume-role-policy-document '{
      "Version": "2012-10-17",
      "Statement": [
        {
          "Effect": "Allow",
          "Principal": {
             "Service": "rds.amazonaws.com"
           },
          "Action": "sts:AssumeRole"
        }
      ]
    }'
```

  Per Windows:

  Assicurarsi di cambiare i caratteri di fine riga con quelli supportati dall'interfaccia in uso (^ al posto di \). Inoltre, in Windows, è necessario applicare a tutte le doppie virgolette il carattere di escape \. Per evitare l'uso del carattere di escape per le virgolette in JSON, è possibile salvarlo in un file e passarlo come parametro.

  Per prima cosa creare il file `assume_role_policy.json` con la seguente policy:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
```

```
            "Principal": {
                "Service": [
                    "rds.amazonaws.com"
                ]
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

Usare il comando seguente per creare il ruolo IAM:

```
aws iam create-role ^
     --role-name rds-s3-integration-role ^
     --assume-role-policy-document file://file_path/assume_role_policy.json
```

Example di utilizzare la chiave del contesto delle condizioni globale per creare il ruolo IAM

Si consiglia di utilizzare le chiavi di contesto delle condizioni globali `aws:SourceArn` e `aws:SourceAccount` nelle policy basate sulle risorse per limitare le autorizzazioni del servizio a una risorsa specifica. Questo è il modo più efficace per proteggersi dal problema di deputy confused.

Puoi usare le chiavi di contesto delle condizioni globali e avere il valore `aws:SourceArn` che contiene l'ID dell'account. In questo caso, il valore `aws:SourceAccount` e l'account nel valore `aws:SourceArn` deve utilizzare lo stesso ID account quando viene utilizzato nella stessa istruzione della policy.

- Utilizzare `aws:SourceArn` se si desidera un accesso cross-service per una singola risorsa.
- Utilizzare `aws:SourceAccount` se si desidera consentire l'associazione di qualsiasi risorsa in tale account all'uso cross-service.

Nella policy, assicurati di utilizzare la chiave di contesto della condizione globale `aws:SourceArn` con l'Amazon Resource Name (ARN) completo delle risorse che accedono al ruolo. Per l'integrazione con S3, assicurati di includere gli ARN dell'istanza database, come mostrato nell'esempio seguente.

Per LinuxmacOS, oUnix:

```
aws iam create-role \
    --role-name rds-s3-integration-role \
    --assume-role-policy-document '{
      "Version": "2012-10-17",
      "Statement": [
        {
          "Effect": "Allow",
          "Principal": {
             "Service": "rds.amazonaws.com"
           },
          "Action": "sts:AssumeRole",
                "Condition": {
                    "StringEquals": {

 "aws:SourceArn":"arn:aws:rds:Region:my_account_ID:db:db_instance_identifier"
                    }
                }
        }
      ]
    }'
```

Per Windows:

Aggiungi la chiave di contesto delle condizioni globali a `assume_role_policy.json`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "rds.amazonaws.com"
                ]
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {

 "aws:SourceArn":"arn:aws:rds:Region:my_account_ID:db:db_instance_identifier"
                }
            }
```

```
        }
    ]
}
```

Per allegare la policy IAM al ruolo IAM

- Il seguente comando dell'AWS CLI collega la policy al ruolo denominato `rds-s3-integration-role`. Sostituire *`your-policy-arn`* con l'ARN della policy annotato nel passaggio precedente.

  Example

  Per LinuxmacOS, oUnix:

  ```
  aws iam attach-role-policy \
      --policy-arn your-policy-arn \
      --role-name rds-s3-integration-role
  ```

  Per Windows:

  ```
  aws iam attach-role-policy ^
      --policy-arn your-policy-arn ^
      --role-name rds-s3-integration-role
  ```

Abilitazione dell'integrazione di RDS for SQL Server con S3

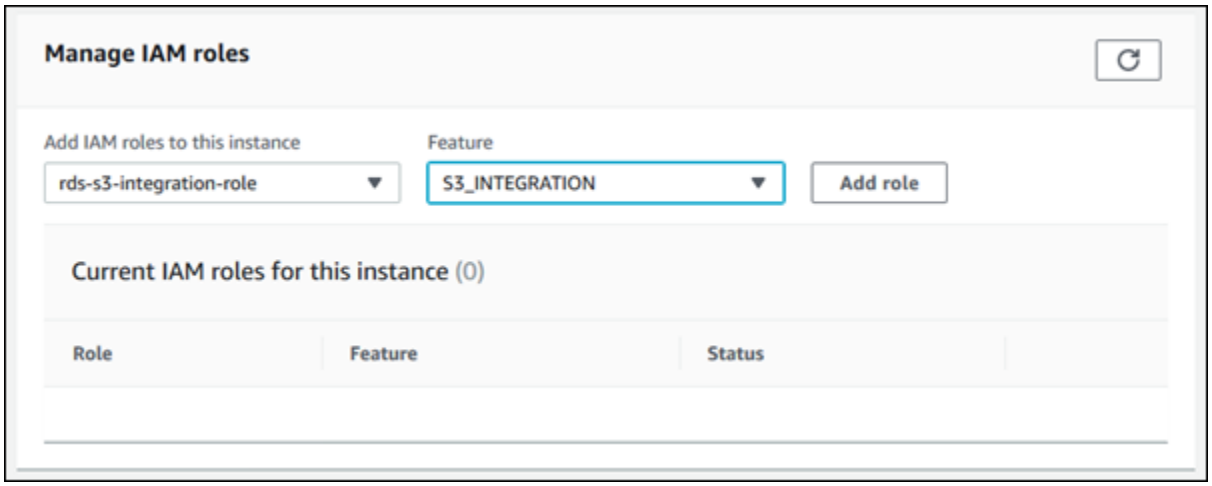
Nella sezione seguente viene descritto come abilitare l'integrazione di Amazon S3 con Amazon RDS for SQL Server. Per utilizzare l'integrazione di S3, l'istanza database deve essere associata al ruolo IAM precedentemente creato prima di utilizzare il parametro feature-name `S3_INTEGRATION`.

> **Note**
>
> Per aggiungere un ruolo IAM a un'istanza database, lo stato dell'istanza database deve essere available (disponibile).

Console

Per associare il ruolo IAM all'istanza database

1. Accedi alla AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Scegli il nome dell'istanza database RDS for SQL Server per visualizzarne i dettagli.
3. Nella sezione Manage IAM roles (Gestisci ruoli IAM) della scheda Connectivity & security (Connettività e sicurezza), selezionare il ruolo da aggiungere per Add IAM roles to this instance (Aggiungi ruoli IAM a questa istanza).
4. Per Feature (Caratteristica), selezionare S3_INTEGRATION.



5. Scegliere Add role (Aggiungi ruolo).

AWS CLI

Per aggiungere il ruolo IAM all'istanza database RDS for SQL Server

- Il seguente comando di AWS CLI aggiunge il ruolo IAM a un'istanza database RDS for SQL Server denominata *mydbinstance*.

  Example

  Per LinuxmacOS, oUnix:

```
aws rds add-role-to-db-instance \
    --db-instance-identifier mydbinstance \
    --feature-name S3_INTEGRATION \
```

```
    --role-arn your-role-arn
```

Per Windows:

```
aws rds add-role-to-db-instance ^
    --db-instance-identifier mydbinstance ^
    --feature-name S3_INTEGRATION ^
    --role-arn your-role-arn
```

Sostituire *your-role-arn* con il ruolo ARN annotato nel passaggio precedente. `S3_INTEGRATION` deve essere specificato per l'opzione `--feature-name`.

Trasferimento di file tra RDS for SQL Server e Amazon S3

Puoi utilizzare le stored procedure Amazon RDS per scaricare e caricare file tra Amazon S3 e l'istanza database RDS. Puoi anche usare le stored procedure Amazon RDS per elencare ed eliminare i file nell'istanza RDS.

I file che scarichi e carichi su S3 sono archiviati nella cartella `D:\S3`. Questa è l'unica cartella che puoi usare per accedere ai file. Puoi organizzare i file in sottocartelle che vengono create automaticamente quando specifichi la cartella di destinazione al momento del download.

Alcune delle stored procedure richiedono la specifica di un Amazon Resource Name (ARN) al tuo file e al bucket S3. Il formato per l'ARN è `arn:aws:s3:::bucket_name/file_name`. Amazon S3 non richiede un numero di account o una regione AWS negli ARN.

Le attività di integrazione di S3 vengono eseguite in sequenza e condividono la stessa coda delle attività di backup e ripristino native. In questa coda possono essere presenti al massimo due attività in esecuzione in qualsiasi momento. Possono essere necessari fino a cinque minuti prima che l'attività inizi l'elaborazione.

Download di file da un bucket Amazon S3 in un'istanza database SQL Server

Per scaricare file da un bucket S3 in un'istanza database RDS for SQL Server, usa la stored procedure Amazon RDS `msdb.dbo.rds_download_from_s3` con i seguenti parametri.

Nome del parametro	Tipo di dati	Default	Campo obbligatorio	Descrizione
`@s3_arn_of_file`	NVARCHAR	–	Campo obbligatorio	L'ARN S3 del file da scaricare, ad esempio: `arn:aws:s3:::bucket_name/mydata.csv`
`@rds_file_path`	NVARCHAR	–	Facoltativo	Il percorso del file per l'istanza RDS. Se non specificato, il percorso del file è `D:\S3\`*`<filename in s3>`*. RDS supporta percorsi assoluti e percorsi relativi. Per creare una sottocartella è necessario includerla nel percorso del file.
`@overwrite_file`	INT	0	Facoltativo	Sovrascrittura del file esistente: 0 = Non sovrascrivere 1 = Sovrascrivi

È possibile scaricare file senza estensione e file con le seguenti estensioni: .bcp, .csv, .dat, .fmt, .info, .lst, .tbl, .txt e .xml.

> **Note**
>
> I file con estensione .ispac sono supportati per il download quando SQL Server Integration Services è abilitato. Per ulteriori informazioni sull'abilitazione di SSIS, vedere SQL Server Integration Services (SSIS).
> I file con le seguenti estensioni di file sono supportati per il download quando SQL Server Analysis Services è

abilitato: .abf, .asdatabase, .configsettings, .deploymentoptions, .deploymenttargets e .xmla. Per ulteriori informazioni sull'abilitazione di SSAS, vedere SQL Server Analysis Services (SSAS).

L'esempio seguente mostra la stored procedure per scaricare i file da S3.

```
exec msdb.dbo.rds_download_from_s3
     @s3_arn_of_file='arn:aws:s3:::bucket_name/bulk_data.csv',
     @rds_file_path='D:\S3\seed_data\data.csv',
     @overwrite_file=1;
```

L'operazione `rds_download_from_s3` di esempio crea una cartella denominata `seed_data` in `D:\S3\`, se non esiste già. Quindi, viene scaricato il file di origine `bulk_data.csv` da S3 in un nuovo file denominato `data.csv` nell'istanza database. Se il file esisteva in precedenza, viene sovrascritto perché il parametro `@overwrite_file` è impostato su 1.

Aggiornamenti di file da un'istanza database SQL Server in un bucket Amazon S3

Per aggiornare i file da un'istanza database RDS for SQL Server in un bucket S3, usa la stored procedure Amazon RDS `msdb.dbo.rds_upload_to_s3` con i seguenti parametri.

Nome del parametro	Tipo di dati	Default	Campo obbligatorio	Descrizione
`@s3_arn_of_file`	NVARCHAR	–	Campo obbligatorio	L'ARN S3 del file da creare in S3, ad esempio: `arn:aws:s3:::bucket_name/mydata.csv`
`@rds_file_path`	NVARCHAR	–	Campo obbligatorio	Il percorso del file da caricare in S3. Sono supportati percorsi assoluti e relativi.
`@overwrite_file`	INT	–	Facoltativo	Sovrascrittura del file esistente:

Nome del parametro	Tipo di dati	Default	Campo obbligatorio	Descrizione
				0 = Non sovrascrivere 1 = Sovrascrivi

L'esempio seguente carica il file denominato `data.csv` dal percorso specificato in `D:\S3\seed_data\` nel file `new_data.csv` del bucket S3 specificato dall'ARN.

```
exec msdb.dbo.rds_upload_to_s3
  @rds_file_path='D:\S3\seed_data\data.csv',
  @s3_arn_of_file='arn:aws:s3:::bucket_name/new_data.csv',
  @overwrite_file=1;
```

Se il file esisteva in precedenza in S3, viene sovrascritto perché il parametro @overwrite_file è impostato su 1.

Visualizzazione di file nell'istanza database RDS

Per elencare i file disponibili nell'istanza database, si utilizza sia una stored procedure che una funzione. Innanzitutto, esegui la seguente stored procedure per raccogliere i dettagli dai file in `D:\S3\`.

```
exec msdb.dbo.rds_gather_file_details;
```

La stored procedure restituisce l'ID dell'attività. Come altre attività, questa stored procedure viene eseguita in modo asincrono. Non appena lo stato dell'attività diventa `SUCCESS`, puoi usare l'ID attività nella funzione `rds_fn_list_file_details` per elencare i file e le directory esistenti in D:\S3\, come indicato di seguito.

```
SELECT * FROM msdb.dbo.rds_fn_list_file_details(TASK_ID);
```

La funzione `rds_fn_list_file_details` restituisce una tabella con le seguenti colonne:

Parametro di output	Descrizione
`filepath`	Percorso assoluto del file (ad esempio, `D:\S3\mydata.csv`)
`size_in_bytes`	Dimensione del file (in byte)
`last_modified_utc`	Data e ora dell'ultima modifica in formato UTC
`is_directory`	Opzione che indica se l'elemento è una directory (`true`/`false`)

Eliminazione di file nell'istanza database RDS

Per eliminare i file disponibili nell'istanza DB, utilizza la stored procedure Amazon RDS `msdb.dbo.rds_delete_from_filesystem` con i seguenti parametri.

Nome del parametro	Tipo di dati	Default	Campo obbligatorio	Descrizione
`@rds_file_path`	NVARCHAR	–	Campo obbligatorio	Il percorso del file da eliminare. Sono supportati i percorsi assoluti e relativi.
`@force_delete`	INT	0	Facoltativo	Per eliminare una directory, questo flag deve essere incluso e impostato su 1. 1 = Elimina una directory Questo parametro viene ignorato se si elimina un file.

Per eliminare una directory, `@rds_file_path` deve terminare con una barra (\) e `@force_delete` deve essere impostato su `1`.

L'esempio seguente elimina il file `D:\S3\delete_me.txt`.

```
exec msdb.dbo.rds_delete_from_filesystem
    @rds_file_path='D:\S3\delete_me.txt';
```

L'esempio seguente elimina la directory `D:\S3\example_folder\`.

```
exec msdb.dbo.rds_delete_from_filesystem
    @rds_file_path='D:\S3\example_folder\',
    @force_delete=1;
```

Monitoraggio dello stato di un'attività di trasferimento di file

Per tenere traccia dello stato dell'attività di integrazione di S3, chiama la funzione `rds_fn_task_status` che accetta due parametri. Il primo parametro deve essere sempre `NULL` perché non si applica all'integrazione di S3. Il secondo parametro accetta un ID attività.

Per visualizzare l'elenco di tutte le attività, imposta il primo parametro su `NULL` e il secondo parametro su `0`, come indicato nell'esempio seguente.

```
SELECT * FROM msdb.dbo.rds_fn_task_status(NULL,0);
```

Per ottenere un'attività specifica, imposta il primo parametro su `NULL` e il secondo parametro sull'ID attività, come indicato nell'esempio seguente.

```
SELECT * FROM msdb.dbo.rds_fn_task_status(NULL,42);
```

La funzione `rds_fn_task_status` restituisce le seguenti informazioni.

Parametro di output	Descrizione
`task_id`	L'ID dell'attività.
`task_type`	Per l'integrazione di S3, le attività possono avere i seguenti tipi:

Parametro di output	Descrizione
	• DOWNLOAD_FROM_S3 • UPLOAD_TO_S3 • LIST_FILES_ON_DISK • DELETE_FILES_ON_DISK
database_name	Non applicabile alle attività di integrazione di S3.
% complete	L'avanzamento dell'attività espresso come percentuale.
duration(mins)	La quantità di tempo dedicato all'attività, in minuti.

Parametro di output	Descrizione
`lifecycle`	Lo stato dell'attività. I possibili stati sono i seguenti: • `CREATED` – Dopo aver chiamato una delle stored procedure di integrazione di S3, viene creata un'attività e lo stato viene impostato su `CREATED`. • `IN_PROGRESS` – Dopo l'avvio di un'attività, lo stato viene impostato su `IN_PROGRESS`. Possono essere necessari fino a 5 minuti perché lo stato cambi da `CREATED` in `IN_PROGRESS`. • `SUCCESS` – Al termine di un'attività, lo stato viene impostato su `SUCCESS`. • `ERROR` – Se un'attività non riesce, lo stato viene impostato su `ERROR`. Per ulteriori informazioni sull'errore, consulta la colonna `task_info`. • `CANCEL_REQUESTED` – Quando chiami `rds_cancel_task`, lo stato dell'attività viene impostato su `CANCEL_REQUESTED`. • `CANCELLED` – Dopo che un'attività è stata annullata, lo stato dell'attività viene impostato su `CANCELLED`.
`task_info`	Ulteriori informazioni sull'attività. Se si verifica un errore durante l'elaborazione, questa colonna contiene informazioni sull'errore.
`last_updated`	La data e l'ora dell'ultimo aggiornamento dello stato dell'attività.
`created_at`	La data e l'ora di creazione dell'attività.

Parametro di output	Descrizione
`S3_object_arn`	L'ARN dell'oggetto S3 scaricato o caricato.
`overwrite_S3_backup_file`	Non applicabile alle attività di integrazione di S3.
`KMS_master_key_arn`	Non applicabile alle attività di integrazione di S3.
`filepath`	Il percorso del file nell'istanza database RDS.
`overwrite_file`	Opzione che indica se un file esistente viene sovrascritto.
`task_metadata`	Non applicabile alle attività di integrazione di S3.

Annullamento di un'attività

Per annullare le attività di integrazione di S3, utilizzare la stored procedure `msdb.dbo.rds_cancel_task` con il parametro `task_id`. Le attività di eliminazione ed elenco in corso non possono essere annullate. L'esempio seguente mostra una richiesta per annullare un'attività.

```
exec msdb.dbo.rds_cancel_task @task_id = 1234;
```

Per ottenere una panoramica di tutte le attività e dei relativi ID attività, utilizza la funzione `rds_fn_task_status` come descritto in Monitoraggio dello stato di un'attività di trasferimento di file.

Limitazioni Multi-AZ per l'integrazione S3

Nelle istanze Multi-AZ, i file nella cartella `D:\S3` vengono eliminati nella replica di standby dopo un failover. Un failover può essere pianificato, ad esempio, durante le modifiche dell'istanza database, come la modifica della classe di istanza o l'aggiornamento della versione del motore. Oppure un failover può essere non pianificato, durante un'interruzione dell'istanza primaria.

> **Note**
>
> Non è consigliabile utilizzare la cartella `D:\S3` per lo storage di file. La best practice consiste nel caricare i file creati in Amazon S3 per renderli durevoli e scaricare i file quando è necessario importare i dati.

Per determinare l'ora dell'ultimo failover, puoi utilizzare la stored procedure `msdb.dbo.rds_failover_time`. Per ulteriori informazioni, consulta Determinazione dell'ora dell'ultimo failover.

Example di nessun failover recente

Questo esempio mostra l'output quando il log degli errori non contiene alcun failover recente. Nessun failover si è verificato dal 29-04-2020 alle 23:59:00.01.

Pertanto, tutti i file scaricati dopo tale ora che non sono stati eliminati utilizzando la stored procedure `rds_delete_from_filesystem` sono ancora accessibili sull'host corrente. Anche i file scaricati prima di tale ora potrebbero essere disponibili.

errorlog_available_from	recent_failover_time
2020-04-29 23:59:00.0100000	null

Example di failover recente

Questo esempio mostra l'output quando il log degli errori contiene un failover. Il failover più recente è stato il 05-05-2020 alle 18:57:51.89.

Tutti i file scaricati dopo quest'ora che non sono stati eliminati utilizzando la stored procedure `rds_delete_from_filesystem` sono ancora accessibili sull'host corrente.

errorlog_available_from	recent_failover_time
2020-04-29 23:59:00.0100000	2020-05-05 18:57:51.8900000

Disabilitazione dell'integrazione di RDS for SQL Server con S3

Nella sezione seguente sono riportate le indicazioni per disabilitare l'integrazione di Amazon S3 con Amazon RDS for SQL Server. I file in `D:\S3\` non vengono eliminati quando si disabilita l'integrazione S3.

> **Note**
>
> Per rimuovere un ruolo IAM da un'istanza database, lo stato dell'istanza database deve essere `available`.

Console

Per dissociare il ruolo IAM dall'istanza database

1. Accedi alla AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Scegli il nome dell'istanza database RDS for SQL Server per visualizzarne i dettagli.
3. Nella sezione Manage IAM roles (Gestisci ruoli IAM) della scheda Connectivity & security (Connettività e sicurezza), scegliere il ruolo IAM da rimuovere.
4. Scegliere Delete (Elimina).

AWS CLI

Per rimuovere il ruolo IAM dall'istanza database RDS for SQL Server

- Il seguente comando di AWS CLI rimuove il ruolo IAM da un'istanza database RDS for SQL Server denominata *mydbinstance*.

  Example

  Per LinuxmacOS, oUnix:

```
aws rds remove-role-from-db-instance \
    --db-instance-identifier mydbinstance \
    --feature-name S3_INTEGRATION \
    --role-arn your-role-arn
```

Per Windows:

```
aws rds remove-role-from-db-instance ^
    --db-instance-identifier mydbinstance ^
    --feature-name S3_INTEGRATION ^
    --role-arn your-role-arn
```

Sostituire *your-role-arn* con l'ARN del ruolo IAM appropriato per l'opzione `--feature-name`.

Utilizzo di Database Mail su Amazon RDS for SQL Server

Puoi utilizzare Database Mail per inviare messaggi di posta elettronica agli utenti da Amazon RDS dall'istanza database di SQL Server. I messaggi possono contenere file e risultati delle query. Database Mail include i seguenti componenti:

- Oggetti di configurazione e sicurezza – Questi oggetti creano profili e account e sono memorizzati nel database `msdb`.
- Oggetti di messaggistica – Questi oggetti includono la stored procedure sp_send_dbmail utilizzata per inviare messaggi e strutture di dati che contengono informazioni sui messaggi. Sono memorizzati nel database `msdb`.
- Oggetti di registrazione e controllo – Database Mail scrive le informazioni di registrazione nel database `msdb` e nel registro eventi applicazioni di Microsoft Windows.
- Eseguibile di Database Mail – `DatabaseMail.exe` legge da una coda nel database `msdb` e invia messaggi di posta elettronica.

RDS supporta Database Mail per tutte le versioni di SQL Server nelle edizioni Web, Standard ed Enterprise.

Limitazioni

Le seguenti limitazioni si applicano all'utilizzo di posta elettronica di database nell'istanza database di SQL Server:

- Database Mail non è supportato per SQL Server Express Edition.
- La modifica dei parametri di configurazione di Database Mail non è supportata. Per visualizzare i valori predefiniti, utilizza la stored procedure sysmail_help_configure_sp .
- Gli allegati dei file non sono completamente supportati. Per ulteriori informazioni, consulta Utilizzo di file allegati.
- La dimensione massima del file allegato è 1 MB.
- Database Mail richiede una configurazione aggiuntiva su istanze database multi-AZ. Per ulteriori informazioni, consulta Considerazioni per le implementazioni Multi-AZ.
- La configurazione di SQL Server Agent per l'invio di messaggi di posta elettronica agli operatori predefiniti non è supportata.

Abilitazione di Database Mail

Per abilitare Database Mail per l'istanza database, completa la seguente procedura:

1. Crea un nuovo set di parametri.
2. Modificare il gruppo di parametri per impostare il parametro `database mail xps` su 1 o 2.
3. Associa il nuovo gruppo di parametri all'istanza database.

Creazione del gruppo di parametri per Database Mail

Crea o modifica un gruppo di parametri per il parametro `database mail xps` corrispondente all'edizione di SQL Server e alla versione dell'istanza database.

> **Note**
>
> Puoi anche modificare un gruppo di parametri esistente. Segui la procedura riportata in Modifica del parametro che abilita Database Mail.

Console

Nell'esempio seguente viene creato un gruppo di parametri per SQL Server Standard Edition 2016.

Per creare il gruppo di parametri

1. Accedi alla AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel pannello di navigazione, scegli Parameter groups (Gruppi di parametri).
3. Scegliere Create parameter group (Crea gruppo di parametri).
4. Nel riquadro Create parameter group (Crea gruppi di parametri), procedi nel modo seguente:
   a. Per Famiglia del gruppo di parametri, scegliere sqlserver-se-13.0.
   b. Per Group name (Nome gruppo), immettere un identificatore per il gruppo di parametri, ad esempio **`dbmail-sqlserver-se-13`**.
   c. Per Description (Descrizione), immettere **`Database Mail XPs`**.
5. Scegliere Create (Crea).

CLI

Nell'esempio seguente viene creato un gruppo di parametri per SQL Server Standard Edition 2016.

Per creare il gruppo di parametri

- Utilizzare uno dei seguenti comandi.

  Example

  Per LinuxmacOS, oUnix:

```
aws rds create-db-parameter-group \
    --db-parameter-group-name dbmail-sqlserver-se-13 \
    --db-parameter-group-family "sqlserver-se-13.0" \
    --description "Database Mail XPs"
```

  Per Windows:

```
aws rds create-db-parameter-group ^
    --db-parameter-group-name dbmail-sqlserver-se-13 ^
    --db-parameter-group-family "sqlserver-se-13.0" ^
    --description "Database Mail XPs"
```

Modifica del parametro che abilita Database Mail

Modifica il parametro `database mail xps` nel gruppo di parametri che corrisponde all'edizione di SQL Server e alla versione dell'istanza database.

Per abilitare Database Mail, imposta il `database mail xps` parametro su 1.

Console

Nell'esempio seguente il gruppo di parametri creato per SQL Server Standard Edition 2016 viene modificato.

Per modificare il gruppo di parametri

1. Accedi alla AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione scegliere Parameter groups (Gruppi di parametri).

3. Scegliete il gruppo di parametri, ad esempio dbmail-sqlserver-se-13.
4. In Parameters (Parametri), filtrare l'elenco dei parametri per **`mail`**.
5. Seleziona database mail xps.
6. Scegliere Edit parameters (Modifica parametri).
7. Specificare (sì **1**.
8. Seleziona Save changes (Salva modifiche).

CLI

Nell'esempio seguente il gruppo di parametri creato per SQL Server Standard Edition 2016 viene modificato.

Per modificare il gruppo di parametri

- Utilizzare uno dei seguenti comandi.

  Example

  Per LinuxmacOS, oUnix:

```
aws rds modify-db-parameter-group \
    --db-parameter-group-name dbmail-sqlserver-se-13 \
    --parameters "ParameterName='database mail
 xps',ParameterValue=1,ApplyMethod=immediate"
```

  Per Windows:

```
aws rds modify-db-parameter-group ^
    --db-parameter-group-name dbmail-sqlserver-se-13 ^
    --parameters "ParameterName='database mail
 xps',ParameterValue=1,ApplyMethod=immediate"
```

Associazione del gruppo di parametri all'istanza database

Per associare il gruppo di parametri di Database Mail all'istanza database, puoi utilizzare AWS Management Console o AWS CLI.

Console

Puoi associare il gruppo di parametri di Database Mail a un'istanza database nuova o esistente.

- Per una nuova istanza database, associarli all'avvio dell'istanza. Per ulteriori informazioni, consulta Creazione di un'istanza database Amazon RDS.
- Per un'istanza database esistente, associarli modificando l'istanza. Per ulteriori informazioni, consulta Modifica di un'istanza database Amazon RDS.

CLI

Puoi associare il gruppo di parametri di Database Mail a un'istanza database nuova o esistente.

Per creare un'istanza database con il gruppo di parametri di Database Mail

- Specificare lo stesso tipo di motore di database e la versione principale utilizzati durante la creazione del gruppo di parametri.

  Example

  Per LinuxmacOS, oUnix:

```
aws rds create-db-instance \
    --db-instance-identifier mydbinstance \
    --db-instance-class db.m5.2xlarge \
    --engine sqlserver-se \
    --engine-version 13.00.5426.0.v1 \
    --allocated-storage 100 \
    --manage-master-user-password \
    --master-username admin \
    --storage-type gp2 \
    --license-model li
    --db-parameter-group-name dbmail-sqlserver-se-13
```

  Per Windows:

```
aws rds create-db-instance ^
    --db-instance-identifier mydbinstance ^
    --db-instance-class db.m5.2xlarge ^
    --engine sqlserver-se ^
    --engine-version 13.00.5426.0.v1 ^
```

```
    --allocated-storage 100 ^
    --manage-master-user-password ^
    --master-username admin ^
    --storage-type gp2 ^
    --license-model li ^
    --db-parameter-group-name dbmail-sqlserver-se-13
```

Per modificare un'istanza database e associare il gruppo di parametri di Database Mail

- Utilizzare uno dei seguenti comandi.

  Example

  Per LinuxmacOS, oUnix:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --db-parameter-group-name dbmail-sqlserver-se-13 \
    --apply-immediately
```

  Per Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --db-parameter-group-name dbmail-sqlserver-se-13 ^
    --apply-immediately
```

Configurazione di Database Mail

Per configurare Database Mail, puoi completare le attività riportate di seguito.

1. Crea il profilo Database Mail.
2. Crea l'account Database Mail.
3. Aggiungi l'account Database Mail al profilo Database Mail.
4. Aggiungi utenti al profilo Database Mail.

> **Note**
>
> Per configurare Database Mail, assicurati di disporre delle autorizzazioni `execute` per le stored procedure nel database `msdb`.

Creazione del profilo Database Mail

Per creare il profilo Database Mail, utilizza la stored procedure sysmail_add_profile_sp . Nell'esempio seguente viene creato un profilo denominato `Notifications`.

Per creare il profilo

- Utilizza la seguente istruzione SQL.

```
USE msdb
GO

EXECUTE msdb.dbo.sysmail_add_profile_sp
    @profile_name         = 'Notifications',
    @description          = 'Profile used for sending outgoing notifications using
 Amazon SES.';
GO
```

Creazione dell'account Database Mail

Per creare l'account Database Mail, utilizza la stored procedure sysmail_add_account_sp . Nell'esempio seguente viene creato un account denominato `SES` in un'istanza DB RDS per SQL Server in un VPC privato utilizzando Amazon Simple Email Service.

L'utilizzo di Amazon SES richiede i parametri seguenti:

- `@email_address`: un'identità verificata da Amazon SES. Per ulteriori informazioni su , consulta Verifica delle identità in Amazon SES.
- `@mailserver_name`: un endpoint SMTP Amazon SES. Per ulteriori informazioni, consulta Connessione a un endpoint SMTP Amazon SES.
- `@username`: un nome utente SMTP Amazon SES. Per ulteriori informazioni, consulta Come ottenere le credenziali SMTP in Amazon SES.

Non utilizzare un nome utente AWS Identity and Access Management.

- `@password`: password SMTP Amazon SES. Per ulteriori informazioni, consulta Come ottenere le credenziali SMTP in Amazon SES.

Per creare l'account

- Utilizza la seguente istruzione SQL.

```
USE msdb
GO

EXECUTE msdb.dbo.sysmail_add_account_sp
    @account_name        = 'SES',
    @description         = 'Mail account for sending outgoing notifications.',
    @email_address       = 'nobody@example.com',
    @display_name        = 'Automated Mailer',
    @mailserver_name     = 'vpce-0a1b2c3d4e5f-01234567.email-smtp.us-
west-2.vpce.amazonaws.com',
    @port                = 587,
    @enable_ssl          = 1,
    @username            = 'Smtp_Username',
    @password            = 'Smtp_Password';
GO
```

> **Note**
>
> Specifica credenziali diverse dai prompt mostrati qui come best practice per la sicurezza.

Aggiunta dell'account Database Mail al profilo Database Mail

Per aggiungere l'account Database Mail al profilo Database Mail, utilizza la stored procedure sysmail_add_profileaccount_sp . Nell'esempio seguente viene aggiunto l'account `SES` al profilo `Notifications`.

Per aggiungere l'account al profilo

- Utilizza la seguente istruzione SQL.

```
USE msdb
GO

EXECUTE msdb.dbo.sysmail_add_profileaccount_sp
    @profile_name        = 'Notifications',
    @account_name        = 'SES',
    @sequence_number     = 1;
GO
```

Aggiunta di utenti al profilo Database Mail

Per concedere l'autorizzazione a un principal di database `msdb` per l'utilizzo di un profilo Database Mail, utilizza la stored procedure sysmail_add_principalprofile_sp . Un principal è un'entità che può richiedere risorse di SQL Server. Il principal del database deve essere mappato a un utente di autenticazione di SQL Server, un utente di autenticazione di Windows o un gruppo di autenticazione di Windows.

Nell'esempio seguente viene concesso l'accesso pubblico al profilo `Notifications`.

Per aggiungere un utente al profilo

- Utilizza la seguente istruzione SQL.

```
USE msdb
GO

EXECUTE msdb.dbo.sysmail_add_principalprofile_sp
    @profile_name       = 'Notifications',
    @principal_name     = 'public',
    @is_default         = 1;
GO
```

Amazon RDS stored procedure e funzioni per Database Mail

Microsoft fornisce Procedure archiviate per l'utilizzo di Mail database, ad esempio la creazione, l'elenco, l'aggiornamento ed eliminazione di account e profili. Inoltre, RDS fornisce le stored procedure e le funzioni per Database Mail riportate nella tabella seguente.

Stored procedure/Funzione	Descrizione
rds_fn_sysmail_allitems	Mostra i messaggi inviati, inclusi quelli inviati da altri utenti.
rds_fn_sysmail_event_log	Mostra gli eventi, inclusi quelli relativi ai messaggi inviati da altri utenti.
rds_fn_sysmail_mailattachments	Mostra gli allegati, inclusi quelli inviati da altri utenti.
rds_sysmail_control	Avvia e arresta la coda di posta (DatabaseMailprocesso.exe).
rds_sysmail_delete_mailitems_sp	Elimina i messaggi di posta elettronica inviati da tutti gli utenti dalle tabelle interne a Database Mail.

Invio di messaggi di posta elettronica tramite Database Mail

Per inviare messaggi di posta elettronica utilizzando Database Mail, puoi utilizzare la stored procedure sp_send_dbmail.

Utilizzo

```
EXEC msdb.dbo.sp_send_dbmail
@profile_name = 'profile_name',
@recipients = 'recipient1@example.com[; recipient2; ... recipientn]',
@subject = 'subject',
@body = 'message_body',
[@body_format = 'HTML'],
[@file_attachments = 'file_path1; file_path2; ... file_pathn'],
[@query = 'SQL_query'],
[@attach_query_result_as_file = 0|1]';
```

I parametri seguenti sono obbligatori:

- `@profile_name` – Il nome del profilo Database Mail da cui inviare il messaggio.
- `@recipients` – L'elenco delimitato da punto e virgola di indirizzi di posta elettronica a cui inviare il messaggio.
- `@subject` – L'oggetto del messaggio.

- `@body` – Il corpo del messaggio. Puoi inoltre utilizzare una variabile dichiarata come corpo.

I parametri seguenti sono facoltativi:

- `@body_format` – Questo parametro viene utilizzato con una variabile dichiarata per inviare e-mail in formato HTML.
- `@file_attachments` – L'elenco delimitato da punto e virgola degli allegati dei messaggi. I percorsi dei file devono essere percorsi assoluti.
- `@query` – Una query SQL da eseguire. I risultati della query possono essere allegati come file o inclusi nel corpo del messaggio.
- `@attach_query_result_as_file` – Indica se allegare il risultato della query come file. Imposta su 0 per no, 1 per sì. Il valore predefinito è 0.

Esempi

Negli esempi seguenti viene illustrato come inviare messaggi di posta elettronica.

Example di invio di un messaggio a un singolo destinatario

```
USE msdb
GO

EXEC msdb.dbo.sp_send_dbmail
     @profile_name       = 'Notifications',
     @recipients         = 'nobody@example.com',
     @subject            = 'Automated DBMail message - 1',
     @body               = 'Database Mail configuration was successful.';
GO
```

Example di invio di un messaggio a più destinatari

```
USE msdb
GO

EXEC msdb.dbo.sp_send_dbmail
     @profile_name       = 'Notifications',
     @recipients         = 'recipient1@example.com;recipient2@example.com',
     @subject            = 'Automated DBMail message - 2',
     @body               = 'This is a message.';
```

```
GO
```

Example di invio di un risultato di una query SQL come file allegato

```
USE msdb
GO

EXEC msdb.dbo.sp_send_dbmail
     @profile_name       = 'Notifications',
     @recipients         = 'nobody@example.com',
     @subject            = 'Test SQL query',
     @body               = 'This is a SQL query test.',
     @query              = 'SELECT * FROM abc.dbo.test',
     @attach_query_result_as_file = 1;
GO
```

Example di invio di un messaggio in formato HTML

```
USE msdb
GO

DECLARE @HTML_Body as NVARCHAR(500) = 'Hi, <h4> Heading </h4> </br> See the report. <b>
 Regards </b>';

EXEC msdb.dbo.sp_send_dbmail
     @profile_name       = 'Notifications',
     @recipients         = 'nobody@example.com',
     @subject            = 'Test HTML message',
     @body               = @HTML_Body,
     @body_format        = 'HTML';
GO
```

Example di invio di un messaggio utilizzando un trigger quando si verifica un evento specifico nel database

```
USE AdventureWorks2017
GO
IF OBJECT_ID ('Production.iProductNotification', 'TR') IS NOT NULL
DROP TRIGGER Purchasing.iProductNotification
GO
```

```
CREATE TRIGGER iProductNotification ON Production.Product
   FOR INSERT
   AS
   DECLARE @ProductInformation nvarchar(255);
   SELECT
   @ProductInformation = 'A new product, ' + Name + ', is now available for $' +
 CAST(StandardCost AS nvarchar(20)) + '!'
   FROM INSERTED i;

EXEC msdb.dbo.sp_send_dbmail
     @profile_name       = 'Notifications',
     @recipients         = 'nobody@example.com',
     @subject            = 'New product information',
     @body               = @ProductInformation;
GO
```

Visualizzazione di messaggi, log e allegati

È possibile utilizzare le stored procedure RDS per visualizzare messaggi, log di eventi e allegati.

Per visualizzare tutti i messaggi di posta elettronica

- Digita la seguente query SQL:

```
SELECT * FROM msdb.dbo.rds_fn_sysmail_allitems(); --WHERE sent_status='sent' or
 'failed' or 'unsent'
```

Per visualizzare tutti i log di eventi di posta elettronica

- Digita la seguente query SQL:

```
SELECT * FROM msdb.dbo.rds_fn_sysmail_event_log();
```

Per visualizzare tutti gli allegati di posta elettronica

- Digita la seguente query SQL:

```
SELECT * FROM msdb.dbo.rds_fn_sysmail_mailattachments();
```

Eliminazione dei messaggi

Per eliminare i messaggi, puoi utilizzare la stored procedure `rds_sysmail_delete_mailitems_sp`.

> **Note**
>
> RDS elimina automaticamente gli elementi della tabella di posta quando i dati della cronologia di DBMail raggiungono le dimensioni di 1 GB, con un periodo di conservazione di almeno 24 ore.
> Se desideri conservare gli elementi di posta per un periodo più lungo, puoi archiviarli. Per maggiori informazioni, consulta Creazione di un processo di SQL Server Agent per archiviare i messaggi di Database Mail e i log di eventi nella documentazione di Microsoft.

Per eliminare tutti i messaggi di posta elettronica

- Utilizza la seguente istruzione SQL.

```
DECLARE @GETDATE datetime
SET @GETDATE = GETDATE();
EXECUTE msdb.dbo.rds_sysmail_delete_mailitems_sp @sent_before = @GETDATE;
GO
```

Per eliminare tutti i messaggi di posta elettronica con uno stato particolare

- Utilizza la seguente istruzione SQL per eliminare tutti i messaggi non riusciti.

```
DECLARE @GETDATE datetime
SET @GETDATE = GETDATE();
EXECUTE msdb.dbo.rds_sysmail_delete_mailitems_sp @sent_status = 'failed';
GO
```

Avvio della coda di posta

Per avviare il processo Database Mail, utilizza la stored procedure `rds_sysmail_control` .

> **Note**
>
> L'abilitazione di Database Mail avvia automaticamente la coda di posta.

Per avviare la coda di posta

- Utilizza la seguente istruzione SQL.

```
EXECUTE msdb.dbo.rds_sysmail_control start;
GO
```

Arresto della coda di posta

Per arrestare il processo di Database Mail, utilizza la stored procedure `rds_sysmail_control`.

Per arrestare la coda di posta

- Utilizza la seguente istruzione SQL.

```
EXECUTE msdb.dbo.rds_sysmail_control stop;
GO
```

Utilizzo di file allegati

Le seguenti estensioni di file allegati non sono supportate nei messaggi di Database Mail provenienti da RDS su SQL
Server: .ade, .adp, .apk, .appx, .appxbundle, .bat, .bak, .cab, .chm, .cmd, .com, .cpl, .dll, .dmg, .exe, .hta, .inf e .wsh.

Database Mail utilizza il contesto di protezione di Microsoft Windows dell'utente corrente per controllare l'accesso ai file. Gli utenti che accedono con l'autenticazione di SQL Server non possono allegare file utilizzando il parametro `@file_attachments` con la stored procedure `sp_send_dbmail`. Windows non consente a SQL Server di fornire le credenziali da un computer remoto a un altro computer remoto. Di conseguenza, Database Mail non può allegare file da una condivisione di rete quando il comando viene eseguito da un computer diverso dal computer che esegue SQL Server.

Tuttavia, per allegare i file puoi utilizzare i processi di SQL Server Agent. Per ulteriori informazioni su SQL Server Agent, consulta Uso di SQL Server Agent e SQL Server Agent nella documentazione Microsoft.

Considerazioni per le implementazioni Multi-AZ

Quando configuri Database Mail in un'istanza database Multi-AZ, la configurazione non viene propagata automaticamente al nodo secondario. Si consiglia di convertire l'istanza Multi-AZ in un'istanza Single-AZ, configurare Database Mail e quindi riconvertire l'istanza database in Multi-AZ. In questo modo, entrambi i nodi primario e secondario avranno la configurazione di Database Mail.

Se crei una replica di lettura dall'istanza Multi-AZ in cui è configurato Database Mail, la replica eredita la configurazione, ma senza la password sul server SMTP. Aggiorna l'account Database Mail con la password.

Supporto dell'archivio istanze per il database tempdb su Amazon RDS for SQL Server

Un archivio istanze fornisce uno storage temporaneo di livello per l'istanza database. Lo storage è collocato all'interno dei dischi fisicamente collegati al computer host. Questi dischi dispongono di storage di istanza NVMe (Non-Volatile Memory Express) basata su unità a stato solido (SSD). Questo storage è ottimizzato per bassa latenza, prestazioni I/O casuali molto elevate ed elevata velocità di lettura sequenziale.

Inserendo i file di dati `tempdb` e i file di log `tempdb` nell'archivio istanze, sarà possibile ottenere latenze di lettura e scrittura inferiori rispetto allo storage standard basato su Amazon EBS.

> **Note**
>
> I file di database di SQL Server e i file di log del database non vengono inseriti nell'archivio istanze.

Abilitazione dell'archivio istanze

Quando RDS esegue il provisioning di istanze database con una delle seguenti classi di istanza, il database `tempdb` viene automaticamente inserito nell'archivio istanze:

- db.m5d
- db.r5d
- db.x2iedn

Per abilitare l'archivio istanze, effettua una delle seguenti operazioni:

- Crea un'istanza database di SQL Server utilizzando uno di questi tipi di istanza. Per ulteriori informazioni, consulta Creazione di un'istanza database Amazon RDS.
- Modifica un'istanza database di SQL Server esistente per utilizzarne una. Per ulteriori informazioni, consulta Modifica di un'istanza database Amazon RDS.

L'archivio istanze è disponibile in tutte le regioni AWS in cui sono supportati uno o più di questi tipi di istanza. Per ulteriori informazioni sulle classi di istanza `db.m5d` e `db.r5d`, consulta Classi di istanze

database. Per ulteriori informazioni sulle classi di istanza supportate da Amazon RDS for SQL Server, consulta Supporto classe istanza database per Microsoft SQL Server.

Considerazioni sulla posizione e sulle dimensioni dei file

Nelle istanze senza un archivio di istanze, RDS memorizza i file di dati e di log `tempdb` nella directory `D:\rdsdbdata\DATA`. Entrambi i file iniziano da 8 MB per impostazione predefinita.

Nelle istanze con un archivio istanze, RDS memorizza i file di dati e di log `tempdb` nella directory `T:\rdsdbdata\DATA`.

Quando `tempdb` ha un solo file di dati (`tempdb.mdf`) e un file di log (`templog.ldf`), `templog.ldf` inizia da 8 MB per impostazione predefinita e `tempdb.mdf` inizia all'80% o più della capacità di archiviazione dell'istanza. Il 20% della capacità di storage o 200 GB, a seconda di quale sia inferiore, viene mantenuto libero per iniziare. Più file di dati `tempdb` dividono uniformemente l'80% dello spazio su disco, mentre i file di log hanno sempre una dimensione iniziale di 8 MB.

Ad esempio, se si modifica la classe di istanza database da `db.m5.2xlarge` a `db.m5d.2xlarge`, la dimensione dei file di dati `tempdb` aumenta da 8 MB ciascuno a 234 GB in totale.

> **Note**
>
> Oltre ai file di dati e di log di `tempdb` nell'archivio istanze (`T:\rdsdbdata\DATA`), puoi ancora creare file di dati e file di log `tempdb` nel volume di dati (`D:\rdsdbdata\DATA`). Questi file hanno sempre una dimensione iniziale di 8 MB.

Considerazioni sul backup

Potrebbe essere necessario conservare i backup per lunghi periodi, con costi nel tempo. I blocchi di dati e log di `tempdb` possono cambiare molto spesso a seconda del carico di lavoro. Ciò può aumentare notevolmente la dimensione degli snapshot del database.

Quando si `tempdb` trova nell'archivio istanze, le istantanee non includono file temporanei. Ciò significa che le dimensioni degli snapshot sono più piccole e consumano meno l'allocazione di backup gratuita rispetto allo storage solo EBS.

Errori di disco pieno

Se si utilizza tutto lo spazio disponibile nell'archivio istanze, è possibile che vengano visualizzati errori come i seguenti:

- Il log delle transazioni per il database 'tempdb' è pieno a causa di 'ACTIVE_TRANSACTION '.
- Impossibile allocare spazio per l'oggetto 'dbo.sort temporanea di archiviazione esecuzione: 140738941419520' nel database 'tempdb' perché il filegroup 'PRIMARY' è pieno. Crea spazio su disco eliminando i file non necessari, rilasciando oggetti nel gruppo di file, aggiungendo altri file al gruppo di file o impostando il parametro autogrowth per i file esistenti nel gruppo di file.

È possibile eseguire una o più delle seguenti operazioni quando l'archivio istanze è pieno:

- Regola il carico di lavoro o il modo in cui utilizzi `tempdb`.
- Scala fino all'utilizzo di una classe di istanza database con più storage NVMe.
- Interrompi l'utilizzo dell'archivio istanze e utilizza una classe di istanza con solo storage EBS.
- Utilizza una modalità mista aggiungendo dati secondari o file di log per `tempdb` sul volume EBS.

Rimozione dell'archivio istanze

Per rimuovere l'archivio dell'istanza, modifica l'istanza database di SQL Server per utilizzare un tipo di istanza che non supporta l'archivio dell'istanza, ad esempio db.m5, db.r5 o db.x1e.

> **Note**
>
> Quando rimuovi l'archivio istanze, i file temporanei vengono spostati nella directory `D:\rdsdbdata\DATA` e le dimensioni sono ridotte a 8 MB.

Utilizzo di eventi estesi con Amazon RDS for Microsoft SQL Server.

È possibile utilizzare eventi estesi in Microsoft SQL Server per acquisire informazioni di debug e risoluzione dei problemi per Amazon RDS for SQL Server. Gli eventi estesi sostituiscono SQL Trace e Server Profiler, che sono stati deprecati da Microsoft. Gli eventi estesi sono simili alle tracce del profiler, ma hanno un controllo più granulare sugli eventi tracciati. Gli eventi estesi sono supportati per SQL Server versioni 2014 e successive su Amazon RDS. Per ulteriori informazioni, consulta Panoramica degli eventi estesi nella documentazione di Microsoft.

Gli eventi estesi vengono attivati automaticamente per gli utenti con privilegi utente master in Amazon RDS for SQL Server.

Argomenti

Limitazioni e consigli

Quando si utilizzano eventi estesi su RDS per SQL Server, si applicano le seguenti limitazioni:

- Gli eventi estesi sono supportati solo per le edizioni Enterprise e Standard.
- Non è possibile modificare le sessioni degli eventi estesi predefinite.
- Assicurati di impostare la modalità di partizione della memoria della sessione su `NONE`.
- La modalità di conservazione degli eventi della sessione può essere `ALLOW_SINGLE_EVENT_LOSS` o `ALLOW_MULTIPLE_EVENT_LOSS`.
- Le destinazioni di traccia eventi per Windows (ETW) non sono supportate.
- Assicurati che le destinazioni dei file siano nella directory `D:\rdsdbdata\log`.
- Per le destinazioni corrispondenti alle coppie, imposta la proprietà `respond_to_memory_pressure` su `1`.
- La memoria di destinazione del buffer ring non può essere maggiore di 4 MB.
- Le seguenti azioni non sono supportate:
  - `debug_break`
  - `create_dump_all_threads`

  - `create_dump_single_threads`
- L'evento `rpc_completed` è supportato nelle seguenti versioni: 15.0.4083.2, 14.0.3370.1, 13.0.5865.1, 12.0.6433.1, 11.0.7507.2 e nelle versioni successive.

Configurazione di eventi estesi su RDS per SQL Server

In RDS per SQL Server, è possibile configurare i valori di determinati parametri delle sessioni degli eventi estesi. Nella tabella seguente vengono descritti i parametri configurabili.

Nome del parametro	Descrizione
`xe_session_max_memory`	Specifica la quantità massima di memoria da allocare alla corrisponde all'impostazione `max_memory` della sessior
`xe_session_max_event_size`	Specifica la dimensione massima di memoria consentita p corrisponde all'impostazione `max_event_size` della se
`xe_session_max_dispatch_latency`	Specifica il tempo in cui gli eventi vengono memorizzati n sessioni degli eventi estesi. Questo valore corrisponde all sessione evento.
`xe_file_target_size`	Specifica la dimensione massima della destinazione file. `max_file_size` della destinazione del file.
`xe_file_retention`	Specifica il tempo di conservazione (in giorni) per i file ge

> **Note**
>
> Impostando `xe_file_retention` su zero, i file .xel vengono rimossi automaticamente dopo il blocco di questi file viene rilasciato da SQL Server. Il blocco viene rilasciato ogni volta che un file con estensione .xel raggiunge il limite di dimensione impostato in `xe_file_target_size`.

È possibile utilizzare la stored procedure `rdsadmin.dbo.rds_show_configuration` per visualizzare i valori correnti di questi parametri. Ad esempio, utilizzare l'istruzione SQL seguente per visualizzare l'impostazione corrente di `xe_session_max_memory`.

```
exec rdsadmin.dbo.rds_show_configuration 'xe_session_max_memory'
```

È possibile utilizzare la stored procedure `rdsadmin.dbo.rds_set_configuration` per modificarle. Ad esempio, utilizzare l'istruzione SQL seguente per impostare `xe_session_max_memory` su 4 MB.

```
exec rdsadmin.dbo.rds_set_configuration 'xe_session_max_memory', 4
```

Considerazioni per le implementazioni Multi-AZ

Quando si crea una sessione per un evento esteso in un'istanza DB primaria, la sessione non viene propagata alla replica in standby. È possibile eseguire il failover e creare la sessione dell'evento esteso nella nuova istanza DB primaria. In alternativa, è possibile rimuovere e aggiungere nuovamente la configurazione Multi-AZ per propagare la sessione di eventi estesi alla replica in attesa. RDS interrompe tutte le sessioni degli eventi estesi non predefinite sulla replica in standby, in modo che queste sessioni non utilizzino risorse in standby. Per questo motivo, dopo che una replica in standby diventa l'istanza DB primaria, assicurarsi di avviare manualmente le sessioni dell'evento esteso sul nuovo primario.

> **Note**
>
> Questo approccio si applica sia ai gruppi di disponibilità Always On che al mirroring del database.

È inoltre possibile utilizzare un processo di SQL Server Agent per tenere traccia della replica in standby e avviare le sessioni se lo standby diventa primario. Ad esempio, utilizzare la seguente query nel passaggio del processo Agente SQL Server per riavviare le sessioni di eventi in un'istanza DB primaria.

```
BEGIN
    IF (DATABASEPROPERTYEX('rdsadmin','Updateability')='READ_WRITE'
    AND DATABASEPROPERTYEX('rdsadmin','status')='ONLINE'
    AND (DATABASEPROPERTYEX('rdsadmin','Collation') IS NOT NULL OR
 DATABASEPROPERTYEX('rdsadmin','IsAutoClose')=1)
    )
    BEGIN
        IF NOT EXISTS (SELECT 1 FROM sys.dm_xe_sessions WHERE name='xe1')
```

```
            ALTER EVENT SESSION xe1 ON SERVER STATE=START
        IF NOT EXISTS (SELECT 1 FROM sys.dm_xe_sessions WHERE name='xe2')
            ALTER EVENT SESSION xe2 ON SERVER STATE=START
    END
END
```

Questa query riavvia le sessioni di evento `xe1` e `xe2` su un'istanza DB primaria se queste sessioni sono in uno stato interrotto. È inoltre possibile aggiungere una pianificazione con un intervallo conveniente a questa query.

Esecuzione di query sui file di eventi estesi

È possibile utilizzare SQL Server Management Studio o la `sys.fn_xe_file_target_read_file` funzione per visualizzare i dati da eventi estesi che utilizzano destinazioni file. Per ulteriori informazioni su questa funzione, consulta sys.fn_xe_file_target_read_file (Transact-SQL) nella documentazione di Microsoft.

Le destinazioni dei file degli eventi estesi possono scrivere solo file nella directory `D:\rdsdbdata\log` su RDS for SQL Server.

Ad esempio, utilizzare la seguente query SQL per elencare il contenuto di tutti i file delle sessioni di eventi estesi i cui nomi iniziano con `xe`.

```
SELECT * FROM sys.fn_xe_file_target_read_file('d:\rdsdbdata\log\xe*', null,null,null);
```

Accesso ai backup dei log delle transazioni con RDS per SQL Server

Con l'accesso ai backup dei log delle transazioni di RDS per SQL Server, puoi elencare i file di backup dei log delle transazioni per un database e copiarli in un bucket Amazon S3 di destinazione. Copiando i backup dei log delle transazioni in un bucket Amazon S3, puoi utilizzarli in combinazione con backup completi e differenziali del database per eseguire ripristini point-in-time del database. Utilizzi le stored procedure RDS per configurare l'accesso ai backup dei log delle transazioni, elencare i backup dei log delle transazioni disponibili e copiarli nel bucket Amazon S3.

L'accesso ai backup dei log delle transazioni offre le funzionalità e i vantaggi seguenti:

- Elenca e visualizza i metadati dei backup dei log delle transazioni disponibili per un database su un'istanza database di RDS per SQL Server.
- Copia i backup dei log delle transazioni disponibili da RDS per SQL Server in un bucket Amazon S3 di destinazione.
- Esegui ripristini point-in-time dei database senza la necessità di ripristinare un'intera istanza database. Per ulteriori informazioni sul ripristino point-in-time di un'istanza database, consulta Ripristino a un'ora specifica per un'istanza database.

Disponibilità e supporto

L'accesso ai backup dei log delle transazioni è supportato in tutte le Regioni AWS. L'accesso ai backup dei log delle transazioni è disponibile per tutte le edizioni e le versioni di Microsoft SQL Server supportate su Amazon RDS.

Requisiti

I seguenti requisiti devono essere soddisfatti prima di consentire l'accesso ai backup dei log delle transazioni:

- I backup automatici devono essere abilitati sull'istanza database e la conservazione del backup deve essere impostata su un valore di uno o più giorni. Per ulteriori informazioni sull'attivazione dei backup automatici e sulla configurazione di una policy di conservazione, consulta Abilitazione dei backup automatici.
- Un bucket Amazon S3 deve trovarsi nello stesso account e nella stessa regione dell'istanza database di origine. Prima di abilitare l'accesso ai backup dei log delle transazioni, scegli un bucket Amazon S3 esistente o crea un nuovo bucket da utilizzare per i file di backup dei log delle transazioni.

- Una policy di autorizzazione per i bucket Amazon S3 deve essere configurata come segue per consentire ad Amazon RDS di copiarvi i file di log delle transazioni:
  1. Imposta la proprietà del proprietario dell'account dell'oggetto nel bucket su Bucket Owner Preferred (Proprietario preferito del bucket).
  2. Aggiungi la policy seguente. Non ci sarà alcuna policy per impostazione predefinita, quindi utilizza le liste di controllo degli accessi (ACL) del bucket per modificare la policy del bucket e aggiungerla.

Nell'esempio che segue viene utilizzato un ARN per specificare la risorsa. Si consiglia di utilizzare le chiavi di contesto delle condizioni globali `SourceArn` e `SourceAccount` nelle relazioni di trust basate sulle risorse per limitare le autorizzazioni del servizio relative a una risorsa specifica. Per ulteriori informazioni sull'utilizzo degli ARN, consulta Amazon resource names (ARN) e Utilizzo di Amazon Resource Name (ARN) in Amazon RDS.

Example di policy di autorizzazione di Amazon S3 per l'accesso ai backup dei log delle transazioni

```
    {
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Only allow writes to my bucket with bucket owner full control",
            "Effect": "Allow",
            "Principal": {
                "Service": "backups.rds.amazonaws.com"
            },
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::{customer_bucket}/{customer_path}/*",
            "Condition": {
                "StringEquals": {
                    "s3:x-amz-acl": "bucket-owner-full-control",
                    "aws:sourceAccount": "{customer_account}",
                    "aws:sourceArn": "{db_instance_arn}"
                }
            }
        }
    ]
}
```

- Un ruolo AWS Identity and Access Management (IAM) per accedere al bucket Amazon S3. Se disponi già di un ruolo IAM, puoi utilizzarlo. In alternativa, puoi scegliere che venga creato automaticamente un nuovo ruolo IAM quando aggiungi l'opzione `SQLSERVER_BACKUP_RESTORE` usando la AWS Management Console. In alternativa, è possibile crearne uno nuovo manualmente. Per ulteriori informazioni sulla creazione e la configurazione di un ruolo IAM con `SQLSERVER_BACKUP_RESTORE`, consulta Creazione manuale di un ruolo IAM per backup e ripristino nativi.
- L'opzione `SQLSERVER_BACKUP_RESTORE` deve essere aggiunta a un gruppo di opzioni nella tua istanza database. Per ulteriori informazioni sull'aggiunta dell'opzione `SQLSERVER_BACKUP_RESTORE`, consulta Supporto per backup nativo e ripristino in SQL Server.

  > **Note**
  >
  > Se l'istanza database ha la crittografia dell'archiviazione abilitata, le azioni e la chiave AWS KMS (KMS) devono essere fornite nel ruolo IAM specificato nel gruppo di opzioni di backup e ripristino nativo.

  Facoltativamente, se intendi utilizzare la stored procedure `rds_restore_log` per eseguire ripristini point-in-time del database, ti consigliamo di utilizzare lo stesso percorso Amazon S3 per il gruppo di opzioni di backup e ripristino nativo e l'accesso ai backup dei log delle transazioni. Questo metodo garantisce che, quando Amazon RDS assume il ruolo del gruppo di opzioni per eseguire le funzioni del log di ripristino, abbia accesso ai backup dei log delle transazioni dallo stesso percorso di Amazon S3.
- Se l'istanza DB è crittografata, indipendentemente dal tipo di crittografia (chiave gestita da AWS o chiave gestita dal cliente), è necessario fornire una chiave KMS gestita dal cliente nel ruolo IAM e nella stored procedure `rds_tlog_backup_copy_to_S3`.

Limitazioni e consigli

L'accesso ai backup dei log delle transazioni include le seguenti limitazioni e raccomandazioni:

- È possibile elencare e copiare fino agli ultimi sette giorni di backup dei log delle transazioni per qualsiasi istanza database la cui conservazione del backup è configurata tra uno e 35 giorni.

- Il bucket Amazon S3 utilizzato per l'accesso ai backup dei log delle transazioni deve essere presente nello stesso account e nella stessa regione dell'istanza database di origine. La copia tra account e tra regioni non è supportata.
- È possibile configurare un solo bucket Amazon S3 come destinazione in cui copiare i backup dei log delle transazioni. Puoi scegliere un nuovo bucket Amazon S3 di destinazione con la stored procedure `rds_tlog_copy_setup`. Per ulteriori informazioni sulla scelta di un nuovo bucket Amazon S3 di destinazione, consulta Configurazione dell'accesso ai backup dei log delle transazioni.
- Non è possibile specificare la chiave KMS quando si utilizza la stored procedure `rds_tlog_backup_copy_to_S3` se l'istanza RDS non è abilitata per la crittografia dell'archiviazione.
- La copia tra account non è supportata. Il ruolo IAM utilizzato per la copia consente l'accesso in scrittura ai bucket Amazon S3 solo all'interno dell'account proprietario dell'istanza database.
- Solo due attività simultanee di qualsiasi tipo possono essere eseguite su un'istanza database RDS per SQL Server.
- È possibile eseguire una sola operazione di copia alla volta per ogni singolo database. Se desideri copiare i backup dei log delle transazioni per più database sull'istanza database, utilizza un'attività di copia separata per ogni database.
- Se copi un backup dei log delle transazioni già esistente con lo stesso nome nel bucket Amazon S3, il backup dei log delle transazioni esistente verrà sovrascritto.
- È possibile eseguire solo le stored procedure che consentono l'accesso ai backup dei log delle transazioni sull'istanza database primaria. Non è possibile eseguire queste stored procedure su una replica di lettura di RDS per SQL Server o su un'istanza secondaria di un cluster database multi-AZ.
- Se l'istanza database di RDS per SQL Server viene riavviata mentre la stored procedure `rds_tlog_backup_copy_to_S3` è in esecuzione, l'attività verrà riavviata automaticamente dall'inizio quando l'istanza database tornerà online. Qualsiasi backup dei log delle transazioni copiato nel bucket Amazon S3 mentre l'attività è in esecuzione prima del riavvio verrà sovrascritto.
- I database di sistema Microsoft SQL Server e il database `RDSAdmin` non possono essere configurati per l'accesso ai backup dei log delle transazioni.
- La copia su bucket con crittografia SSE-KMS non è supportata.

Configurazione dell'accesso ai backup dei log delle transazioni

Per configurare l'accesso ai backup dei log delle transazioni, completa l'elenco dei requisiti nella sezione Requisiti, quindi esegui la stored procedure `rds_tlog_copy_setup`. La procedura consente l'accesso alla funzionalità di backup dei log delle transazioni a livello di istanza database. Non è necessario eseguirla per ogni singolo database nell'istanza database.

> **Important**
>
> All'utente del database deve essere concesso il ruolo `db_owner` in SQL Server su ogni database per configurare e utilizzare la funzionalità di accesso ai backup dei log delle transazioni.

Example di utilizzo:

```
exec msdb.dbo.rds_tlog_copy_setup
@target_s3_arn='arn:aws:s3:::mybucket/myfolder';
```

Il parametro seguente è obbligatorio:

- `@target_s3_arn`: l'ARN del bucket Amazon S3 di destinazione in cui copiare i file di backup dei log delle transazioni.

Example di impostazione di un bucket Amazon S3:

```
exec msdb.dbo.rds_tlog_copy_setup @target_s3_arn='arn:aws:s3:::accesstlogs-
testbucket/mytestdb1';
```

Per convalidare la configurazione, chiama la stored procedure `rds_show_configuration`.

Example di convalida della configurazione:

```
exec rdsadmin.dbo.rds_show_configuration @name='target_s3_arn_for_tlog_copy';
```

Per modificare l'accesso ai backup dei log delle transazioni in modo che punti a un bucket Amazon S3 diverso, puoi visualizzare il valore corrente del bucket Amazon S3 ed eseguire nuovamente la store procedure `rds_tlog_copy_setup` utilizzando un nuovo valore per `@target_s3_arn`.

Example di visualizzazione del bucket Amazon S3 esistente configurato per l'accesso ai backup dei log delle transazioni

```
exec rdsadmin.dbo.rds_show_configuration @name='target_s3_arn_for_tlog_copy';
```

Example di aggiornamento a un nuovo bucket Amazon S3 di destinazione

```
exec msdb.dbo.rds_tlog_copy_setup
 @target_s3_arn='arn:aws:s3:::mynewbucket/mynewfolder';
```

Elenco dei backup dei log delle transazioni disponibili

Con RDS per SQL Server, i database configurati per utilizzare il modello di ripristino completo e la conservazione del backup dell'istanza database impostata su uno o più giorni, hanno automaticamente abilitati i backup dei log delle transazioni. Abilitando l'accesso ai backup dei log delle transazioni, potrai copiarli nel tuo bucket Amazon S3 per un massimo di sette giorni.

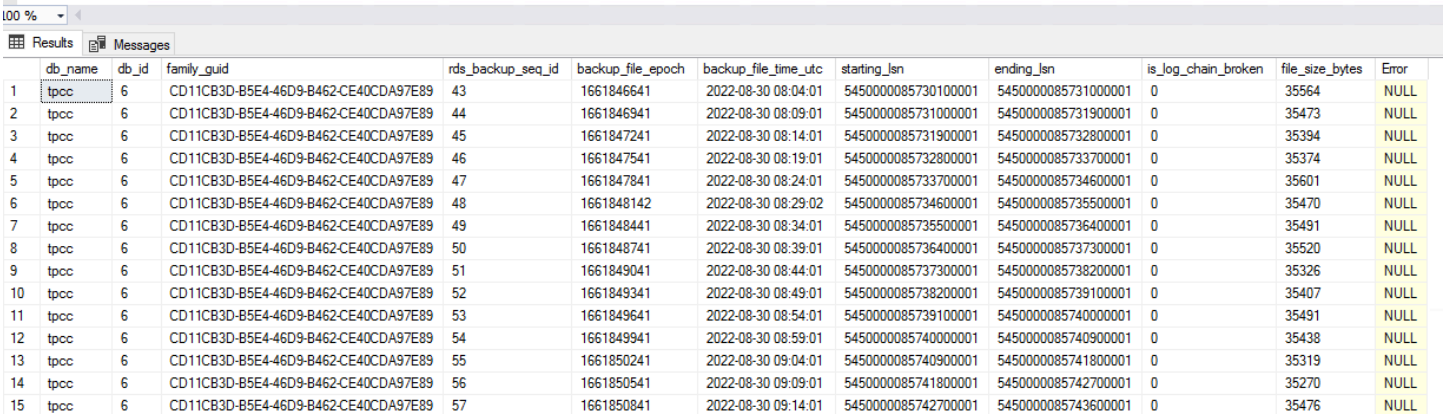
Dopo aver abilitato l'accesso ai backup dei log delle transazioni, puoi iniziare a utilizzarlo per elencare e copiare i file di backup dei log delle transazioni disponibili.

Elenco dei backup dei log delle transazioni

Per elencare tutti i backup dei log delle transazioni disponibili per un singolo database, chiama la funzione `rds_fn_list_tlog_backup_metadata`. È possibile utilizzare una clausola `ORDER BY` o `WHERE` quando si chiama la funzione.

Example di elenco e filtro dei file di backup dei log delle transazioni disponibili

```
SELECT * from msdb.dbo.rds_fn_list_tlog_backup_metadata('mydatabasename');
SELECT * from msdb.dbo.rds_fn_list_tlog_backup_metadata('mydatabasename') WHERE
 rds_backup_seq_id = 3507;
SELECT * from msdb.dbo.rds_fn_list_tlog_backup_metadata('mydatabasename') WHERE
 backup_file_time_utc > '2022-09-15 20:44:01' ORDER BY backup_file_time_utc DESC;
```



100 %

Results Messages

	db_name	db_id	family_guid	rds_backup_seq_id	backup_file_epoch	backup_file_time_utc	starting_lsn	ending_lsn	is_log_chain_broken	file_size_bytes	Error
1	tpcc	6	CD11CB3D-B5E4-46D9-B462-CE40CDA97E89	43	1661846641	2022-08-30 08:04:01	5450000085730100001	5450000085731000001	0	35564	NULL
2	tpcc	6	CD11CB3D-B5E4-46D9-B462-CE40CDA97E89	44	1661846941	2022-08-30 08:09:01	5450000085731000001	5450000085731900001	0	35473	NULL
3	tpcc	6	CD11CB3D-B5E4-46D9-B462-CE40CDA97E89	45	1661847241	2022-08-30 08:14:01	5450000085731900001	5450000085732800001	0	35394	NULL
4	tpcc	6	CD11CB3D-B5E4-46D9-B462-CE40CDA97E89	46	1661847541	2022-08-30 08:19:01	5450000085732800001	5450000085733700001	0	35374	NULL
5	tpcc	6	CD11CB3D-B5E4-46D9-B462-CE40CDA97E89	47	1661847841	2022-08-30 08:24:01	5450000085733700001	5450000085734600001	0	35601	NULL
6	tpcc	6	CD11CB3D-B5E4-46D9-B462-CE40CDA97E89	48	1661848142	2022-08-30 08:29:02	5450000085734600001	5450000085735500001	0	35470	NULL
7	tpcc	6	CD11CB3D-B5E4-46D9-B462-CE40CDA97E89	49	1661848441	2022-08-30 08:34:01	5450000085735500001	5450000085736400001	0	35491	NULL
8	tpcc	6	CD11CB3D-B5E4-46D9-B462-CE40CDA97E89	50	1661848741	2022-08-30 08:39:01	5450000085736400001	5450000085737300001	0	35520	NULL
9	tpcc	6	CD11CB3D-B5E4-46D9-B462-CE40CDA97E89	51	1661849041	2022-08-30 08:44:01	5450000085737300001	5450000085738200001	0	35326	NULL
10	tpcc	6	CD11CB3D-B5E4-46D9-B462-CE40CDA97E89	52	1661849341	2022-08-30 08:49:01	5450000085738200001	5450000085739100001	0	35407	NULL
11	tpcc	6	CD11CB3D-B5E4-46D9-B462-CE40CDA97E89	53	1661849641	2022-08-30 08:54:01	5450000085739100001	5450000085740000001	0	35491	NULL
12	tpcc	6	CD11CB3D-B5E4-46D9-B462-CE40CDA97E89	54	1661849941	2022-08-30 08:59:01	5450000085740000001	5450000085740900001	0	35438	NULL
13	tpcc	6	CD11CB3D-B5E4-46D9-B462-CE40CDA97E89	55	1661850241	2022-08-30 09:04:01	5450000085740900001	5450000085741800001	0	35319	NULL
14	tpcc	6	CD11CB3D-B5E4-46D9-B462-CE40CDA97E89	56	1661850541	2022-08-30 09:09:01	5450000085741800001	5450000085742700001	0	35270	NULL
15	tpcc	6	CD11CB3D-B5E4-46D9-B462-CE40CDA97E89	57	1661850841	2022-08-30 09:14:01	5450000085742700001	5450000085743600001	0	35476	NULL

La funzione `rds_fn_list_tlog_backup_metadata` restituisce il seguente output:

Nome colonna	Tipo di dati	Descrizione
`db_name`	sysname	Il nome del database fornito per elencare i backup dei log delle transazioni.
`db_id`	int	L'identificatore del database interno per il parametro di input `db_name`.
`family_guid`	uniqueidentifier	L'ID univoco del database originale al momento della creazione. Questo valore rimane invariato quando il database viene ripristinato, anche con un nome di database diverso.
`rds_backup_seq_id`	int	L'ID che RDS utilizza internamente per mantenere un numero di sequenza per ogni file di backup dei log delle transazioni.
`backup_file_epoch`	bigint	L'ora in cui è stato generato un file di backup delle transazioni.
`backup_file_time_utc`	datetime	Il valore UTC per il valore `backup_file_epoch` .
`starting_lsn`	numeric(25,0)	Il numero di sequenza di log del primo o dell'ultimo record di log di un file di backup dei log delle transazioni.

Nome colonna	Tipo di dati	Descrizione
`ending_lsn`	numeric(25,0)	Il numero di sequenza di log dell'ultimo o del penultimo record di log di un file di backup dei log delle transazioni.
`is_log_chain_broken`	bit	Un valore booleano che indica se la catena di log è interrotta tra il file di backup dei log delle transazioni corrente e il file di backup dei log delle transazioni precedente.
`file_size_bytes`	bigint	Le dimensioni del set di backup delle transazioni in byte.
`Error`	varchar(4000)	Messaggio di errore se la funzione `rds_fn_list_tlog_backup_metadata` genera un'eccezione. NULL in assenza di eccezioni.

Copia dei backup dei log delle transazioni

Per copiare un set di backup dei log delle transazioni disponibili per un singolo database nel bucket Amazon S3, chiama la stored procedure `rds_tlog_backup_copy_to_S3`. La stored procedure `rds_tlog_backup_copy_to_S3` avvia una nuova attività per copiare i backup dei log delle transazioni.

> **Note**
>
> La store procedure `rds_tlog_backup_copy_to_S3` memorizzata copia i backup dei log delle transazioni senza convalidarli con l'attributo `is_log_chain_broken`. Per questo motivo, è necessario confermare manualmente una catena di log ininterrotta prima di eseguire la stored procedure `rds_tlog_backup_copy_to_S3`. Per ulteriori informazioni, consulta Convalida della catena di log di backup dei log delle transazioni.

Example di utilizzo della stored procedure **`rds_tlog_backup_copy_to_S3`**

```
exec msdb.dbo.rds_tlog_backup_copy_to_S3
 @db_name='mydatabasename',
```

```
[@kms_key_arn='arn:aws:kms:region:account-id:key/key-id'],
[@backup_file_start_time='2022-09-01 01:00:15'],
[@backup_file_end_time='2022-09-01 21:30:45'],
[@starting_lsn=149000000112100001],
[@ending_lsn=149000000120400001],
[@rds_backup_starting_seq_id=5],
[@rds_backup_ending_seq_id=10];
```

Sono disponibili i seguenti parametri di input:

Parametro	Descrizione
`@db_name`	Il nome del database per cui copiare i backup dei log delle transazioni
`@kms_key_arn`	L'ARN della chiave KMS utilizzata per crittografare un'istanza database crittografata nell'archiviazione.
`@backup_file_start_time`	Il timestamp UTC fornito dalla colonna `[backup_file_time_utc]` della funzione `rds_fn_list_tlog_backup_metadata` .
`@backup_file_end_time`	Il timestamp UTC fornito dalla colonna `[backup_file_time_utc]` della funzione `rds_fn_list_tlog_backup_metadata` .
`@starting_lsn`	Il numero di sequenza di log fornito dalla colonna `[starting_lsn]` della funzione `rds_fn_list_tlog_backup_metadata` .
`@ending_lsn`	Il numero di sequenza di log fornito dalla colonna `[ending_lsn]` della funzione `rds_fn_list_tlog_backup_metadata` .
`@rds_backup_starting_seq_id`	L'ID sequenza fornito dalla colonna `[rds_backup_seq_id]` della funzione `rds_fn_list_tlog_backup_metadata` .
`@rds_backup_ending_seq_id`	L'ID sequenza fornito dalla colonna `[rds_backup_seq_id]` della funzione `rds_fn_list_tlog_backup_metadata` .

È possibile specificare un set di parametri relativi all'ora, al numero di sequenza di log o all'ID sequenza. È richiesto un solo set di parametri.

È inoltre possibile specificare un solo parametro in uno qualsiasi dei set. Ad esempio, fornendo un valore solo per il parametro `backup_file_end_time`, tutti i file di backup dei log delle transazioni disponibili prima di quel momento entro il limite di sette giorni verranno copiati nel bucket Amazon S3.

Di seguito sono riportate le combinazioni di parametri di input valide per la stored procedure `rds_tlog_backup_copy_to_S3`.

Parametri forniti	Risultato previsto	
`exec msdb.dbo.rds_tlog_backup_copy_to_S3 @db_name = 'testdb1', @backup_file_start_time='2022-08-23 00:00:00', @backup_file_end_time='2022-08-30 00:00:00';`	Copia i backup dei log delle transazioni degli ultimi sette giorni ed esistenti nell'intervallo fornito compreso tra `backup_file_start_time` e `backup_file_end_time`. In questo esempio, la stored procedure copia i backup dei log delle transazioni generati tra "2022-08-23 00:00:00" e "2022-08-30 00:00:00".	
`exec msdb.dbo.rds_tlog_backup_copy_to_S3`	Copia i backup dei log delle transazioni degli ultimi sette giorni e a partire dalla data specificata in `backup_fi`	

Parametri forniti	Risultato previsto
`@db_name = 'testdb1', @backup_file_start_time='2022-08-23 00:00:00';`	`le_start_time` . In questo esempio, la stored procedure copia i backup dei log delle transazioni dal "2022-08-23 00:00:00" fino al più recente backup dei log delle transazioni.
`exec msdb.dbo.rds_tlog_backup_copy_to_S3 @db_name = 'testdb1', @backup_file_end_time='2022-08-30 00:00:00';`	Copia i backup dei log delle transazioni dagli ultimi sette giorni fino alla data specificata in `backup_file_end_time` . In questo esempio, la stored procedure copia i backup dei log delle transazioni dal "2022-08-23 00:00:00" al "2022-08-30 00:00:00".

Parametri forniti	Risultato previsto	
`exec msdb.dbo.rds_tlog_backup_copy_to_S3` `@db_name='testdb1',` `@starting_lsn =14900000 00040007,` `@ending_lsn = 149000000 0050009;`	Copia i backup dei log delle transazioni disponibili negli ultimi sette giorni e compresi nell'intervallo fornito compreso tra `starting_lsn` e `ending_lsn`. In questo esempio, la stored procedure copia i backup dei log delle transazioni degli ultimi sette giorni con un intervallo di numero di sequenza di log compreso tra 1490000000040007 e 149000000 0050009.	

Parametri forniti	Risultato previsto
`exec msdb.dbo.rds_tlog_backup_copy_to_S3 @db_name='testdb1', @starting_lsn =14900000 00040007;`	Copia i backup dei log delle transazioni disponibili negli ultimi sette giorni, a partire dal numero specificato in `starting_lsn` . In questo esempio, la stored procedure copia i backup dei log delle transazioni dal numero di sequenza di log 1490000000040007 fino al più recente backup dei log delle transazioni.
`exec msdb.dbo.rds_tlog_backup_copy_to_S3 @db_name='testdb1', @ending_lsn =14900000 00050009;`	Copia i backup dei log delle transazioni disponibili negli ultimi sette giorni fino al numero specificato in `ending_lsn` . In questo esempio, la stored procedure copia i backup dei log delle transazioni degli ultimi sette giorni fino al numero di sequenza di log 149000000 0050009.

Parametri forniti	Risultato previsto
`exec msdb.dbo.rds_tlog_backup_copy_to_S3 @db_name='testdb1', @rds_backup_starting_seq_id= 2000, @rds_backup_ending_seq_id= 5000;`	Copia i backup dei log delle transazioni disponibili negli ultimi sette giorni ed esistenti nell'intervallo fornito compreso tra `rds_backup_starting_seq_id` e `rds_backup_ending_seq_id` . In questo esempio, la stored procedure copia i backup dei log delle transazioni a partire dagli ultimi sette giorni e compresi nell'intervallo di ID sequenza di backup RDS fornito, a partire da seq_id 2000 fino a seq_id 5000.

Parametri forniti	Risultato previsto
`exec msdb.dbo.rds_tlog_backup_copy_to_S3 @db_name='testdb1', @rds_backup_starting_seq_id= 2000;`	Copia i backup dei log delle transazioni disponibili negli ultimi sette giorni, a partire dal numero specificato in `rds_backup_starting_seq_id` . In questo esempio, la stored procedure copia i backup dei log delle transazioni a partire da seq_id 2000 fino al più recente backup dei log delle transazioni.
`exec msdb.dbo.rds_tlog_backup_copy_to_S3 @db_name='testdb1', @rds_backup_ending_seq_id= 5000;`	Copia i backup dei log delle transazioni disponibili negli ultimi sette giorni fino al numero specificato in `rds_backup_ending_seq_id` . In questo esempio, la stored procedure copia i backup dei log delle transazioni degli ultimi sette giorni fino a seq_id 5000.

Parametri forniti	Risultato previsto	
`exec msdb.dbo.rds_tlog_backup_copy_to_S3 @db_name='testdb1', @rds_backup_starting_seq_id= 2000; @rds_backup_ending_seq_id= 2000;`	Copia un singolo backup dei log delle transazioni con l'elemento `rds_backup_starting_seq_id` specificato, se disponibile negli ultimi sette giorni. In questo esempio, la stored procedure copia un singolo backup dei log delle transazioni con seq_id 2000, se esistente negli ultimi sette giorni.	

Convalida della catena di log di backup dei log delle transazioni

I database configurati per l'accesso ai backup dei log delle transazioni devono avere la conservazione automatica dei backup abilitata. La conservazione automatica dei backup imposta i database sull'istanza database in base al modello di ripristino FULL. Per supportare il ripristino point-in-time di un database, evita di modificare il modello di ripristino del database per impedire l'interruzione della catena di log. Si consiglia di mantenere il database impostato sul modello di ripristino FULL.

Per convalidare manualmente la catena di log prima di copiare i backup dei log delle transazioni, chiama la funzione `rds_fn_list_tlog_backup_metadata` ed esamina i valori nella colonna `is_log_chain_broken`. Il valore "1" indica che la catena di log è stata interrotta tra il backup dei log corrente e il backup dei log precedente.

L'esempio seguente mostra una catena di log interrotta nell'output della stored procedure `rds_fn_list_tlog_backup_metadata`.

rds_sequence_id	first_lsn	last_lsn	is_log_chain_broken
43	90023	90457	0
44	90457	90985	0
45	90987	92034	1

In una normale catena di log, il valore del numero di sequenza di log per first_lsn per un dato rds_sequence_id deve corrispondere al valore di last_lsn nell'elemento rds_sequence_id precedente. Nell'immagine, il valore rds_sequence_id 45 ha il valore first_lsn 90987, che non corrisponde al valore last_lsn 90985 che dell'elemento rds_sequence_id 44 precedente.

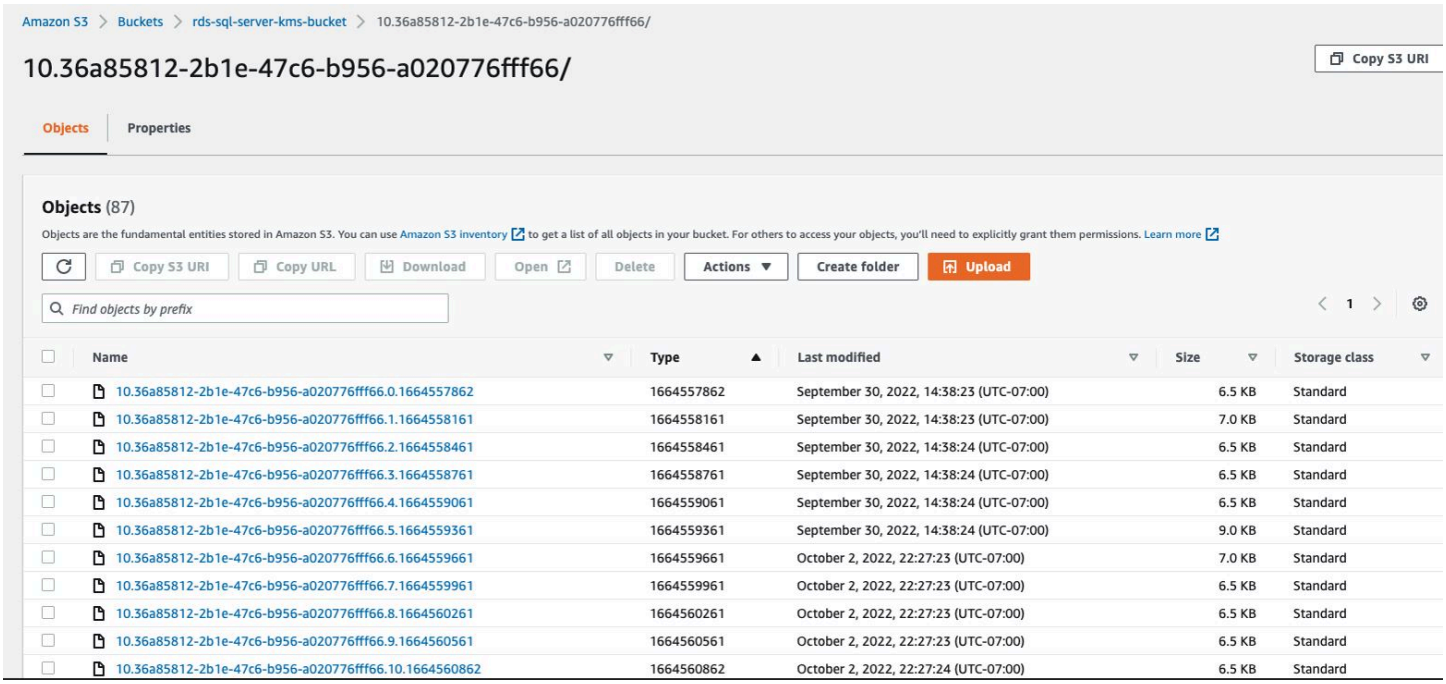
Per ulteriori informazioni sull'architettura dei log delle transazioni di SQL Server e sui numeri di sequenza di log, consulta Architettura logica del log delle transazioni nella documentazione di Microsoft SQL Server.

Struttura di file e cartelle del bucket Amazon S3

I backup dei log delle transazioni hanno la struttura e la convenzione di denominazione standard seguenti in un bucket Amazon S3:

- Viene creata una nuova cartella nel percorso `target_s3_arn` di ogni database con la struttura di denominazione `{db_id}.{family_guid}`.
- All'interno della cartella, i backup dei log delle transazioni hanno la struttura di denominazione dei file `{db_id}.{family_guid}.{rds_backup_seq_id}.{backup_file_epoch}`.
- È possibile visualizzare i dettagli di `family_guid,db_id,rds_backup_seq_id and backup_file_epoch` con la funzione `rds_fn_list_tlog_backup_metadata`.

L'esempio seguente mostra la struttura di cartelle e file di un set di backup dei log delle transazioni in un bucket Amazon S3.



Monitoraggio dello stato delle attività

Per monitorare lo stato delle attività di copia, chiama la stored procedure `rds_task_status`. Se non fornisci alcun parametro, la stored procedure restituisce lo stato di tutte le attività.

Example di utilizzo:

```
exec msdb.dbo.rds_task_status
  @db_name='database_name',
  @task_id=ID_number;
```

I parametri seguenti sono facoltativi:

- `@db_name` – Il nome del database per il quale visualizzare lo stato dell'attività.
- `@task_id` – L'ID dell'attività per la quale visualizzare lo stato.

Example di elenco dello stato per un ID attività specifico:

```
exec msdb.dbo.rds_task_status @task_id=5;
```

Example di elenco dello stato per un database e un'attività specifici:

```
exec msdb.dbo.rds_task_status@db_name='my_database',@task_id=5;
```

Example di elenco di tutte le attività e relativi stati per un database specifico:

```
exec msdb.dbo.rds_task_status @db_name='my_database';
```

Example di elenco di tutte le attività e relativi stati per l'istanza database corrente:

```
exec msdb.dbo.rds_task_status;
```

Annullamento di un'attività

Per annullare un'attività in esecuzione, chiama la stored procedure `rds_cancel_task`.

Example di utilizzo:

```
exec msdb.dbo.rds_cancel_task @task_id=ID_number;
```

Il parametro seguente è obbligatorio:

- `@task_id` – L'ID dell'attività da annullare. Puoi esaminare l'ID attività chiamando la stored procedure `rds_task_status`.

Per ulteriori informazioni sulla visualizzazione e sull'annullamento delle attività in esecuzione, consulta Importazione ed esportazione di database SQL Server mediante backup e ripristino nativi.

Risoluzione dei problemi di accesso ai backup dei log delle transazioni

Di seguito sono elencati i problemi che si potrebbero riscontrare quando si utilizzano le stored procedure per accedere ai backup dei log delle transazioni.

Stored procedure	Messaggio di errore	Problema	Suggerimenti sulla risoluzione dei problemi
rds_tlog_copy_setup	I backup sono disabilitati su questa istanza database. Abilita i backup dell'istanza database con un valore per la conservazione di almeno "1" e riprova.	I backup automatici non sono abilitati per l'istanza database.	La conservazione dei backup dell'istanza database deve essere abilitata con un valore di almeno un giorno. Per ulteriori informazioni sull'abilitazione dei backup automatici e sulla configurazione della conservazione dei backup, consulta Backup retention period (Periodo di retention dei backup).
rds_tlog_copy_setup	Errore durante l'esecuzione della stored procedure rds_tlog_copy_setup. Riconnettiti all'endpoint RDS e riprova.	Si è verificato un errore interno.	Riconnettiti all'endpoint RDS ed esegui nuovamente la stored procedure `rds_tlog_copy_setup` .
rds_tlog_copy_setup	L'esecuzione della stored	La stored procedure è stata avviata in una transazione utilizzando `BEGIN` e `END`.	Evita di utilizzare `BEGIN` e `END` durante l'esecuzione della

Stored procedure	Messaggio di errore	Problema	Suggerimenti sulla risoluzione dei problemi
	procedure rds_tlog_backup_copy_setup in una transazione non è supportata. Verifica che nella sessione non ci siano transazioni aperte e riprova.		stored procedure `rds_tlog_copy_setup` .
rds_tlog_copy_setup	Il nome del bucket S3 per il parametro di input `@target_s3_arn` deve contenere almeno un carattere diverso da spazio.	È stato fornito un valore errato per il parametro di input `@target_s3_arn` .	Assicurati che il parametro di input `@target_s3_arn` specifichi l'ARN completo del bucket Amazon S3.

Stored procedure	Messaggio di errore	Problema	Suggerimenti sulla risoluzione dei problemi
rds_tlog_copy_setup	L'opzione SQLSERVER_BACKUP_RESTORE non è abilitata o è in fase di abilitazione. Abilita l'opzione o riprova più tardi.	L'opzione SQLSERVER_BACKUP_RESTORE non è abilitata sull'istanza database o è stata abilitata ma è in attesa dell'attivazione interna.	Attiva l'opzione SQLSERVER_BACKUP_RESTORE come specificato nella sezione Requisiti. Attendi qualche minuto ed esegui nuovamente la stored procedure rds_tlog_copy_setup .
rds_tlog_copy_setup	L'ARN S3 di destinazione per il parametro di input @target_s3_arn non può essere vuoto o nullo.	È stato fornito un valore NULL per il parametro di input @target_s3_arn oppure il valore non è stato fornito.	Assicurati che il parametro di input @target_s3_arn specifichi l'ARN completo del bucket Amazon S3.

Stored procedure	Messaggio di errore	Problema	Suggerimenti sulla risoluzione dei problemi
rds_tlog_copy_setup	L'ARN S3 di destinazione per il parametro di input `@target_s3_arn` deve iniziare con arn:aws.	Il parametro di input `@target_s3_arn` è stato fornito senza `arn:aws` all'inizio.	Assicurati che il parametro di input `@target_s3_arn` specifichi l'ARN completo del bucket Amazon S3.
rds_tlog_copy_setup	L'ARN S3 di destinazione è già impostato sul valore fornito.	La stored procedure `rds_tlog_copy_setup` veniva precedentemente eseguita ed era configurata con un bucket ARN Amazon S3.	Per modificare il valore del bucket Amazon S3 per l'accesso ai backup dei log delle transazioni, fornisci un valore diverso per `target S3 ARN`.
rds_tlog_copy_setup	Impossibile generare credenziali per abilitare l'accesso ai backup dei log delle transazioni. Conferma l'ARN del percorso S3 fornito con `rds_tlog_copy_setup` e riprova.	Si è verificato un errore non specificato durante la generazione delle credenziali per consentire l'accesso ai backup dei log delle transazioni.	Esamina la configurazione dell'impostazione e riprova.

Stored procedure	Messaggio di errore	Problema	Suggerimenti sulla risoluzione dei problemi
rds_tlog_copy_setup	Non è possibile eseguire la stored procedure rds_tlog_copy_setup se sono presenti attività in sospeso. Attendi il completamento delle attività in sospeso e riprova.	È possibile eseguire solo due attività alla volta. Sono presenti attività in attesa del completamento.	Visualizza le attività in sospeso e attendi che vengano completate. Per ulteriori informazioni sul monitoraggio dello stato delle attività, consulta Monitoraggio dello stato delle attività.
rds_tlog_backup_copy_to_S3	È già stata eseguita un'operazione di copia del file di backup T-log per il database: %s con ID attività: %d, riprova più tardi.	È possibile eseguire una sola operazione di copia alla volta per un determinato database. È presente un'operazione di copia in attesa del completamento.	Visualizza le attività in sospeso e attendi che vengano completate. Per ulteriori informazioni sul monitoraggio dello stato delle attività, consulta Monitoraggio dello stato delle attività.

Stored procedure	Messaggio di errore	Problema	Suggerimenti sulla risoluzione dei problemi
rds_tlog_backup_copy_to_S3	È necessario fornire almeno uno di questi tre set di parametri. SET-1:(@backup_file_start_time, @backup_file_end_time) | SET-2:(@starting_lsn, @ending_lsn) | SET-3:(@rds_backup_starting_seq_id, @rds_backup_ending_seq_id)	Nessuno dei tre set di parametri è stato fornito oppure in un set di parametri fornito manca un parametro obbligatorio.	È possibile specificare i parametri relativi all'ora, al numero di sequenza di log o all'ID sequenza. È richiesto uno di questi tre set di parametri . Per ulteriori informazioni sui parametri obbligatori, consulta Copia dei backup dei log delle transazioni.

Stored procedure	Messaggio di errore	Problema	Suggerimenti sulla risoluzione dei problemi
rds_tlog_backup_copy_to_S3	I backup sono disabilitati sull'istanza. Abilita i backup e riprova tra qualche minuto.	I backup automatici non sono abilitati per l'istanza database.	Per ulteriori informazioni sull'abilitazione dei backup automatici e sulla configurazione della conservazione dei backup, consulta Backup retention period (Periodo di retention dei backup).
rds_tlog_backup_copy_to_S3	Impossibile trovare il database specificato %s.	Il valore fornito per il parametro di input `@db_name` non corrisponde al nome di un database nell'istanza database.	Usa un nome di database corretto. Per elencare tutti i database per nome, esegui `SELECT * from sys.databases`
rds_tlog_backup_copy_to_S3	Impossibile eseguire la stored procedure rds_tlog_backup_copy_to_S3 per i database di sistema SQL Server o il database rdsadmin.	Il valore fornito per il parametro di input `@db_name` corrisponde al nome di un database di sistema SQL Server o al database RDSAdmin.	I seguenti database non possono essere utilizzati per accedere ai backup dei log delle transazioni: `master, model, msdb, tempdb, RDSAdmin.`

Stored procedure	Messaggio di errore	Problema	Suggerimenti sulla risoluzione dei problemi
rds_tlog_backup_copy_to_S3	Il nome di database per il parametro di input @db_name non può essere vuoto o nullo.	Il valore fornito per il parametro di input `@db_name` era vuoto o `NULL`.	Usa un nome di database corretto. Per elencare tutti i database per nome, esegui `SELECT * from sys.databases`
rds_tlog_backup_copy_to_S3	Il periodo di conservazione dei backup dell'istanza database deve essere impostato almeno su 1 per eseguire la stored procedure rds_tlog_backup_copy_setup.	I backup automatici non sono abilitati per l'istanza database.	Per ulteriori informazioni sull'abilitazione dei backup automatici e sulla configurazione della conservazione dei backup, consulta Backup retention period (Periodo di retention dei backup).

Stored procedure	Messaggio di errore	Problema	Suggerimenti sulla risoluzione dei problemi
rds_tlog_backup_copy_to_S3	Errore durante l'esecuzione della stored procedure rds_tlog_backup_copy_to_S3. Riconnettiti all'endpoint RDS e riprova.	Si è verificato un errore interno.	Riconnettiti all'endpoint RDS ed esegui nuovamente la stored procedure `rds_tlog_backup_copy_to_S3` .

Stored procedure	Messaggio di errore	Problema	Suggerimenti sulla risoluzione dei problemi
rds_tlog_backup_copy_to_S3	È possibile fornire uno solo di questi tre set di parametri. SET-1:(@backup_file_start_time, @backup_file_end_time) | SET-2:(@starting_lsn, @ending_lsn) | SET-3:(@rds_backup_starting_seq_id, @rds_backup_ending_seq_id)	Sono stati forniti diversi set di parametri.	È possibile specificare i parametri relativi all'ora, al numero di sequenza di log o all'ID sequenza. È richiesto uno di questi tre set di parametri . Per ulteriori informazioni sui parametri obbligatori, consulta Copia dei backup dei log delle transazioni.

Stored procedure	Messaggio di errore	Problema	Suggerimenti sulla risoluzione dei problemi
rds_tlog_backup_copy_to_S3	L'esecuzione della stored procedure rds_tlog_backup_copy_to_S3 in una transazione non è supportata. Verifica che nella sessione non ci siano transazioni aperte e riprova.	La stored procedure è stata avviata in una transazione utilizzando `BEGIN` e `END`.	Evita di utilizzare `BEGIN` e `END` durante l'esecuzione della stored procedure `rds_tlog_backup_copy_to_S3` .

Stored procedure	Messaggio di errore	Problema	Suggerimenti sulla risoluzione dei problemi
rds_tlog_backup_copy_to_S3	I parametri forniti non rientrano nel periodo di conservazione dei log di backup delle transazioni. Per visualizzare l'elenco dei file di backup dei log delle transazioni disponibili, esegui la funzione rds_fn_list_tlog_backup_metadata.	Non sono disponibili i backup dei log delle transazioni per i parametri di input forniti che rientrano nella finestra di conservazione delle copie.	Riprova con un set di parametri valido. Per ulteriori informazioni sui parametri obbligatori, consulta Copia dei backup dei log delle transazioni.

Stored procedure	Messaggio di errore	Problema	Suggerimenti sulla risoluzione dei problemi
rds_tlog_backup_copy_to_S3	Si è verificato un errore di autorizzazione durante l'elaborazione della richiesta. Assicurati che il bucket si trovi nello stesso account e nella stessa regione dell'istanza database e conferma le autorizzazioni della policy del bucket S3 rispetto al modello della documentazione pubblica.	È stato rilevato un problema relativo al bucket S3 fornito o alle autorizzazioni di policy.	Verifica che la configurazione per l'accesso ai backup dei log delle transazioni sia corretta. Per ulteriori informazioni sui requisiti di configurazione per il bucket S3, consulta Requisiti.

Stored procedure	Messaggio di errore	Problema	Suggerimenti sulla risoluzione dei problemi
rds_tlog_backup_copy_to_S3	L'esecuzione della stored procedure `rds_tlog_backup_copy_to_S3` su un'istanza di replica di lettura RDS non è consentita.	La stored procedure è stata avviata su un'istanza di replica di lettura RDS.	Connettiti all'istanza database primaria RDS per eseguire la stored procedure `rds_tlog_backup_copy_to_S3` .
rds_tlog_backup_copy_to_S3	Il numero di sequenza di log per il parametro di input `@starting_lsn` deve essere inferiore a `@ending_lsn` .	Il valore fornito per il parametro di input `@starting_lsn` era maggiore del valore fornito per il parametro di input `@ending_lsn` .	Assicurati che il valore fornito per il parametro di input `@starting_lsn` sia inferiore al valore fornito per il parametro di input `@ending_lsn` .

Stored procedure	Messaggio di errore	Problema	Suggerimenti sulla risoluzione dei problemi
rds_tlog_backup_copy_to_S3	La stored procedure `rds_tlog_backup_copy_to_S3` può essere eseguita solo dai membri con il ruolo `db_owner` nel database di origine.	Il ruolo `db_owner` non è stato concesso all'account che tenta di eseguire la stored procedure `rds_tlog_backup_copy_to_S3` sul provider `db_name`.	Assicurati che l'account che esegue la stored procedure sia autorizzato con il ruolo `db_owner` per il `db_name` specificato.
rds_tlog_backup_copy_to_S3	L'ID sequenza per il parametro di input `@rds_backup_starting_seq_id` deve essere minore o uguale a `@rds_backup_ending_seq_id` .	Il valore fornito per il parametro di input `@rds_backup_starting_seq_id` era maggiore del valore fornito per il parametro di input `@rds_backup_ending_seq_id` .	Assicurati che il valore fornito per il parametro di input `@rds_backup_starting_seq_id` sia inferiore al valore fornito per il parametro di input `@rds_backup_ending_seq_id` .

Stored procedure	Messaggio di errore	Problema	Suggerimenti sulla risoluzione dei problemi
rds_tlog_backup_copy_to_S3	L'opzione SQLSERVER_BACKUP_RESTORE non è abilitata o è in fase di abilitazione. Abilita l'opzione o riprova più tardi.	L'opzione SQLSERVER_BACKUP_RESTORE non è abilitata sull'istanza database o è stata abilitata ma è in attesa dell'attivazione interna.	Attiva l'opzione SQLSERVER_BACKUP_RESTORE come specificato nella sezione Requisiti. Attendi qualche minuto ed esegui nuovamente la stored procedure rds_tlog_backup_copy_to_S3 .
rds_tlog_backup_copy_to_S3	L'ora di inizio del parametro di input @backup_file_start_time deve essere inferiore a @backup_file_end_time .	Il valore fornito per il parametro di input @backup_file_start_time era maggiore del valore fornito per il parametro di input @backup_file_end_time .	Assicurati che il valore fornito per il parametro di input @backup_file_start_time sia inferiore al valore fornito per il parametro di input @backup_file_end_time .

Stored procedure	Messaggio di errore	Problema	Suggerimenti sulla risoluzione dei problemi
rds_tlog_backup_copy_to_S3	Impossibile elaborare la richiesta a causa della mancanza di accesso. Controlla le impostazioni e le autorizzazioni per la funzionalità.	Potrebbe esserci un problema con le autorizzazioni del bucket Amazon S3 oppure il bucket Amazon S3 fornito si trova in un altro account o un'altra regione.	Assicurati che le autorizzazioni della policy dei bucket di Amazon S3 siano autorizzate a consentire l'accesso RDS. Assicurati che il bucket Amazon S3 si trovi nello stesso account e nella stessa regione dell'istanza database.
rds_tlog_backup_copy_to_S3	Non è possibile fornire l'ARN di una chiave KMS come parametro di input alla stored procedure per le istanze che non sono crittografate nell'archiviazione.	Quando la crittografia dell'archiviazione non è abilitata sull'istanza database, il parametro di input `@kms_key_arn` non deve essere fornito.	Non fornire un parametro di input per `@kms_key_arn` .

Stored procedure	Messaggio di errore	Problema	Suggerimenti sulla risoluzione dei problemi
rds_tlog_backup_copy_to_S3	È necessario fornire l'ARN di una chiave KMS come parametro di input per la stored procedure delle istanze crittografate nell'archiviazione.	Quando la crittografia dell'archiviazione è abilitata sull'istanza database, è necessario fornire il parametro di input `@kms_key_arn` .	Fornisci un parametro di input per `@kms_key_arn` con un valore che corrisponda all'ARN del bucket Amazon S3 da utilizzare per i backup dei log delle transazioni.
rds_tlog_backup_copy_to_S3	È necessario eseguire la stored procedure `rds_tlog_copy_setup` e impostare `@target_s3_arn` , prima di eseguire la stored procedure `rds_tlog_backup_copy_to_S3` .	La procedura di impostazione dell'accesso ai backup dei log delle transazioni non è stata completata prima di tentare di eseguire la stored procedure `rds_tlog_backup_copy_to_S3` .	Esegui la stored procedure `rds_tlog_copy_setup` prima di eseguire la stored procedure `rds_tlog_backup_copy_to_S3` . Per ulteriori informazioni sull'esecuzione della procedura di impostazione per l'accesso ai backup dei log delle transazioni, consulta Configurazione dell'accesso ai backup dei log delle transazioni.

Opzioni per il motore di database di Microsoft SQL Server

In questa sezione vengono descritte le opzioni disponibili per le istanze Amazon RDS che eseguono il motore di database di Microsoft SQL Server. Per abilitare queste opzioni, dovrai aggiungerle a un gruppo di opzioni e quindi associare il gruppo di opzioni all'istanza database. Per ulteriori informazioni, consulta Uso di gruppi di opzioni.

Se cerchi caratteristiche opzionali che non sono aggiunte tramite i gruppi di opzioni RDS (ad esempio SSL, Windows Authentication e Amazon S3 integration), consulta Funzionalità opzionali per Microsoft SQL Server su Amazon RDS.

Amazon RDS supporta le seguenti opzioni per le istanze database Microsoft SQL Server.

Opzione	ID opzione	Edizioni del motore
Server collegati con Oracle OLEDB	`OLEDB_ORACLE`	SQL Server Enterprise Edition SQL Server Standard Edition
Backup nativo e ripristino	`SQLSERVER_BACKUP_RESTORE`	SQL Server Enterprise Edition SQL Server Standard Edition SQL Server Web Edition SQL Server Express Edition
Transparent Data Encryption	`TRANSPARENT_DATA_ENCRYPTION` (console RDS)	Edizione aziendale di SQL Server 2014-2022 Edizione standard di SQL Server 2022

Opzione	ID opzione	Edizioni del motore
	TDE (AWS CLI e API RDS)	
Audit in SQL Server	SQLSERVER_AUDIT	In RDS, a partire da SQL Server 2014, tutte le edizioni di SQL Server supportano gli audit a livello di server; inoltre, l'edizione Enterprise supporta anche gli audit a livello di database. A partire da SQL Server 2016 (13.x) SP1, tutte le edizioni supportano gli audit sia a livello di server che di database. Per ulteriori informazioni, consulta Audit in SQL Server (Motore di database) nella documentazione di SQL Server.
SQL Server Analysis Services (SSAS)	SSAS	SQL Server Enterprise Edition SQL Server Standard Edition

Opzione	ID opzione	Edizioni del motore
SQL Server Integration Services (SSIS)	SSIS	SQL Server Enterprise Edition SQL Server Standard Edition
SQL Server Reporting Service (SSRS)	SSRS	SQL Server Enterprise Edition SQL Server Standard Edition
Microsoft Distributed Transaction Coordinator	MSDTC	In RDS, a partire da SQL Server 2014, tutte le edizioni di SQL Server supportano le transazioni distribuite.

Elenco delle opzioni disponibili per le versioni e le edizioni di SQL Server

È possibile utilizzare il comando `describe-option-group-options` AWS CLI per elencare le opzioni disponibili per le versioni e le edizioni di SQL Server e le impostazioni per tali opzioni.

Nell'esempio seguente vengono illustrate le opzioni e le impostazioni delle opzioni per SQL Server 2019 Enterprise Edition. L'opzione `--engine-name` è obbligatoria.

```
aws rds describe-option-group-options --engine-name sqlserver-ee --major-engine-version
 15.00
```

L'output è simile a quello riportato di seguito.

```
{
    "OptionGroupOptions": [
        {
            "Name": "MSDTC",
            "Description": "Microsoft Distributed Transaction Coordinator",
```

```
            "EngineName": "sqlserver-ee",
            "MajorEngineVersion": "15.00",
            "MinimumRequiredMinorEngineVersion": "4043.16.v1",
            "PortRequired": true,
            "DefaultPort": 5000,
            "OptionsDependedOn": [],
            "OptionsConflictsWith": [],
            "Persistent": false,
            "Permanent": false,
            "RequiresAutoMinorEngineVersionUpgrade": false,
            "VpcOnly": false,
            "OptionGroupOptionSettings": [
                {
                    "SettingName": "ENABLE_SNA_LU",
                    "SettingDescription": "Enable support for SNA LU protocol",
                    "DefaultValue": "true",
                    "ApplyType": "DYNAMIC",
                    "AllowedValues": "true,false",
                    "IsModifiable": true,
                    "IsRequired": false,
                    "MinimumEngineVersionPerAllowedValue": []
                },
        ...

        {
            "Name": "TDE",
            "Description": "SQL Server - Transparent Data Encryption",
            "EngineName": "sqlserver-ee",
            "MajorEngineVersion": "15.00",
            "MinimumRequiredMinorEngineVersion": "4043.16.v1",
            "PortRequired": false,
            "OptionsDependedOn": [],
            "OptionsConflictsWith": [],
            "Persistent": true,
            "Permanent": false,
            "RequiresAutoMinorEngineVersionUpgrade": false,
            "VpcOnly": false,
            "OptionGroupOptionSettings": []
        }
    ]
}
```

Supporto per server collegati con Oracle OLEDB in Amazon RDS per SQL Server

I server collegati con Oracle Provider for OLEDB su RDS per SQL Server consentono di accedere alle origini dei dati esterne in un database Oracle. È possibile leggere dati da origini dei dati Oracle remote ed eseguire comandi su server di database Oracle remoti all'esterno dell'istanza database RDS per SQL Server. Utilizzando i server collegati con Oracle OLEDB, è possibile:

- Accedere direttamente a origini dei dati diverse da SQL Server
- Eseguire query su origini dei dati Oracle diverse con la stessa query senza spostare i dati
- Eseguire query, aggiornamenti, comandi e transazioni distribuiti sulle origini dei dati in un ecosistema aziendale
- Integrare le connessioni a un database Oracle dall'interno della suite Microsoft Business Intelligence (SSIS, SSRS, SSAS)
- Eseguire la migrazione da un database Oracle a RDS per SQL Server

È possibile attivare uno o più server collegati per Oracle su un'istanza database RDS per SQL Server nuova o esistente. Quindi puoi integrare le origini dei dati Oracle esterne nella tua istanza database.

Indice

Versioni e regioni supportate

RDS per SQL Server supporta i server collegati con Oracle OLEDB in tutte le regioni per SQL Server Standard ed Enterprise Edition nelle seguenti versioni:

- SQL Server 2022, tutte le versioni
- SQL Server 2019, tutte le versioni
- SQL Server 2017, tutte le versioni

I server collegati con Oracle OLEDB sono supportati per le seguenti versioni di Oracle Database:

- Oracle Database 21c, tutte le versioni
- Oracle Database 19c, tutte le versioni
- Oracle Database 18c, tutte le versioni

Limitazioni e consigli

Tieni presente le seguenti limitazioni e raccomandazioni che si applicano ai server collegati con Oracle OLEDB:

- Consenti il traffico di rete aggiungendo la porta TCP applicabile nel gruppo di sicurezza per ogni istanza database RDS per SQL Server. Ad esempio, se stai configurando un server collegato tra un'istanza database Oracle EC2 e un'istanza database RDS per SQL Server, devi consentire il traffico dall'indirizzo IP dell'istanza database Oracle EC2. È inoltre necessario consentire il traffico sulla porta utilizzata da SQL Server per ascoltare le comunicazioni con il database. Per ulteriori informazioni sui gruppi di sicurezza, consulta Controllo dell'accesso con i gruppi di sicurezza.
- Esegui un riavvio dell'istanza database RDS per SQL Server dopo aver attivato, disattivato o modificato l'opzione `OLEDB_ORACLE` nel tuo gruppo di opzioni. Lo stato del gruppo di opzioni è `pending_reboot` per questi eventi ed è obbligatorio.
- È supportata solo l'autenticazione semplice con un nome utente e una password per l'origine dei dati Oracle.
- I driver Open Database Connectivity (ODBC) non sono supportati. È supportata solo la versione più recente del driver OLEDB.
- Le transazioni distribuite (XA) sono supportate. Per attivare le transazioni distribuite, attiva l'opzione `MSDTC` nel gruppo di opzioni per la tua istanza database e assicurati che le transazioni XA siano attivate. Per ulteriori informazioni, consulta Supporto per Microsoft Distributed Transaction Coordinator in RDS per SQL Server.
- La creazione di nomi di origine dei dati (DSN) da utilizzare come scelta rapida per una stringa di connessione non è supportata.

- Il tracciamento dei driver OLEDB non è supportato. È possibile utilizzare gli eventi estesi SQL Server per tracciare gli eventi OLEDB. Per ulteriori informazioni, consulta Set up Extended Events in RDS for SQL Server (Impostazione degli eventi estesi in RDS per SQL Server).
- L'accesso alla cartella dei cataloghi per un server collegato Oracle non è supportato con SQL Server Management Studio (SSMS).

Attivazione di server collegati con Oracle

Attiva i server collegati con Oracle aggiungendo l' opzione `OLEDB_ORACLE` all'istanza database RDS per SQL Server. Utilizzare il seguente processo:

1. Creare un nuovo gruppo di opzioni oppure utilizzare un gruppo di opzioni esistente.
2. Aggiungere l'opzione `OLEDB_ORACLE` al gruppo di opzioni.
3. Scegli la versione del driver OLEDB da usare.
4. Associare il gruppo di opzioni a questa istanza database.
5. Riavvia l'istanza database.

Creazione del gruppo di opzioni per OLEDB_ORACLE

Per utilizzare i server collegati con Oracle, crea un gruppo di opzioni o modifica un gruppo di opzioni che corrisponda all'edizione di SQL Server e alla versione dell'istanza database che vuoi utilizzare. Per completare questa procedura, utilizza la AWS Management Console o AWS CLI.

Console

La seguente procedura crea un gruppo di opzioni per SQL Server Standard Edition 2019.

Per creare il gruppo di opzioni

1. Accedi alla AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel pannello di navigazione scegliere Option groups (Gruppi di opzioni).
3. Seleziona Create group (Crea gruppo).
4. Nella finestra Create option group (Crea gruppo di opzioni) eseguire queste operazioni:

a. Per Nome, immettere un nome per il gruppo di opzioni che sia univoco all'interno dell'account AWS, ad esempio **`oracle-oledb-se-2019`**. Il nome può includere solo lettere, cifre e trattini.

b. Per Descrizione, immettere una breve descrizione del gruppo di opzioni, ad esempio **`OLEDB_ORACLE option group for SQL Server SE 2019`**. La descrizione viene usata per la visualizzazione.

c. Per Engine (Motore), scegliere sqlserver-se.

d. Per Major engine version (Versione del motore principale), scegli 15.00.

5. Scegli Crea.

CLI

La seguente procedura crea un gruppo di opzioni per SQL Server Standard Edition 2019.

Per creare il gruppo di opzioni

- Eseguire uno dei seguenti comandi.

  Example

  Per LinuxmacOS, oUnix:

```
aws rds create-option-group \
    --option-group-name oracle-oledb-se-2019 \
    --engine-name sqlserver-se \
    --major-engine-version 15.00 \
    --option-group-description "OLEDB_ORACLE option group for SQL Server SE 2019"
```

  Per Windows:

```
aws rds create-option-group ^
    --option-group-name oracle-oledb-se-2019 ^
    --engine-name sqlserver-se ^
    --major-engine-version 15.00 ^
    --option-group-description "OLEDB_ORACLE option group for SQL Server SE 2019"
```

Aggiunta dell'opzione **OLEDB_ORACLE** al gruppo di opzioni

Utilizzare la AWS Management Console o l'AWS CLI per aggiungere l'opzione OLEDB_ORACLE al gruppo di opzioni.

Console

Per aggiungere l'opzione OLEDB_ORACLE

1. Accedi alla AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel pannello di navigazione scegliere Option groups (Gruppi di opzioni).
3. Scegliete il gruppo di opzioni che avete appena creato, che in questo esempio è oracle-oledb-se-2019.
4. Scegliere Add option (Aggiungi opzione).
5. In Option details (Dettagli opzione), scegli OLEDB_ORACLE per Option name (Nome opzione).
6. In Scheduling (Pianificazione), scegliere se aggiungere l'opzione immediatamente o alla finestra di manutenzione successiva.
7. Scegliere Add option (Aggiungi opzione).

CLI

Per aggiungere l'opzione OLEDB_ORACLE

- Aggiungere l'opzione OLEDB_ORACLE al gruppo di opzioni.

  Example

  Per LinuxmacOS, oUnix:

  ```
  aws rds add-option-to-option-group \
      --option-group-name oracle-oledb-se-2019 \
      --options OptionName=OLEDB_ORACLE \
      --apply-immediately
  ```

  Per Windows:

  ```
  aws rds add-option-to-option-group ^
      --option-group-name oracle-oledb-se-2019 ^
  ```

```
    --options OptionName=OLEDB_ORACLE ^
    --apply-immediately
```

Associazione del gruppo di opzioni all'istanza database

Per associare il gruppo di opzioni `OLEDB_ORACLE` e il gruppo di parametri all'istanza database, utilizza la AWS Management Console o AWS CLI

Console

Per completare l'attivazione dei server collegati per Oracle, associa il gruppo di opzioni `OLEDB_ORACLE` a un'istanza database nuova o esistente:

- Per una nuova istanza database, associarli all'avvio dell'istanza. Per ulteriori informazioni, consulta Creazione di un'istanza database Amazon RDS.
- Per un'istanza database esistente, associarli modificando l'istanza. Per ulteriori informazioni, consulta Modifica di un'istanza database Amazon RDS.

CLI

Puoi associare il gruppo di opzioni `OLEDB_ORACLE` e il gruppo di parametri a un'istanza database nuova o esistente.

Per creare un'istanza con il gruppo di opzioni **`OLEDB_ORACLE`** e il gruppo di parametri

- Specificare lo stesso tipo di motore database e la versione principale utilizzata durante la creazione del gruppo di opzioni.

  Example

  Per LinuxmacOS, oUnix:

```
aws rds create-db-instance \
    --db-instance-identifier mytestsqlserveroracleoledbinstance \
    --db-instance-class db.m5.2xlarge \
    --engine sqlserver-se \
    --engine-version 15.0.4236.7.v1 \
    --allocated-storage 100 \
    --manage-master-user-password \
    --master-username admin \
```

```
    --storage-type gp2 \
    --license-model li \
    --domain-iam-role-name my-directory-iam-role \
    --domain my-domain-id \
    --option-group-name oracle-oledb-se-2019 \
    --db-parameter-group-name my-parameter-group-name
```

Per Windows:

```
aws rds create-db-instance ^
    --db-instance-identifier mytestsqlserveroracleoledbinstance ^
    --db-instance-class db.m5.2xlarge ^
    --engine sqlserver-se ^
    --engine-version 15.0.4236.7.v1 ^
    --allocated-storage 100 ^
    --manage-master-user-password ^
    --master-username admin ^
    --storage-type gp2 ^
    --license-model li ^
    --domain-iam-role-name my-directory-iam-role ^
    --domain my-domain-id ^
    --option-group-name oracle-oledb-se-2019 ^
    --db-parameter-group-name my-parameter-group-name
```

Per modificare un'istanza e associare il gruppo di opzioni **OLEDB_ORACLE**

- Eseguire uno dei seguenti comandi.

  Example

  Per LinuxmacOS, oUnix:

```
aws rds modify-db-instance \
    --db-instance-identifier mytestsqlserveroracleoledbinstance \
    --option-group-name oracle-oledb-se-2019 \
    --db-parameter-group-name my-parameter-group-name \
    --apply-immediately
```

  Per Windows:

```
aws rds modify-db-instance ^
```

```
    --db-instance-identifier mytestsqlserveroracleoledbinstance ^
    --option-group-name oracle-oledb-se-2019 ^
    --db-parameter-group-name my-parameter-group-name ^
    --apply-immediately
```

Modifica delle proprietà del provider OLEDB

È possibile visualizzare e modificare le proprietà del provider OLEDB. Solo l'utente `master` può eseguire questa attività. Tutti i server collegati per Oracle creati sull'istanza database utilizzano le stesse proprietà del provider OLEDB. Chiama la stored procedure `sp_MSset_oledb_prop` per modificare le proprietà del provider OLEDB.

Per modificare le proprietà del provider OLEDB

```
USE [master]
GO
EXEC sp_MSset_oledb_prop N'OraOLEDB.Oracle', N'AllowInProcess', 1
EXEC sp_MSset_oledb_prop N'OraOLEDB.Oracle', N'DynamicParameters', 0
GO
```

È possibile modificare le seguenti proprietà:

Nome proprietà	Valore consigliato (1 = attivato, 0 = disattivato)	Descrizione
`Dynamic parameter`	1	Consente segnaposti SQL (rappresentati da "?") nelle query parametrizzate.
`Nested queries`	1	Consente istruzioni annidate `SELECT` nella clausola `FROM`, ad esempio le query secondarie.
`Level zero only`	0	Solo le interfacce OLEDB di livello base vengono chiamate sul provider.
`Allow inprocess`	1	Se attivato, Microsoft SQL Server consente di creare l'istanza del provider come server in

Nome proprietà	Valore consigliato (1 = attivato, 0 = disattivato)	Descrizione
		esecuzione. Imposta questa proprietà su 1 per utilizzare i server collegati Oracle.
`Non transacted updates`	0	Se diverso da zero, SQL Server consente gli aggiornamenti.
`Index as access path`	False	Se diverso da zero, SQL Server tenta di utilizzare gli indici del provider per recuperare i dati.
`Disallow adhoc access`	False	Se impostato, SQL Server non consente l'esecuzione di query pass-through sul provider OLEDB. Sebbene questa opzione possa essere selezionata, a volte è opportuno eseguire query pass-through.
`Supports LIKE operator`	1	Indica che il provider supporta le query utilizzando la parola chiave LIKE.

Modifica delle proprietà del driver OLEDB

È possibile visualizzare e modificare le proprietà del driver OLEDB per la creazione di un server collegato per Oracle. Solo l'utente `master` può eseguire questa attività. Le proprietà del driver definiscono il modo in cui il driver OLEDB gestisce i dati quando utilizza un'origine dei dati Oracle remota. Le proprietà del driver sono specifiche per ogni server collegato Oracle creato nell'istanza database. Chiama la stored procedure `master.dbo.sp_addlinkedserver` per modificare le proprietà del driver OLEDB.

Esempio: per creare un server collegato e modificare la proprietà `FetchSize` del driver OLEDB

```
EXEC master.dbo.sp_addlinkedserver
@server = N'Oracle_link2',
@srvproduct=N'Oracle',
@provider=N'OraOLEDB.Oracle',
@datasrc=N'my-oracle-test.cnetsipka.us-west-2.rds.amazonaws.com:1521/ORCL',
@provstr='FetchSize=200'
```

```
GO
```

```
EXEC master.dbo.sp_addlinkedsrvlogin
@rmtsrvname=N'Oracle_link2',
@useself=N'False',
@locallogin=NULL,
@rmtuser=N'master',
@rmtpassword='Test#1234'
GO
```

> **Note**
>
> Specifica una password diversa dal prompt mostrato qui come best practice per la sicurezza.

Disattivazione dei server collegati con Oracle

Per disattivare i server collegati con Oracle, rimuovi l'opzione `OLEDB_ORACLE` dal gruppo di opzioni.

> **Important**
>
> La rimozione dell'opzione non elimina le configurazioni del server collegato esistente nell'istanza database. È necessario eliminarle manualmente per rimuoverle dall'istanza database.
> È possibile riattivare l'opzione `OLEDB_ORACLE` dopo la rimozione per riutilizzare le configurazioni del server collegato precedentemente presenti nell'istanza database.

Console

La procedura seguente rimuove l'opzione `OLEDB_ORACLE`.

Per rimuovere l'opzione OLEDB_ORACLE dal suo gruppo di opzioni

1. Accedi alla AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel pannello di navigazione scegliere Option groups (Gruppi di opzioni).

3. Scegliere il gruppo di opzioni con l'opzione `OLEDB_ORACLE` (`oracle-oledb-se-2019` negli esempi precedenti).
4. Scegliere Delete option (Elimina opzione).
5. In Deletion options (Opzioni di eliminazione), scegli OLEDB_ORACLE per Options to delete (Opzioni da eliminare).
6. In Apply immediately (Applica immediatamente), scegli Yes (Sì) per eliminare immediatamente l'opzione oppure No per eliminarla nella finestra di manutenzione successiva.
7. Scegli Elimina.

CLI

La procedura seguente rimuove l'opzione `OLEDB_ORACLE`.

Per rimuovere l'opzione OLEDB_ORACLE dal suo gruppo di opzioni

- Eseguire uno dei seguenti comandi.

  Example

  Per LinuxmacOS, oUnix:

```
aws rds remove-option-from-option-group \
    --option-group-name oracle-oledb-se-2019 \
    --options OLEDB_ORACLE \
    --apply-immediately
```

  Per Windows:

```
aws rds remove-option-from-option-group ^
    --option-group-name oracle-oledb-se-2019 ^
    --options OLEDB_ORACLE ^
    --apply-immediately
```

Supporto per backup nativo e ripristino in SQL Server

Utilizzando il backup nativo e il ripristino per i database di SQL Server, puoi creare un backup differenziale o completo del database locale e archiviare i file di backup in Amazon S3. È quindi possibile ripristinarli in un'istanza database Amazon RDS che esegue SQL Server. Si può anche eseguire il backup di un database RDS for SQL Server, archiviarlo in Amazon S3 e ripristinarlo in altre posizioni. Inoltre, il backup può essere ripristinato in un server locale o in un'altra istanza database Amazon RDS che esegue SQL Server. Per ulteriori informazioni, consulta Importazione ed esportazione di database SQL Server mediante backup e ripristino nativi.

Amazon RDS supporta il backup nativo e il ripristino dei database di Microsoft SQL Server tramite file di backup differenziali e completi (file .bak).

Aggiunta dell'opzione Native Backup and Restore (Backup nativo e ripristino)

Di seguito è riportato il processo generale per aggiungere l'opzione di backup nativo e ripristino a un'istanza database:

1. Creare un nuovo gruppo di opzioni oppure copiare o modificare un gruppo di opzioni esistente.
2. Aggiungere l'opzione `SQLSERVER_BACKUP_RESTORE` al gruppo di opzioni.
3. Associare un ruolo AWS Identity and Access Management (IAM) all'opzione. Il ruolo IAM deve avere accesso a un bucket S3 per archiviare i backup del database.

   In altre parole, nell'impostazione dell'opzione deve essere specificato un Amazon Resource Name (ARN) valido in formato `arn:aws:iam::`*`account-id`*`:role/`*`role-name`*. Per ulteriori informazioni, consulta Amazon Resource Names (ARNs) nella Riferimenti generali di AWS.

   Il ruolo IAM deve inoltre avere una relazione di fiducia e una policy di autorizzazione allegata. La relazione di attendibilità consente a RDS di assumere il ruolo e la policy delle autorizzazioni definisce le azioni che il ruolo può eseguire. Per ulteriori informazioni, consulta Creazione manuale di un ruolo IAM per backup e ripristino nativi.
4. Associare il gruppo di opzioni a questa istanza database.

Dopo aver aggiunto l'opzione di backup nativo e ripristino, non è necessario riavviare l'istanza database. Non appena il gruppo di opzioni è attivo, potrai iniziare immediatamente a eseguire il backup e il ripristino.

Console

Per aggiungere l'opzione di backup nativo e ripristino

1. Accedi alla AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione scegliere Option groups (Gruppi di opzioni).
3. Creare un nuovo gruppo di opzioni oppure utilizzare un gruppo di opzioni esistente. Per informazioni su come creare un gruppo di opzioni database personalizzato, consultare Creazione di un gruppo di opzioni.

   Per utilizzare un gruppo di opzioni esistente, passare alla fase successiva.
4. Aggiungere l'opzione SQLSERVER_BACKUP_RESTORE al gruppo di opzioni. Per ulteriori informazioni sull'aggiunta di opzioni, consulta Aggiunta di un'opzione a un gruppo di opzioni.
5. Scegliere una delle seguenti operazioni:

   - Per utilizzare un ruolo IAM esistente e le impostazioni Amazon S3, scegliere un ruolo IAM esistente per IAM Role (Ruolo IAM). Se si sceglie un ruolo IAM esistente, RDS utilizza le impostazioni Amazon S3 configurate per tale ruolo.
   - Per creare un nuovo ruolo e configurare nuove impostazioni Amazon S3, procedere come segue:

     1. Per Ruolo IAM, scegliere Crea un nuovo ruolo.
     2. Per S3 bucket name (Nome bucket S3), scegli un bucket S3 esistente dall'elenco.
     3. Per S3 folder path prefix (optional) (Prefisso percorso cartella S3, facoltativo), specificare un prefisso da utilizzare per i file archiviati nel bucket Amazon S3.

        Questo prefisso può includere un percorso del file, ma non è obbligatorio. Se si include un prefisso, RDS lo aggiunge a tutti i file di backup. RDS utilizza quindi il prefisso durante il ripristino per identificare i file correlati e ignorare quelli irrilevanti. Ad esempio, si può utilizzare il bucket S3 per scopi diversi dal mantenimento dei file di backup. In questo caso, è possibile utilizzare il prefisso per far eseguire a RDS un backup nativo e un ripristino solo su una specifica cartella e sulle relative sottocartelle.

        Se il prefisso viene lasciato vuoto, RDS non lo utilizza per identificare i file per il backup o per il ripristino. Di conseguenza, durante un ripristino di più file, RDS prova a ripristinare tutti i file in tutte le cartelle del bucket S3.

4. In Enable Encryption (Abilita crittografia) scegliere Yes (Sì) per crittografare il file di backup. Lasciare deselezionata la casella di controllo (impostazione predefinita) per non crittografare il file di backup.

   Se hai scelto Abilita crittografia, seleziona una chiave di crittografia per AWS KMS key. Per ulteriori informazioni sulle chiavi di crittografia, consulta Nozioni di base nella Guida per gli sviluppatori di AWS Key Management Service.

6. Scegliere Add option (Aggiungi opzione).
7. Applicare il gruppo di opzioni a un'istanza database nuova o esistente:

   - Per una nuova istanza database, applicare il gruppo di opzioni quando viene avviata l'istanza. Per ulteriori informazioni, consulta Creazione di un'istanza database Amazon RDS.
   - Per un'istanza database esistente, applicare il gruppo di opzioni modificando l'istanza e collegando il nuovo gruppo di opzioni. Per ulteriori informazioni, consulta Modifica di un'istanza database Amazon RDS.

CLI

Questa procedura prevede i seguenti presupposti:

- Si sta aggiungendo l'opzione SQLSERVER_BACKUP_RESTORE a un gruppo di opzioni già esistente. Per ulteriori informazioni sull'aggiunta di opzioni, consulta Aggiunta di un'opzione a un gruppo di opzioni.
- Si sta associando l'opzione a un ruolo IAM già esistente e che ha accesso a un bucket S3 per archiviare i backup.
- Si sta applicando il gruppo di opzioni a un'istanza database già esistente. Per ulteriori informazioni, consulta Modifica di un'istanza database Amazon RDS.

Per aggiungere l'opzione di backup nativo e ripristino

1. Aggiungere l'opzione `SQLSERVER_BACKUP_RESTORE` al gruppo di opzioni.

   Example

   Per LinuxmacOS, oUnix:

   ```
   aws rds add-option-to-option-group \
   ```

```
 --apply-immediately \
 --option-group-name mybackupgroup \
 --options "OptionName=SQLSERVER_BACKUP_RESTORE, \
   OptionSettings=[{Name=IAM_ROLE_ARN,Value=arn:aws:iam::account-id:role/role-
name}]"
```

Per Windows:

```
aws rds add-option-to-option-group ^
 --option-group-name mybackupgroup ^
 --options "[{\"OptionName\": \"SQLSERVER_BACKUP_RESTORE\", ^
 \"OptionSettings\": [{\"Name\": \"IAM_ROLE_ARN\", ^
 \"Value\": \"arn:aws:iam::account-id:role/role-name"}]}]" ^
 --apply-immediately
```

> **Note**
>
> Quando usi il prompt comandi di Windows, non devi inserire le doppie virgolette (") nel codice JSON precedendole con il backslash (\).

2. Applicare il gruppo di opzioni all'istanza database.

   Example

   Per LinuxmacOS, oUnix:

```
aws rds modify-db-instance \
 --db-instance-identifier mydbinstance \
 --option-group-name mybackupgroup \
 --apply-immediately
```

   Per Windows:

```
aws rds modify-db-instance ^
 --db-instance-identifier mydbinstance ^
 --option-group-name mybackupgroup ^
 --apply-immediately
```

Modifica delle impostazioni dell'opzione Native Backup and Restore (Backup nativo e ripristino)

Dopo aver abilitato l'opzione di backup nativo e ripristino, puoi modificare le impostazioni per l'opzione. Per ulteriori informazioni su come modificare le impostazioni dell'opzione, consulta Modifica di un'impostazione di un'opzione.

Rimozione dell'opzione Native Backup and Restore (Backup nativo e ripristino)

Puoi disattivare la funzionalità di backup nativo e ripristino rimuovendo l'opzione dall'istanza database. Dopo aver rimosso l'opzione di backup nativo e ripristino, non è necessario riavviare l'istanza database.

Per rimuovere l'opzione di backup nativo e ripristino da un'istanza database, puoi procedere in uno dei seguenti modi:

- Rimuovere l'opzione dal gruppo di opzioni a cui appartiene. Questa modifica coinvolge tutte le istanze database che usano il gruppo di opzioni. Per ulteriori informazioni, consulta Rimozione di un'opzione da un gruppo di opzioni.
- Modifica l'istanza database e specifica un altro gruppo di opzioni che non comprenda l'opzione di backup nativo e ripristino. Questa modifica coinvolge una singola istanza database. È possibile specificare il gruppo di opzioni predefinito (vuoto) o un gruppo di opzioni personalizzato diverso. Per ulteriori informazioni, consulta Modifica di un'istanza database Amazon RDS.

Supporto per Transparent Data Encryption in SQL Server

Amazon RDS supporta l'utilizzo di Transparent Data Encryption (TDE) per crittografare i dati archiviati nelle istanze database che eseguono Microsoft SQL Server. TDE consente la crittografia automatica dei dati prima che vengano trascritti nello storage e la loro decriptazione automatica durante la lettura dallo storage.

Amazon RDS supporta TDE per le seguenti versioni ed edizioni di SQL Server:

- Edizioni SQL Server 2022 Standard ed Enterprise
- SQL Server 2019 Standard ed Enterprise Edition
- SQL Server 2017 Enterprise Edition
- SQL Server 2016 Enterprise Edition
- SQL Server 2014 Enterprise Edition

Transparent Data Encryption per SQL Server offre la gestione delle chiavi di crittografia tramite un'architettura di chiavi a due livelli. Un certificato, generato dalla chiave master del database, viene utilizzato per proteggere le chiavi di crittografia dei dati. La chiave di crittografia del database esegue la crittografia e la decrittografia effettive dei dati nel database utente. Amazon RDS esegue il backup e gestisce la chiave master del database e del certificato TDE.

Transparent Data Encryption è utilizzata negli scenari in cui occorre crittografare i dati sensibili. Ad esempio, potresti dover fornire file di dati e backup a una terza parte o risolvere problemi di conformità relativi a norme di sicurezza. Non è possibile crittografare i database di sistema per SQL Server, ad esempio i database `model` o `master`.

Una descrizione dettagliata di Transparent Data Encryption non rientra nell'ambito di questa guida, ma assicurati di comprendere i vantaggi e gli svantaggi in termini di sicurezza di ciascuna chiave e ciascun algoritmo di crittografia. Per informazioni su Transparent Data Encryption per SQL Server, consulta Transparent Data Encryption (TDE) sul sito Web Microsoft.

Argomenti

- Attivazione di TDE per RDS per SQL Server
- Crittografia dei dati su RDS per SQL Server
- Backup e ripristino dei certificati TDE su RDS per SQL Server
- Backup e ripristino di certificati TDE per database on-premise

- Disattivazione di TDE per RDS per SQL Server

Attivazione di TDE per RDS per SQL Server

Per attivare Transparent Data Encryption per un'istanza database di RDS per SQL Server, specifica l'opzione TDE in gruppo di opzioni RDS associato a tale istanza database.

1. Stabilisci se l'istanza database è già associata a un gruppo di opzioni contenente l'opzione TDE. Per visualizzare il gruppo di opzioni a cui è associata un'istanza DB, utilizzare la console RDS, il describe-db-instanceAWS CLIcomando o l'operazione API DescribedBInstances.
2. Se l'istanza database non è associata a un gruppo di opzioni con TDE attivato, le opzioni sono due. Si può creare un gruppo di opzioni e aggiungere l'opzione TDE o si può modificare il gruppo di opzioni associato in modo da aggiungerlo.

   > **Note**
   >
   > Nella console RDS l'opzione è denominata `TRANSPARENT_DATA_ENCRYPTION`. Nell'AWS CLI e nell'API RDS è denominata `TDE`.

   Per informazioni sulla creazione o la modifica di un gruppo di opzioni, consulta Uso di gruppi di opzioni. Per informazioni sull'aggiunta di un'opzione a un gruppo di opzioni, consulta Aggiunta di un'opzione a un gruppo di opzioni.
3. Associa l'istanza database con il gruppo di opzioni contenente l'opzione TDE. Per informazioni su come associare un'istanza database con un gruppo di opzioni, consulta Modifica di un'istanza database Amazon RDS.

Considerazioni su gruppi di opzioni

L'opzione TDE è un'opzione persistente. Non può essere rimossa da un gruppo di opzioni, a meno che tutte le istanze database e i backup non siano più associati gruppo di opzioni. Una volta aggiunta l'opzione TDE a un gruppo di opzioni, quest'ultimo può essere associato solo a istanze database che utilizzano TDE. Per ulteriori informazioni sulle opzioni persistenti in un gruppo di opzioni, consulta Panoramica dei gruppi di opzioni.

Poiché l'opzione TDE è persistente, è possibile che si verifichi un conflitto tra un gruppo di opzioni e un'istanza database associata. Nelle seguenti situazioni si può verificare un conflitto:

- Il gruppo di opzioni corrente dispone dell'opzione TDE e può essere sostituito con un gruppo di opzioni che non dispone dell'opzione TDE.
- Si esegue il ripristino da uno snapshot DB a una nuova istanza database che non dispone di un gruppo di opzioni che contiene l'opzione TDE. Per ulteriori informazioni su questo scenario, consulta Considerazioni su gruppi di opzioni.

Considerazioni sulle prestazioni di SQL Server

L'utilizzo di Transparent Data Encryption può influenzare le prestazioni di un'istanza database di SQL Server.

Le prestazioni dei database non crittografati possono anche essere ridotte se i database si trovano su un'istanza database con almeno un database crittografato. Ti consigliamo pertanto di mantenere i database crittografati e non crittografati su istanze database separate.

Crittografia dei dati su RDS per SQL Server

Quando l'opzione TDE viene aggiunta a un gruppo di opzioni, Amazon RDS genera un certificato utilizzato nel processo di crittografia. È quindi possibile utilizzare il certificato per eseguire le istruzioni SQL che consentono di crittografare i dati in un database nell'istanza database.

L'esempio seguente utilizza il certificato creato da RDS `RDSTDECertificateName` per crittografare un database denominato `myDatabase`.

```
---------- Turning on TDE -------------

-- Find an RDS TDE certificate to use
USE [master]
GO
SELECT name FROM sys.certificates WHERE name LIKE 'RDSTDECertificate%'
GO

USE [myDatabase]
GO
-- Create a database encryption key (DEK) using one of the certificates from the
 previous step
CREATE DATABASE ENCRYPTION KEY WITH ALGORITHM = AES_256
ENCRYPTION BY SERVER CERTIFICATE [RDSTDECertificateName]
GO

-- Turn on encryption for the database
```

```
ALTER DATABASE [myDatabase] SET ENCRYPTION ON
GO

-- Verify that the database is encrypted
USE [master]
GO
SELECT name FROM sys.databases WHERE is_encrypted = 1
GO
SELECT db_name(database_id) as DatabaseName, * FROM sys.dm_database_encryption_keys
GO
```

Il tempo necessario a crittografare un database SQL Server utilizzando l'opzione TDE dipende da diversi fattori. Tali fattori includono le dimensioni dell'istanza database, se l'istanza utilizza archiviazione IOPS allocata, la quantità di dati e altri fattori.

Backup e ripristino dei certificati TDE su RDS per SQL Server

RDS per SQL Server fornisce stored procedure per il backup, il ripristino e il rilascio di certificati TDE. RDS per SQL Server fornisce inoltre una funzione per la visualizzazione dei certificati TDE utente ripristinati.

I certificati TDE utente vengono utilizzati per ripristinare i database su RDS per SQL Server on-premise e con TDE attivato. Questi certificati hanno il prefisso `UserTDECertificate_`. Dopo aver ripristinato i database e prima di renderli disponibili per l'uso, RDS modifica i database in cui TDE è attivato per utilizzare i certificati TDE generati da RDS. Questi certificati hanno il prefisso `RDSTDECertificate`.

I certificati TDE dell'utente rimangono nell'istanza database RDS per SQL Server, a meno che non vengano rilasciati utilizzando la stored procedure `rds_drop_tde_certificate`. Per ulteriori informazioni, consulta Ripristino di certificati TDE ripristinati.

Puoi utilizzare un certificato TDE utente per ripristinare altri database dall'istanza database di origine. I database da ripristinare devono utilizzare lo stesso certificato TDE e avere TDE attivato. Non è necessario importare (ripristinare) nuovamente lo stesso certificato.

Argomenti

- Prerequisiti
- Limitazioni
- Backup di un certificato TDE

- Ripristino di un certificato TDE
- Ripristino di certificati TDE ripristinati
- Ripristino di certificati TDE ripristinati

Prerequisiti

Prima poter eseguire il backup o il ripristino dei certificati TDE su RDS per SQL Server, assicurati di eseguire le seguenti attività. Le prime tre sono descritte in Configurazione di backup e ripristino nativi.

1. Crea bucket Amazon S3 per l'archiviazione di file di cui eseguire il backup e il ripristino.

   Si consiglia di utilizzare bucket separati per i backup del database e per i backup dei certificati TDE.

2. Crea un ruolo IAM per il backup e il ripristino dei file.

   Il ruolo IAM deve essere sia un utente sia un amministratore per AWS KMS key.

   Oltre alle autorizzazioni richieste per il backup e il ripristino nativi di SQL Server, il ruolo IAM richiede anche le seguenti autorizzazioni:

   - `s3:GetBucketACL`, `s3:GetBucketLocation` e `s3:ListBucket` sulla risorsa di bucket S3
   - `s3:ListAllMyBuckets` sulla risorsa *

3. Aggiungi l'opzione `SQLSERVER_BACKUP_RESTORE` a un gruppo di opzioni sull'istanza database.

   Questa è in aggiunta all'opzione `TRANSPARENT_DATA_ENCRYPTION` (`TDE`).

4. Assicurati di disporre di una chiave KMS di crittografia simmetrica. Sono disponibili le seguenti opzioni:

   - Se disponi di una chiave KMS esistente nel tuo account, puoi utilizzarla. Non è richiesta alcuna operazione aggiuntiva.
   - Se non disponi di una chiave KMS di crittografia simmetrica esistente nel tuo account, crea una chiave KMS seguendo le istruzioni in Creazione di chiavi nella Guida per gli sviluppatori di AWS Key Management Service.

5. Abilita l'integrazione con Amazon S3 per trasferire file tra l'istanza database e Amazon S3.

   Per ulteriori informazioni sull'abilitazione dell'integrazione di Amazon S3, consulta Integrazione di un'istanza database Amazon RDS for SQL Server con Amazon S3.

Limitazioni

L'utilizzo di stored procedure per eseguire il backup e il ripristino di certificati TDE presenta le seguenti limitazioni:

- Le opzioni `SQLSERVER_BACKUP_RESTORE` e `TRANSPARENT_DATA_ENCRYPTION` (`TDE`) devono essere entrambe aggiunte al gruppo di opzioni associato all'istanza database.
- Il backup e ripristino del certificato TDE non sono supportati nelle istanze database Multi-AZ.
- L'annullamento delle attività di backup e ripristino del certificato TDE non è supportata.
- Non è possibile utilizzare un certificato TDE utente per la crittografia TDE di qualsiasi altro database nell'istanza database di RDS per SQL Server. Puoi utilizzarlo per ripristinare solo altri database dall'istanza database di origine in cui TDE è attivato e che utilizzano lo stesso certificato TDE.
- Puoi rilasciare solo certificati TDE utente.
- Il numero massimo di certificati TDE utente supportati su RDS è 10. Se il numero supera 10, rilascia i certificati TDE inutilizzati e riprova.
- Il nome del certificato non può essere vuoto o nullo.
- Durante il ripristino di un certificato, il nome del certificato non può includere la parola chiave `RDSTDECERTIFICATE` e deve iniziare con il prefisso `UserTDECertificate_`.
- Il parametro `@certificate_name` può includere solo i seguenti caratteri: a-z, 0-9, @, $, # e carattere di sottolineatura (_).
- L'estensione file per `@certificate_file_s3_arn` deve essere .cer (senza distinzione tra maiuscole e minuscole).
- L'estensione file per `@private_key_file_s3_arn` deve essere .cer (senza distinzione tra maiuscole e minuscole).
- I metadati S3 per il file della chiave privata devono includere il tag `x-amz-meta-rds-tde-pwd`. Per ulteriori informazioni, consulta Backup e ripristino di certificati TDE per database on-premise.

Backup di un certificato TDE

Per eseguire il backup dei certificati TDE, utilizza la stored procedure `rds_backup_tde_certificate`. Di seguito è riportata la sintassi utilizzata.

```
EXECUTE msdb.dbo.rds_backup_tde_certificate
    @certificate_name='UserTDECertificate_certificate_name |
 RDSTDECertificatetimestamp',
```

```
    @certificate_file_s3_arn='arn:aws:s3:::bucket_name/certificate_file_name.cer',
    @private_key_file_s3_arn='arn:aws:s3:::bucket_name/key_file_name.pvk',
    @kms_password_key_arn='arn:aws:kms:region:account-id:key/key-id',
    [@overwrite_s3_files=0|1];
```

I parametri seguenti sono obbligatori:

- `@certificate_name` – Nome del certificato TDE di cui eseguire il backup.
- `@certificate_file_s3_arn` – Nome della risorsa Amazon (ARN) di destinazione per il file di backup del certificato in Amazon S3.
- `@private_key_file_s3_arn` – ARN di S3 di destinazione del file della chiave privata che protegge il certificato TDE.
- `@kms_password_key_arn` – ARN della chiave KMS simmetrica utilizzata per crittografare la password della chiave privata.

Il parametro seguente è facoltativo:

- `@overwrite_s3_files` – Indica se sovrascrivere il certificato esistente e i file di della chiave privata in S3:
  - `0` – Il file esistente non viene sovrascritto. Questo è il valore predefinito.

    L'impostazione di `@overwrite_s3_files` su 0 restituisce un errore se il file esiste già.
  - `1` – Il file esistente con il nome specificato viene sovrascritto, anche se non è un file di backup.

Example di backup di un certificato TDE

```
EXECUTE msdb.dbo.rds_backup_tde_certificate
    @certificate_name='RDSTDECertificate20211115T185333',
    @certificate_file_s3_arn='arn:aws:s3:::TDE_certs/mycertfile.cer',
    @private_key_file_s3_arn='arn:aws:s3:::TDE_certs/mykeyfile.pvk',
    @kms_password_key_arn='arn:aws:kms:us-
west-2:123456789012:key/AKIAIOSFODNN7EXAMPLE',
    @overwrite_s3_files=1;
```

Ripristino di un certificato TDE

La stored procedure `rds_restore_tde_certificate` viene utilizzata per ripristinare (importare) certificati TDE utente. Di seguito è riportata la sintassi utilizzata.

```
EXECUTE msdb.dbo.rds_restore_tde_certificate
    @certificate_name='UserTDECertificate_certificate_name',
    @certificate_file_s3_arn='arn:aws:s3:::bucket_name/certificate_file_name.cer',
    @private_key_file_s3_arn='arn:aws:s3:::bucket_name/key_file_name.pvk',
    @kms_password_key_arn='arn:aws:kms:region:account-id:key/key-id';
```

I parametri seguenti sono obbligatori:

- `@certificate_name` – Nome del certificato TDE di cui eseguire il backup. Il nome deve iniziare con il prefisso `UserTDECertificate_`.
- `@certificate_file_s3_arn` – L'ARN S3 del file di backup utilizzato per ripristinare il certificato TDE.
- `@private_key_file_s3_arn` – L'ARN S3 del file di backup utilizzato per ripristinare il certificato TDE.
- `@kms_password_key_arn` – L'ARN della chiave KMS simmetrica utilizzata per crittografare la password della chiave privata.

Example di ripristino di un certificato TDE

```
EXECUTE msdb.dbo.rds_restore_tde_certificate
    @certificate_name='UserTDECertificate_myTDEcertificate',
    @certificate_file_s3_arn='arn:aws:s3:::TDE_certs/mycertfile.cer',
    @private_key_file_s3_arn='arn:aws:s3:::TDE_certs/mykeyfile.pvk',
    @kms_password_key_arn='arn:aws:kms:us-
west-2:123456789012:key/AKIAIOSFODNN7EXAMPLE';
```

Ripristino di certificati TDE ripristinati

La funzione `rds_fn_list_user_tde_certificates` viene utilizzata per ripristinare (importare) certificati TDE utente. Di seguito è riportata la sintassi utilizzata.

```
SELECT * FROM msdb.dbo.rds_fn_list_user_tde_certificates();
```

L'output è simile a quello riportato di seguito. Non tutte le colonne sono mostrate qui.

```
name   certif princi pvt_ke issuer cert_s thumbp subjec start_ expiry pvt_key_l
       te_id  _id    ncrypt me     al_num t      t       e      te     ast_backu
                                                                      p_date
```

			_type_c							
UserTDCertific_tde_c	343	1	ENCRYPT_BY_MAR_KEY	AnyCompany Shippi	793e57a369fd1d9e472c32671d9caaf	0x6BB2341103 80BFE1BA2C695095B5	AnyCompany Shippi	2022-05 19:49:000000	2023-05 19:49:000000	NULL

Ripristino di certificati TDE ripristinati

Per eliminare i certificati TDE utente ripristinati (importati) che non si utilizzano, utilizzare il`rds_drop_tde_certificate`procedura archiviata. Di seguito è riportata la sintassi utilizzata.

```
EXECUTE msdb.dbo.rds_drop_tde_certificate
 @certificate_name='UserTDECertificate_certificate_name';
```

Il parametro seguente è obbligatorio:

- `@certificate_name` – Nome del certificato TDE da rilasciare.

Puoi rilasciare solo i certificati TDE ripristinati (importati). Non puoi rilasciare i certificati creati da RDS.

Example di rilascio di un certificato TDE

```
EXECUTE msdb.dbo.rds_drop_tde_certificate
 @certificate_name='UserTDECertificate_myTDEcertificate';
```

Backup e ripristino di certificati TDE per database on-premise

Puoi eseguire il backup di certificati TDE per i database on-premise, quindi ripristinarli in seguito su RDS per SQL Server. Inoltre, puoi ripristinare un certificato TDE RDS per SQL Server in un'istanza database on-premise.

La procedura seguente esegue il backup di un certificato TDE e una chiave privata. La chiave privata viene crittografata utilizzando una chiave dei dati generata dalla chiave KMS di crittografia simmetrica.

Eseguire il backup di un certificato TDE on-premise

1. Genera la chiave dati utilizzando il comando. AWS CLI [generate-data-key]

   ```
   aws kms generate-data-key \
       --key-id my_KMS_key_ID \
       --key-spec AES_256
   ```

   L'output è simile a quello riportato di seguito.

   ```
   {
   "CiphertextBlob": "AQIDAHimL2NEoAlOY6Bn7LJfnxi/OZe9kTQo/
   XQXduug1rmerwGiL7g5ux4av9GfZLxYTDATAAAAfjB8BgkqhkiG9w0B
   BwagbzBtAgEAMGgGCSqGSIb3DQEHATAeBglghkgBZQMEAS4wEQQMyCxLMi7GRZgKqD65AgEQgDtjvZLJo2cQ31Vetng
   2RezQy3sAS6ZHrCjfnfn0c65bFdhsXxjSMnudIY7AKw==",
   "Plaintext": "U/fpGtmzGCYBi8A2+0/9qcRQRK2zmG/aOn939ZnKi/0=",
   "KeyId": "arn:aws:kms:us-west-2:123456789012:key/1234abcd-00ee-99ff-88dd-
   aa11bb22cc33"
   }
   ```

   L'output di testo normale nel passaggio successivo viene utilizzato come password della chiave privata.

2. Esegui il backup del certificato TDE come mostrato nell'esempio seguente.

   ```
   BACKUP CERTIFICATE myOnPremTDEcertificate TO FILE = 'D:\tde-cert-backup.cer'
   ```

```
WITH PRIVATE KEY (
FILE = 'C:\Program Files\Microsoft SQL Server\MSSQL14.MSSQLSERVER\MSSQL\DATA\cert-backup-key.pvk',
ENCRYPTION BY PASSWORD = 'U/fpGtmzGCYBi8A2+0/9qcRQRK2zmG/aOn939ZnKi/0=');
```

3. Salva il file di backup del certificato nel bucket di certificato Amazon S3.
4. Salva il file di backup della chiave privata nel bucket di certificato S3, con il seguente tag nei metadati del file:

   - Chiave - `x-amz-meta-rds-tde-pwd`
   - Value – Il valore `CiphertextBlob` risultante dalla generazione della chiave dei dati, come nell'esempio seguente.

```
AQIDAHimL2NEoAlOY6Bn7LJfnxi/OZe9kTQo/
XQXduug1rmerwGiL7g5ux4av9GfZLxYTDATAAAAfjB8BgkqhkiG9w0B
BwagbzBtAgEAMGgGCSqGSIb3DQEHATAeBglghkgBZQMEAS4wEQQMyCxLMi7GRZgKqD65AgEQgDtjvZLJo2cQ31Vet
2RezQy3sAS6ZHrCjfnfn0c65bFdhsXxjSMnudIY7AKw==
```

La procedura seguente consente di ripristinare un certificato TDE RDS per SQL Server in un'istanza database on-premise. Copia e ripristina il certificato TDE sull'istanza database di destinazione utilizzando il backup del certificato, il file della chiave privata corrispondente e la chiave dati. Il certificato ripristinato viene crittografato dalla chiave master del database del nuovo server.

Ripristinare un certificato TDE

1. Copia il file di backup del certificato TDE e il file della chiave privata da Amazon S3 nell'istanza di destinazione. Per ulteriori informazioni sulla copia di file da Amazon S3, consulta Trasferimento di file tra RDS for SQL Server e Amazon S3.
2. Utilizza la chiave KMS per decrittografare il testo crittografato di output per recuperare il testo normale della chiave dei dati. Il testo crittografato si trova nei metadati S3 del file di backup della chiave privata.

```
aws kms decrypt \
    --key-id my_KMS_key_ID \
    --ciphertext-blob fileb://exampleCiphertextFile | base64 -d \
    --output text \
    --query Plaintext
```

L'output di testo normale nel passaggio successivo viene utilizzato come password della chiave privata.

3. Utilizza il comando SQL seguente per ripristinare il certificato TDE.

```
CREATE CERTIFICATE myOnPremTDEcertificate FROM FILE='D:\tde-cert-backup.cer'
WITH PRIVATE KEY (FILE = N'D:\tde-cert-key.pvk',
DECRYPTION BY PASSWORD = 'plain_text_output');
```

Per ulteriori informazioni sulla decrittografia KMS, consulta decrittografare nella sezione KMS del Riferimento ai comandi AWS CLI.

Dopo che il certificato TDE è stato ripristinato sull'istanza database di destinazione, puoi ripristinare i database crittografati con tale certificato.

> **Note**
>
> Puoi utilizzare lo stesso certificato TDE per crittografare più database SQL Server sull'istanza database di origine. Per migrare più database a un'istanza di destinazione, copia il certificato TDE ad essi associato nell'istanza di destinazione una sola volta.

Disattivazione di TDE per RDS per SQL Server

Per disattivare TDE per un'istanza database di RDS per SQL Server, assicurati che non siano presenti oggetti crittografati nell'istanza database. A questo scopo, esegui la decrittografia degli oggetti o rilasciali. Se nell'istanza database sono presenti oggetti crittografati, non è possibile disattivare TDE per l'istanza database. Quando utilizzi la console per rimuovere l'opzione TDE da un gruppo di opzioni, la console indica che il processo è in corso. Inoltre, viene creato un evento di errore se il gruppo di opzioni è associato a un'istanza database o a una snapshot DB crittografati.

L'esempio seguente rimuove la crittografia TDE da un database denominato `customerDatabase`.

```
------------- Removing TDE ----------------

USE [customerDatabase]
GO

-- Turn off encryption of the database
```

```
ALTER DATABASE [customerDatabase]
SET ENCRYPTION OFF
GO

-- Wait until the encryption state of the database becomes 1. The state is 5
 (Decryption in progress) for a while
SELECT db_name(database_id) as DatabaseName, * FROM sys.dm_database_encryption_keys
GO

-- Drop the DEK used for encryption
DROP DATABASE ENCRYPTION KEY
GO

-- Alter to SIMPLE Recovery mode so that your encrypted log gets truncated
USE [master]
GO
ALTER DATABASE [customerDatabase] SET RECOVERY SIMPLE
GO
```

Quando tutti gli oggetti vengono decrittografati, sono disponibili due opzioni.

1. Puoi modificare l'istanza database da associare a un gruppo di opzioni senza l'opzione TDE.
2. Puoi rimuovere l'opzione TDE dal gruppo di opzioni.

Audit in SQL Server

In Amazon RDS, si può eseguire l'audit dei database Microsoft SQL Server utilizzando il meccanismo di audit integrato in SQL Server. Si possono creare audit e specifiche di audit nello stesso modo in cui vengono creati per i server di database locali.

RDS carica i log completi dell'audit nel bucket S3 utilizzando il ruolo IAM che viene fornito. Se si abilita la retention, RDS conserva i log di audit nell'istanza database per il periodo di tempo configurato.

Per ulteriori informazioni, consulta Audit in SQL Server (Motore di database) nella documentazione di Microsoft SQL Server.

Audit di SQL Server con flussi di attività del database

È possibile utilizzare Database Activity Streams for RDS per integrare gli eventi di SQL Server Audit con gli strumenti di monitoraggio delle attività del database di Imperva e McAfee IBM. Per ulteriori informazioni sull'audit con i flussi di attività del database per RDS in SQL Server, consulta Verifica in Microsoft SQL Server

Argomenti

- Supporto per l'audit in SQL Server
- Aggiunta dell'audit in SQL Server alle opzioni dell'istanza database
- Utilizzo dell'audit in SQL Server
- Visualizzazione dei log di audit
- Utilizzo dell'audit in SQL Server con le istanze Multi-AZ
- Configurazione di un bucket S3
- Creazione manuale di un ruolo IAM per l'audit in SQL Server

Supporto per l'audit in SQL Server

In Amazon RDS, a partire da SQL Server 2014, tutte le edizioni di SQL Server supportano gli audit a livello di server; inoltre, l'edizione Enterprise supporta anche gli audit a livello di database. A partire da SQL Server 2016 (13.x) SP1, tutte le edizioni supportano gli audit sia a livello di server che di database. Per ulteriori informazioni, consulta Audit in SQL Server (Motore di database) nella documentazione di SQL Server.

RDS supporta la configurazione delle seguenti impostazioni delle opzioni per l'audit in SQL Server.

Impostazione opzioni	Valori validi	Descrizione
`IAM_ROLE_ARN`	Amazon Resource Name (ARN) valido nel formato `arn:aws:iam::` *`account-id`* `:role/`*`role-name`* .	L'ARN del ruolo IAM che concede l'accesso al bucket S3 dove si desidera archiviare i log di audit. Per ulteriori informazioni, consulta Amazon Resource Names (ARNs) nella Riferimenti generali di AWS.
`S3_BUCKET_ARN`	Un ARN valido nel formato `arn:aws:s3:::bucket-name` o `arn:aws:s3:::bucket-name/key-prefix`	L'ARN del bucket S3 dove si desidera archiviare i log di audit.
`ENABLE_COMPRESSION`	`true` o `false`	Controlla la compressione dei log di audit. Per impostazione predefinita, la compressione è abilitata (impostata su `true`).
`RETENTION_TIME`	`0` Da a `840`	Il periodo di retention (in ore) in cui i log di audit in SQL Server vengono conservati nell'istanza RDS. Per impostazione predefinita, la retention è disabilitata.

RDS supporta SQL Server Audit in tutte le AWS regioni tranne il Medio Oriente (Bahrain).

Aggiunta dell'audit in SQL Server alle opzioni dell'istanza database

L'abilitazione dell'audit in SQL Server richiede due fasi: l'abilitazione dell'opzione nell'istanza database e l'abilitazione della funzione in SQL Server. La procedura per aggiungere l'opzione dell'audit in SQL Server a un'istanza database è la seguente:

1. Creare un nuovo gruppo di opzioni oppure copiare o modificare un gruppo di opzioni esistente.
2. Aggiungere e configurare tutte le opzioni necessarie.
3. Associare il gruppo di opzioni a questa istanza database.

Dopo aver aggiunto l'opzione dell'audit in SQL Server, non è necessario riavviare l'istanza database. Non appena il gruppo di opzioni diventa attivo, è possibile creare audit e archiviarne i log nel bucket S3.

Per aggiungere e configurare l'audit in SQL Server su un gruppo di opzioni di un'istanza database

1. Scegliere una delle seguenti opzioni:
   - Utilizzare un gruppo di opzioni esistente.
   - Creare un gruppo di opzioni database personalizzato e utilizzarlo. Per ulteriori informazioni, consulta Creazione di un gruppo di opzioni.
2. Aggiungere l'opzione SQLSERVER_AUDIT al gruppo di opzioni e configurare le impostazioni dell'opzione. Per ulteriori informazioni sull'aggiunta di opzioni, consulta Aggiunta di un'opzione a un gruppo di opzioni.
   - Per ruolo IAM, se già si dispone di un ruolo IAM con le policy richieste, è possibile sceglierlo. Per creare un nuovo ruolo IAM, selezionare Create a New Role (Crea un nuovo ruolo). Per informazioni sulle policy richieste, consulta Creazione manuale di un ruolo IAM per l'audit in SQL Server.
   - Per Seleziona destinazione S3, se già si dispone di un bucket S3 che si desidera utilizzare, selezionarlo. Per creare un nuovo bucket, selezionare Crea un nuovo bucket S3.
   - Per Abilita compressione, lasciare selezionata quest'opzione per comprimere i file dell'audit. Per impostazione predefinita, la compressione è abilitata. Per disabilitare la compressione, deselezionare Enable Compression (Abilita compressione).
   - Per Retention log di audit, per conservare i log di audit nell'istanza database, selezionare questa opzione. Specificare un periodo di retention in ore. Il periodo di retention massimo è di 35 giorni.
3. Applicare il gruppo di opzioni a un'istanza database nuova o esistente. Scegliere una delle seguenti opzioni:
   - Se si sta creando una nuova istanza database, applicare il gruppo di opzioni quando viene avviata l'istanza.

- In un'istanza database esistente, applicare il gruppo di opzioni modificando l'istanza e poi collegando il nuovo gruppo di opzioni. Per ulteriori informazioni, consulta Modifica di un'istanza database Amazon RDS.

Modifica dell'opzione Audit in SQL Server

Dopo aver abilitato l'opzione Audit in SQL Server, si possono modificare le impostazioni. Per informazioni su come modificare le impostazioni dell'opzione, consulta Modifica di un'impostazione di un'opzione.

Rimozione dell'audit in SQL Server dalle opzioni dell'istanza database

È possibile disattivare la funzione Audit in SQL Server disabilitando gli audit ed eliminando l'opzione.

Per rimuovere gli audit

1. Disabilitare tutte le impostazioni di audit in SQL Server. Per scoprire le posizioni in cui sono in esecuzione gli audit, eseguire una query sulle visualizzazioni del catalogo di sicurezza di SQL Server. Per ulteriori informazioni, consulta la sezione Visualizzazioni del catalogo di sicurezza nella documentazione di Microsoft SQL Server.
2. Rimuovere l'opzione dell'audit in SQL Server dall'istanza database. Scegliere una delle seguenti opzioni:

   - Eliminare l'opzione dell'audit in SQL Server dal gruppo di opzioni utilizzato dall'istanza database. Questa modifica coinvolge tutte le istanze database che utilizzano lo stesso gruppo di opzioni. Per ulteriori informazioni, consulta Rimozione di un'opzione da un gruppo di opzioni.
   - Modificare l'istanza database, quindi selezionare un gruppo di opzioni che non contenga l'opzione dell'audit in SQL Server. Questa modifica influisce solo sull'istanza database che viene modificata. È possibile specificare il gruppo di opzioni predefinito (vuoto) o un gruppo di opzioni personalizzato diverso. Per ulteriori informazioni, consulta Modifica di un'istanza database Amazon RDS.
3. Una volta eliminata l'opzione dell'audit in SQL Server dall'istanza database, non è necessario riavviare l'istanza. Rimozione di file di audit superflui dal bucket S3.

Utilizzo dell'audit in SQL Server

È possibile controllare gli audit del server, le specifiche degli audit del server e le specifiche degli audit del database allo stesso modo in cui si controllano per i server di database locali.

Creazione degli audit

Si possono creare audit di server nello stesso modo in cui vengono creati per i server di database locali. Per informazioni su come creare audit di server, consulta la sezione CREAZIONE DI AUDIT DI SERVER nella documentazione di Microsoft SQL Server.

Per evitare errori, rispettare le seguenti restrizioni:

- Non superare il numero massimo di 50 audit di server supportati per ogni istanza.
- Istruire SQL Server a scrivere dati in un file binario.
- Non utilizzare `RDS_` come prefisso del nome dell'audit di server.
- Per `FILEPATH`, specificare `D:\rdsdbdata\SQLAudit`.
- Per `MAXSIZE`, specificare una dimensione compresa tra 2 MB e 50 MB.
- Non configurare `MAX_ROLLOVER_FILES` o `MAX_FILES`.
- Non configurare SQL Server per l'interruzione dell'istanza database se non riesce a scrivere il registro dell'audit.

Creazione delle specifiche dell'audit

È possibile creare le specifiche degli audit del server e le specifiche degli audit del database allo stesso modo in cui vengono create per i server di database locali. Per informazioni sulla creazione delle specifiche dell'audit, consulta le sezioni CREAZIONE DELLE SPECIFICHE DELL'AUDIT DEL SERVER e CREAZIONE DELLE SPECIFICHE DELL'AUDIT DEL DATABASE nella documentazione di Microsoft SQL Server.

Per evitare errori, non utilizzare `RDS_` come prefisso del nome della specifica dell'audit del database o del server.

Visualizzazione dei log di audit

I log di audit vengono archiviati in `D:\rdsdbdata\SQLAudit`.

Quando SQL Server finisce di scrivere un file di log di audit—quando il file raggiunge la dimensione massima—Amazon RDS carica il file nel bucket S3. Se la retention è abilitata, Amazon RDS sposta il file nella cartella di retentio: `D:\rdsdbdata\SQLAudit\transmitted`.

Per informazioni sulla configurazione della retention, consulta Aggiunta dell'audit in SQL Server alle opzioni dell'istanza database.

I registri dell'audit vengono conservati nell'istanza database fino a quando non viene caricato il file di log dell'audit. È possibile visualizzare i registri dell'audit eseguendo il seguente comando.

```
SELECT   *
 FROM     msdb.dbo.rds_fn_get_audit_file
             ('D:\rdsdbdata\SQLAudit\*.sqlaudit'
             , default
             , default )
```

Lo stesso comando si può utilizzare per visualizzare i registri dell'audit nella cartella di retention modificando il filtro in `D:\rdsdbdata\SQLAudit\transmitted\*.sqlaudit`.

```
SELECT   *
 FROM     msdb.dbo.rds_fn_get_audit_file
             ('D:\rdsdbdata\SQLAudit\transmitted\*.sqlaudit'
             , default
             , default )
```

Utilizzo dell'audit in SQL Server con le istanze Multi-AZ

Per le istanze Multi-AZ, il processo per inviare i file di log dell'audit a Amazon S3 è simile a quello per le istanze Single-AZ. Tuttavia, vi sono alcune differenze importanti:

- Gli oggetti delle specifiche dell'audit del database vengono replicati su tutti i nodi.
- Gli audit del server e le relative specifiche non vengono replicate sui nodi secondari. Occorre crearli o modificarli manualmente.

Per acquisire gli audit del server o una specifica dell'audit del server da entrambi i nodi:

1. Creare un audit del server o una specifica dell'audit del server nel nodo primario:
2. Eseguire il failover sul nodo secondario e creare un audit del server o una specifica dell'audit del server con lo stesso nome e GUID nel nodo secondario. Utilizzare il parametro `AUDIT_GUID` per specificare il GUID.

Configurazione di un bucket S3

I file di log di audit vengono caricati automaticamente dall'istanza database al bucket S3. Per il bucket S3 utilizzato come destinazione dei file dell'audit valgono le seguenti restrizioni:

- Deve trovarsi nella stessa AWS regione dell'istanza DB.
- Non deve essere aperto al pubblico.
- Il proprietario del bucket deve essere anche il proprietario del ruolo IAM.

La chiave di destinazione utilizzata per archiviare i dati segue questo schema di denominazion: `bucket-name/key-prefix/instance-name/audit-name/node_file-name.ext`

> **Note**
>
> I valori del nome del bucket e del prefisso della chiave vanno configurati entrambi con l'impostazione dell'opzione (`S3_BUCKET_ARN`).

Lo schema è costituito dai seguenti elementi:

- **`bucket-name`** – Il nome del bucket S3.
- **`key-prefix`** – Il prefisso della chiave personalizzata da utilizzare per i log di audit.
- **`instance-name`** – Il nome dell'istanza Amazon RDS.
- **`audit-name`** – Il nome dell'audit.
- **`node`** – L'identificatore del nodo che funge da origine dell'audit (`node1` o `node2`). Esiste un nodo per un'istanza Single-AZ e due nodi di replica per un'istanza Multi-AZ. Non si tratta di nodi primari e secondari, poiché i ruoli del nodo primario e di quello secondario cambiano nel tempo. L'identificatore del nodo è piuttosto una semplice etichetta.
  - **`node1`** – Il primo nodo di replica (un'istanza Single-AZ ha un solo nodo).
  - **`node2`** – Il secondo nodo di replica (un'istanza Multi-AZ ha due nodi).
- **`file-name`** – Il nome del file di destinazione. Il nome del file è preso così com'è da SQL Server.
- **`ext`** – L'estensione del file (`zip` o `sqlaudit`):
  - **`zip`** – Se la compressione è abilitata (impostazione predefinita).
  - **`sqlaudit`** – Se la compressione è disabilitata.

Creazione manuale di un ruolo IAM per l'audit in SQL Server

In genere, quando crei una nuova opzione, AWS Management Console crea automaticamente il ruolo IAM e la policy di fiducia IAM. Si può tuttavia creare manualmente un nuovo ruolo IAM da utilizzare con gli audit in SQL Server, in modo che l'utente possa personalizzarlo con tutti gli ulteriori requisiti

che potrebbero servire. Per far ciò, l'utente crea un ruolo IAM e delega le autorizzazioni in modo che il servizio Amazon RDS possa utilizzare il bucket Amazon S3. Quando si crea un ruolo IAM, vengono collegate le policy di attendibilità e autorizzazione. La policy di attendibilità consente a Amazon RDS di assumere questo ruolo. La policy di autorizzazione definisce le operazioni che questo ruolo può eseguire. Per ulteriori informazioni, vedere Creazione di un ruolo per delegare le autorizzazioni a un AWS servizio nella Guida per l'utente di AWS Identity and Access Management.

Si possono utilizzare gli esempi riportati in questa sezione per creare le relazioni di attendibilità e le policy di autorizzazione necessarie.

Il seguente esempio mostra una relazione di attendibilità per la verifica in SQL Server. Utilizza l'entità servizio `rds.amazonaws.com` per autorizzare RDS a scrivere nel bucket S3. Un'entità servizio è un identificatore che viene utilizzato per concedere autorizzazioni a un servizio. Ogni volta che si autorizza l'accesso a `rds.amazonaws.com`, si consente a RDS di eseguire un'operazione per conto dell'utente. Per ulteriori informazioni sulle entità principali del servizio, consulta Elementi della policy JSON AWS : Entità principale.

Example relazione di attendibilità per la verifica in SQL Server

```
{
     "Version": "2012-10-17",
     "Statement": [
         {
             "Effect": "Allow",
             "Principal": {
                 "Service": "rds.amazonaws.com"
             },
             "Action": "sts:AssumeRole"
         }
     ]
 }
```

Si consiglia di utilizzare le chiavi di contesto delle condizioni globali `aws:SourceArn` e `aws:SourceAccount` nelle relazioni di trust basate sulle risorse per limitare le autorizzazioni del servizio relative a una risorsa specifica. Questo è il modo più efficace per proteggersi dal problema di deputy confused.

Puoi usare le chiavi di contesto delle condizioni globali e avere il valore `aws:SourceArn` che contiene l'ID dell'account. In questo caso, il valore `aws:SourceAccount` e l'account nel valore `aws:SourceArn` deve utilizzare lo stesso ID account quando viene utilizzato nella stessa istruzione.

- Utilizzare `aws:SourceArn` se si desidera un accesso cross-service per una singola risorsa.
- Utilizzare `aws:SourceAccount` se si desidera consentire l'associazione di qualsiasi risorsa in tale account all'uso cross-service.

Nella relazione di trust, assicurati di utilizzare la chiave di contesto della condizione globale `aws:SourceArn` con l'Amazon Resource Name (ARN) completo delle risorse che accedono al ruolo. Perla verifica in SQL Server assicurati di includere sia il gruppo di opzioni database che le istanze database, come illustrato nell'esempio seguente.

Example relazione di affidabilità con la chiave di contesto delle condizioni globali per la verifica in SQL Server

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "rds.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceArn": [
                        "arn:aws:rds:Region:my_account_ID:db:db_instance_identifier",
                        "arn:aws:rds:Region:my_account_ID:og:option_group_name"
                    ]
                }
            }
        }
    ]
}
```

Nel seguente esempio di policy di autorizzazione per la verifica in SQL Server, specifichiamo un ARN per il bucket Amazon S3. Si possono utilizzare gli ARN per identificare account, utenti o ruoli specifici a cui si desidera concedere l'accesso. Per ulteriori informazioni sull'utilizzo degli ARN, consulta la pagina Amazon Resource Names (ARN).

Example policy di autorizzazione per la verifica in SQL Server

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "s3:ListAllMyBuckets",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:ListBucket",
                "s3:GetBucketACL",
                "s3:GetBucketLocation"
            ],
            "Resource": "arn:aws:s3:::bucket_name"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",
                "s3:ListMultipartUploadParts",
                "s3:AbortMultipartUpload"
            ],
            "Resource": "arn:aws:s3:::bucket_name/key_prefix/*"
        }
    ]
}
```

> **Note**
>
> L'`s3:ListAllMyBuckets`azione è necessaria per verificare che lo stesso AWS account possieda sia il bucket S3 che l'istanza DB di SQL Server. L'operazione elenca i nomi dei bucket dell'account.
>
> Gli spazi dei nomi dei bucket S3 sono globali. Se elimini accidentalmente il bucket, un altro utente può creare un bucket con lo stesso nome in un account diverso. Quindi i dati di audit di SQL Server vengono scritti nel nuovo bucket.

Supporto for SQL Server Analysis Services in Amazon RDS for SQL Server

Microsoft SQL Server Analysis Services (SSAS) fa parte della suite Microsoft Business Intelligence (MSBI). SSAS è uno strumento di elaborazione analitica online (OLAP) e data mining installato all'interno di SQL Server. SSAS viene utilizzato per analizzare i dati per prendere decisioni aziendali. SSAS differisce dal database relazionale di SQL Server perché SSAS è ottimizzato per query e calcoli comuni in un ambiente di business intelligence.

È possibile abilitare SSAS per le istanze database esistenti o nuove. È installato sulla stessa istanza database del motore del database. Per ulteriori informazioni su SSAS, consulta la documentazione di Microsoft Analysis Services.

Amazon RDS supporta le edizioni Standard ed Enterprise di SSAS per SQL nelle seguenti versioni:

- Modalità tabulare:
  - SQL Server 2019, versione 15.00.4043.16.v1 e successive
  - SQL Server 2017, versione 14.00.3223.3.v1 e successive
  - SQL Server 2016, versione 13.00.5426.0.v1 e successive
- Modalità multidimensionale:
  - SQL Server 2019, versione 15.00.4153.1.v1 e successive
  - SQL Server 2017, versione 14.00.3381.3.v1 e successive
  - SQL Server 2016, versione 13.00.5882.1.v1 e successive

Indice

  - Configurazione di un utente autenticato da Windows per SSAS
  - Aggiunta di un utente di dominio come amministratore di database
  - Creazione di un proxy SSAS
  - Pianificazione dell'elaborazione del database SSAS utilizzando SQL Server Agent
  - Revoca dell'accesso SSAS dal proxy
- Backup di un database SSAS
- Ripristino di un database SSAS
  - Ripristino a un'ora specifica per un'istanza database
- Modifica della modalità SSAS
- Disattivazione di SSAS
- Risoluzione dei problemi SSAS

Limitazioni

Le seguenti limitazioni si applicano all'uso di SSAS su RDS per SQL Server:

- RDS for SQL Server supporta l'esecuzione di SSAS in modalità tabulare o multidimensionale. Per ulteriori informazioni, consulta Comparing tabular and multidimensional solutions (Confronto tra soluzioni tabulari e multidimensionali) nella documentazione di Microsoft.
- Puoi utilizzare una sola modalità SSAS alla volta. Prima di cambiare modalità, assicurati di eliminare tutti i database SSAS.

  Per ulteriori informazioni, consulta Modifica della modalità SSAS.
- Le istanze multi-AZ non sono supportate.
- Le istanze devono utilizzare Active Directory autogestito o per l'autenticazione SSAS. AWS Directory Service for Microsoft Active Directory Per ulteriori informazioni, consulta Utilizzo di Active Directory con RDS per SQL Server.
- Agli utenti non viene concesso l'accesso all'amministratore del server SSAS, ma è possibile concedere loro l'accesso amministratore a livello di database.
- L'unica porta supportata per l'accesso a SSAS è 2383.
- Non è possibile distribuire direttamente i progetti. A tale scopo forniamo una stored procedure RDS. Per ulteriori informazioni, consulta Distribuzione di progetti SSAS su Amazon RDS.
- L'elaborazione durante la distribuzione non è supportata.

- L'utilizzo di file xmla per la distribuzione non è supportato.
- I file di input del progetto SSAS e i file di output di backup del database possono essere solo `D:\S3` nella cartella dell'istanza database.

Attivazione di SSAS

Utilizza il seguente processo per attivare SSAS per l'istanza database:

1. Creare un nuovo gruppo di opzioni oppure utilizzare un gruppo di opzioni esistente.
2. Aggiungere l'opzione `SSAS` al gruppo di opzioni.
3. Associare il gruppo di opzioni a questa istanza database.
4. Consentire l'accesso in ingresso al gruppo di sicurezza Virtual Private Cloud (VPC) per la porta listener SSAS.
5. Attiva l'integrazione Amazon S3.

Creazione di un gruppo di opzioni per SSAS

Utilizza AWS Management Console o AWS CLI per creare un gruppo di opzioni che corrisponda al motore di SQL Server e alla versione dell'istanza DB che intendi utilizzare.

> **Note**
>
> È inoltre possibile utilizzare un gruppo di opzioni esistente se si tratta del motore e della versione di SQL Server corretti.

Console

La seguente procedura della console crea un gruppo di opzioni per SQL Server Standard Edition 2017.

Per creare il gruppo di opzioni

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel pannello di navigazione scegliere Option groups (Gruppi di opzioni).
3. Seleziona Create group (Crea gruppo).

4. Nella finestra Create option group (Crea gruppo di opzioni) eseguire queste operazioni:

   a. Per Nome, inserisci un nome per il gruppo di opzioni che sia unico all'interno del tuo AWS account, ad esempio**ssas-se-2017**. Il nome può includere solo lettere, cifre e trattini.

   b. Per Descrizione, immettere una breve descrizione del gruppo di opzioni, ad esempio **SSAS option group for SQL Server SE 2017**. La descrizione viene usata per la visualizzazione.

   c. Per Engine (Motore), scegliere sqlserver-se.

   d. Per Major engine version (Versione principale del motore), scegli 14.00.

5. Scegliere Create (Crea).

CLI

Nell'esempio seguente di CLI viene creato un gruppo di opzioni per SQL Server Standard Edition 2017.

Per creare il gruppo di opzioni

- Utilizzare uno dei seguenti comandi.

  Example

  Per LinuxmacOS, oUnix:

```
aws rds create-option-group \
    --option-group-name ssas-se-2017 \
    --engine-name sqlserver-se \
    --major-engine-version 14.00 \
    --option-group-description "SSAS option group for SQL Server SE 2017"
```

  Per Windows:

```
aws rds create-option-group ^
    --option-group-name ssas-se-2017 ^
    --engine-name sqlserver-se ^
    --major-engine-version 14.00 ^
    --option-group-description "SSAS option group for SQL Server SE 2017"
```

Aggiunta dell'opzione SSAS al gruppo di opzioni

Quindi, usa AWS Management Console o AWS CLI per aggiungere l'`SSAS`opzione al gruppo di opzioni.

Console

Per aggiungere l'opzione SSAS

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel pannello di navigazione scegliere Option groups (Gruppi di opzioni).
3. Scegliere il gruppo di opzioni appena creato.
4. Scegliere Add option (Aggiungi opzione).
5. In Option details (Dettagli opzione), scegliere SSAS per Option name (Nome opzione).
6. In Impostazioni delle opzioni, effettuare le seguenti operazioni:

   a. Per Max memory (Memoria massima), inserisci un valore nell'intervallo da 10 a 80.

      Max memory (Memoria massima) specifica la soglia superiore al di sopra della quale SSAS inizia a rilasciare la memoria in modo più aggressivo per fare spazio alle richieste in esecuzione e anche alle nuove richieste ad alta priorità. Il numero è una percentuale della memoria totale dell'istanza database. I valori validi sono compresi tra 10–80 e 45 è il valore predefinito.

   b. Per Mode (Modalità), scegli la modalità server SSAS, Tabular (Tabulare) o Multidimensional (Multidimensionale).

      Se non vedi l'impostazione dell'opzione Modalità, significa che la modalità multidimensionale non è supportata nella tua regione. AWS Per ulteriori informazioni, consulta Limitazioni.

      Tabular (Tabulare) è il valore di default.

   c. Per Security groups (Gruppi di sicurezza), scegliere il gruppo di sicurezza VPC da associare all'opzione.

> **Note**
>
> La porta per l'accesso a SSAS, 2383, è prepopolata.

7. In Scheduling (Pianificazione), scegliere se aggiungere l'opzione immediatamente o alla finestra di manutenzione successiva.
8. Scegliere Add option (Aggiungi opzione).

CLI

Per aggiungere l'opzione SSAS

1. Creare un file JSON, ad esempio `ssas-option.json`, con i seguenti parametri:

   - `OptionGroupName` – Il gruppo di opzioni creato o scelto in precedenza (`ssas-se-2017` nell'esempio seguente).
   - `Port` – La porta utilizzata per accedere a SSAS. L'unica porta supportata è 2383.
   - `VpcSecurityGroupMemberships` – Appartenenze ai gruppi di sicurezza VPC per l'istanza database RDS.
   - `MAX_MEMORY` – Specifica la soglia superiore al di sopra della quale SSAS deve iniziare a rilasciare la memoria in modo più aggressivo per fare spazio alle richieste in esecuzione e anche alle nuove richieste ad alta priorità. Il numero è una percentuale della memoria totale dell'istanza database. I valori validi sono compresi tra 10–80 e 45 è il valore predefinito.
   - `MODE` – La modalità server SSAS, `Tabular` o `Multidimensional`. `Tabular` è il valore di default.

     Se ricevi un errore che indica che l'impostazione dell'`MODE`opzione non è valida, significa che la modalità multidimensionale non è supportata nella tua regione. AWS Per ulteriori informazioni, consulta Limitazioni.

   Di seguito è riportato un esempio di file JSON con impostazioni delle opzioni SSAS.

```
{
"OptionGroupName": "ssas-se-2017",
"OptionsToInclude": [
 {
 "OptionName": "SSAS",
 "Port": 2383,
 "VpcSecurityGroupMemberships": ["sg-0abcdef123"],
 "OptionSettings": [{"Name":"MAX_MEMORY","Value":"60"},
{"Name":"MODE","Value":"Multidimensional"}]
 }],
```

```
"ApplyImmediately": true
}
```

2. Aggiungere l'opzione SSAS al gruppo di opzioni.

   Example

   PerLinux, o: macOS Unix

```
aws rds add-option-to-option-group \
    --cli-input-json file://ssas-option.json \
    --apply-immediately
```

   Per Windows:

```
aws rds add-option-to-option-group ^
    --cli-input-json file://ssas-option.json ^
    --apply-immediately
```

Associazione del gruppo di opzioni all'istanza database

È possibile utilizzare la console o la CLI per associare il gruppo di opzioni all'istanza database.

Console

Associare il gruppo di opzioni a un'istanza database nuova o esistente:

- Per una nuova istanza database, associare il gruppo di opzioni all'istanza database quando si avvia l'istanza. Per ulteriori informazioni, consulta Creazione di un'istanza database Amazon RDS.
- Per un'istanza database esistente, modificare l'istanza e associare il nuovo gruppo di opzioni con essa. Per ulteriori informazioni, consulta Modifica di un'istanza database Amazon RDS.

> **Note**
>
> Se si utilizza un'istanza esistente, è necessario che siano già associati un dominio Active Directory e un ruolo AWS Identity and Access Management (IAM) associato. Se si crea una nuova istanza, specificare un dominio Active Directory esistente e un ruolo IAM. Per ulteriori informazioni, consulta Utilizzo di Active Directory con RDS per SQL Server.

CLI

È possibile associare il gruppo di opzioni a un'istanza database nuova o esistente.

> **Note**
>
> Se si utilizza un'istanza esistente, è necessario che siano già associati un dominio Active Directory e un ruolo IAM. Se si crea una nuova istanza, specificare un dominio Active Directory esistente e un ruolo IAM. Per ulteriori informazioni, consulta Utilizzo di Active Directory con RDS per SQL Server.

Per creare un'istanza database che utilizza il gruppo di opzioni

- Specificare lo stesso tipo di motore database e la versione principale utilizzata durante la creazione del gruppo di opzioni.

  Example

  Per LinuxmacOS, oUnix:

```
aws rds create-db-instance \
    --db-instance-identifier myssasinstance \
    --db-instance-class db.m5.2xlarge \
    --engine sqlserver-se \
    --engine-version 14.00.3223.3.v1 \
    --allocated-storage 100 \
    --manage-master-user-password \
    --master-username admin \
    --storage-type gp2 \
    --license-model li \
    --domain-iam-role-name my-directory-iam-role \
    --domain my-domain-id \
    --option-group-name ssas-se-2017
```

  Per Windows:

```
aws rds create-db-instance ^
    --db-instance-identifier myssasinstance ^
    --db-instance-class db.m5.2xlarge ^
    --engine sqlserver-se ^
```

```
    --engine-version 14.00.3223.3.v1 ^
    --allocated-storage 100 ^
    --manage-master-user-password ^
    --master-username admin ^
    --storage-type gp2 ^
    --license-model li ^
    --domain-iam-role-name my-directory-iam-role ^
    --domain my-domain-id ^
    --option-group-name ssas-se-2017
```

Per modificare un'istanza database per associare il gruppo di opzioni

- Utilizzare uno dei seguenti comandi.

  Example

  Per LinuxmacOS, oUnix:

```
aws rds modify-db-instance \
    --db-instance-identifier myssasinstance \
    --option-group-name ssas-se-2017 \
    --apply-immediately
```

  Per Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier myssasinstance ^
    --option-group-name ssas-se-2017 ^
    --apply-immediately
```

Consentire l'accesso in ingresso al gruppo di sicurezza VPC

Creare una regola in ingresso per la porta del listener SSAS specificata nel gruppo di sicurezza VPC associato all'istanza database. Per ulteriori informazioni sulla configurazione di un gruppo di sicurezza, consulta Fornisci accesso alla istanza database nel VPC creando un gruppo di sicurezza.

Abilitazione dell'integrazione Amazon S3

Per scaricare i file di configurazione del modello nell'host per l'implementazione, utilizza l'integrazione Amazon S3. Per ulteriori informazioni, consulta Integrazione di un'istanza database Amazon RDS for SQL Server con Amazon S3.

Distribuzione di progetti SSAS su Amazon RDS

In RDS, non è possibile distribuire progetti SSAS direttamente utilizzando SQL Server Management Studio (SSMS). Per distribuire i progetti, utilizzare una stored procedure RDS.

> **Note**
>
> L'utilizzo di file xmla per la distribuzione non è supportato.

Prima di distribuire i progetti, assicurarsi che:

- L'integrazione Amazon S3 è attivata. Per ulteriori informazioni, consulta Integrazione di un'istanza database Amazon RDS for SQL Server con Amazon S3.
- L'impostazione `Processing Option` di configurazione è impostata su `Do Not Process`. Questa impostazione indica che non viene eseguita alcuna elaborazione dopo la distribuzione.
- Hai entrambi i file *`myssasproject`*`.asdatabase` e *`myssasproject`*`.deploymentoptions`. Vengono generati automaticamente quando si crea il progetto SSAS.

Per distribuire un progetto SSAS in RDS

1. Scaricare il file `.asdatabase` (modello SSAS) dal bucket S3 all'istanza database, come mostrato nell'esempio seguente. Per ulteriori informazioni sui parametri di download, consulta Download di file da un bucket Amazon S3 in un'istanza database SQL Server.

   ```
   exec msdb.dbo.rds_download_from_s3
   @s3_arn_of_file='arn:aws:s3:::bucket_name/myssasproject.asdatabase',
   [@rds_file_path='D:\S3\myssasproject.asdatabase'],
   [@overwrite_file=1];
   ```

2. Scarica il file `.deploymentoptions` dal bucket S3 all'istanza database.

   ```
   exec msdb.dbo.rds_download_from_s3
   @s3_arn_of_file='arn:aws:s3:::bucket_name/myssasproject.deploymentoptions',
   ```

```
[@rds_file_path='D:\S3\myssasproject.deploymentoptions'],
[@overwrite_file=1];
```

3. Distribuisci il progetto

```
exec msdb.dbo.rds_msbi_task
@task_type='SSAS_DEPLOY_PROJECT',
@file_path='D:\S3\myssasproject.asdatabase';
```

Monitoraggio dello stato di un'attività di distribuzione

Per tenere traccia dello stato dell'attività di distribuzione (o download), chiamare la funzione `rds_fn_task_status`. accetta due parametri. Il primo parametro deve essere sempre `NULL` perché non si applica a SSAS. Il secondo parametro accetta un ID attività.

Per visualizzare l'elenco di tutte le attività, imposta il primo parametro su `NULL` e il secondo parametro su `0`, come indicato nell'esempio seguente.

```
SELECT * FROM msdb.dbo.rds_fn_task_status(NULL,0);
```

Per ottenere un'attività specifica, imposta il primo parametro su `NULL` e il secondo parametro sull'ID attività, come indicato nell'esempio seguente.

```
SELECT * FROM msdb.dbo.rds_fn_task_status(NULL,42);
```

La funzione `rds_fn_task_status` restituisce le seguenti informazioni.

Parametro di output	Descrizione
`task_id`	L'ID dell'attività.
`task_type`	Per SSAS, le attività possono avere i seguenti tipi di attività: • `SSAS_DEPLOY_PROJECT` • `SSAS_ADD_DB_ADMIN_MEMBER` • `SSAS_BACKUP_DB` • `SSAS_RESTORE_DB`

Parametro di output	Descrizione
`database_name`	Non applicabile alle attività SSAS.
`% complete`	L'avanzamento dell'attività espresso come percentuale.
`duration (mins)`	La quantità di tempo dedicato all'attività, in minuti.

Parametro di output	Descrizione
`lifecycle`	Lo stato dell'attività. I possibili stati sono i seguenti: • `CREATED` – Dopo aver chiamato una delle stored procedure SSAS, viene creata un'attività e lo stato viene impostato su `CREATED`. • `IN_PROGRESS` – Dopo l'avvio di un'attività, lo stato viene impostato su `IN_PROGRESS`. Possono essere necessari fino a 5 minuti perché lo stato cambi da `CREATED` in `IN_PROGRESS`. • `SUCCESS` – Al termine di un'attività, lo stato viene impostato su `SUCCESS`. • `ERROR` – Se un'attività non riesce, lo stato viene impostato su `ERROR`. Per ulteriori informazioni sull'errore, consulta la colonna `task_info`. • `CANCEL_REQUESTED` – Quando chiami `rds_cancel_task`, lo stato dell'attività viene impostato su `CANCEL_REQUESTED`. • `CANCELLED` – Dopo che un'attività è stata annullata, lo stato dell'attività viene impostato su `CANCELLED`.

Parametro di output	Descrizione
`task_info`	Ulteriori informazioni sull'attività. Se si verifica un errore durante l'elaborazione, questa colonna contiene informazioni sull'errore. Per ulteriori informazioni, consulta Risoluzione dei problemi SSAS.
`last_updated`	La data e l'ora dell'ultimo aggiornamento dello stato dell'attività.
`created_at`	La data e l'ora di creazione dell'attività.
`S3_object_arn`	Non applicabile alle attività SSAS.
`overwrite_S3_backup_file`	Non applicabile alle attività SSAS.
`KMS_master_key_arn`	Non applicabile alle attività SSAS.
`filepath`	Non applicabile alle attività SSAS.
`overwrite_file`	Non applicabile alle attività SSAS.
`task_metadata`	Metadati associati all'attività SSAS.

Utilizzo di SSAS su Amazon RDS

Dopo aver distribuito il progetto SSAS, è possibile elaborare direttamente il database OLAP in SSMS.

Per utilizzare SSAS in RDS

1. In SSMS, connettersi a SSAS utilizzando il nome utente e la password per il dominio Active Directory.
2. Espandere Databases (Database). Viene visualizzato il database SSAS appena distribuito.

3. Individua la stringa di connessione e aggiorna il nome utente e la password per consentire l'accesso al database SQL di origine. Questa operazione è necessaria per l'elaborazione degli oggetti SSAS.

   a. Per la modalità tabulare, procedi come segue:

      1. Espandi la scheda Connections (Connessioni).
      2. Apri il menu di scelta rapida (tasto destro del mouse) per l'oggetto connessione e quindi scegli Properties (Proprietà).
      3. Aggiorna il nome utente e la password nella stringa di connessione.

   b. Per la modalità multidimensionale, procedi come segue:

      1. Espandi la scheda Data Sources (Origini dati).
      2. Apri il menu di scelta rapida (tasto destro del mouse) per l'oggetto origine dati e quindi scegli Properties (Proprietà).
      3. Aggiorna il nome utente e la password nella stringa di connessione.

4. Aprire il menu di scelta rapida (destro del mouse) per il database SSAS creato e scegliere Process Database (Elabora database).

   A seconda delle dimensioni dei dati di input, l'operazione di elaborazione potrebbe richiedere alcuni minuti per il completamento.

Argomenti

- Configurazione di un utente autenticato da Windows per SSAS
- Aggiunta di un utente di dominio come amministratore di database
- Creazione di un proxy SSAS
- Pianificazione dell'elaborazione del database SSAS utilizzando SQL Server Agent
- Revoca dell'accesso SSAS dal proxy

Configurazione di un utente autenticato da Windows per SSAS

L'utente amministratore principale (a volte chiamato utente master) può utilizzare l'esempio di codice riportato di seguito per impostare un accesso autenticato da Windows e concedere le autorizzazioni necessarie per la procedura. In questo modo vengono concesse autorizzazioni all'utente del dominio per eseguire le attività del cliente SSAS, utilizzare procedure di trasferimento

file S3, creare credenziali e lavorare con il proxy Agente SQL Server. Per ulteriori informazioni, consulta Credentials (Database Engine) e Create a SQL Server Agent Proxy nella documentazione di Microsoft.

È possibile concedere alcune o tutte le autorizzazioni seguenti, se necessario, agli utenti autenticati da Windows.

Example

```
-- Create a server-level domain user login, if it doesn't already exist
USE [master]
GO
CREATE LOGIN [mydomain\user_name] FROM WINDOWS
GO

-- Create domain user, if it doesn't already exist
USE [msdb]
GO
CREATE USER [mydomain\user_name] FOR LOGIN [mydomain\user_name]
GO

-- Grant necessary privileges to the domain user
USE [master]
GO
GRANT ALTER ANY CREDENTIAL TO [mydomain\user_name]
GO

USE [msdb]
GO
GRANT EXEC ON msdb.dbo.rds_msbi_task TO [mydomain\user_name] with grant option
GRANT SELECT ON msdb.dbo.rds_fn_task_status TO [mydomain\user_name] with grant option
GRANT EXEC ON msdb.dbo.rds_task_status TO [mydomain\user_name] with grant option
GRANT EXEC ON msdb.dbo.rds_cancel_task TO [mydomain\user_name] with grant option
GRANT EXEC ON msdb.dbo.rds_download_from_s3 TO [mydomain\user_name] with grant option
GRANT EXEC ON msdb.dbo.rds_upload_to_s3 TO [mydomain\user_name] with grant option
GRANT EXEC ON msdb.dbo.rds_delete_from_filesystem TO [mydomain\user_name] with grant
 option
GRANT EXEC ON msdb.dbo.rds_gather_file_details TO [mydomain\user_name] with grant
 option
GRANT EXEC ON msdb.dbo.sp_add_proxy TO [mydomain\user_name] with grant option
GRANT EXEC ON msdb.dbo.sp_update_proxy TO [mydomain\user_name] with grant option
GRANT EXEC ON msdb.dbo.sp_grant_login_to_proxy TO [mydomain\user_name] with grant
 option
```

```
GRANT EXEC ON msdb.dbo.sp_revoke_login_from_proxy TO [mydomain\user_name] with grant
 option
GRANT EXEC ON msdb.dbo.sp_delete_proxy TO [mydomain\user_name] with grant option
GRANT EXEC ON msdb.dbo.sp_enum_login_for_proxy to [mydomain\user_name] with grant
 option
GRANT EXEC ON msdb.dbo.sp_enum_proxy_for_subsystem TO [mydomain\user_name] with grant
 option
GRANT EXEC ON msdb.dbo.rds_sqlagent_proxy TO [mydomain\user_name] with grant option
ALTER ROLE [SQLAgentUserRole] ADD MEMBER [mydomain\user_name]
GO
```

Aggiunta di un utente di dominio come amministratore di database

È possibile aggiungere un utente di dominio come amministratore del database SSAS nei seguenti modi:

- Un amministratore di database può utilizzare SSMS per creare un ruolo con privilegi `admin`, quindi aggiungere utenti a tale ruolo.
- È possibile utilizzare la seguente stored procedure.

  ```
  exec msdb.dbo.rds_msbi_task
  @task_type='SSAS_ADD_DB_ADMIN_MEMBER',
  @database_name='myssasdb',
  @ssas_role_name='exampleRole',
  @ssas_role_member='domain_name\domain_user_name';
  ```

  I parametri seguenti sono obbligatori:

  - `@task_type` – Il tipo di attività MSBI, in questo caso `SSAS_ADD_DB_ADMIN_MEMBER`.
  - `@database_name` – Il nome del database SSAS a cui si concedono i privilegi di amministratore.
  - `@ssas_role_name` – Il nome del ruolo amministratore del database SSAS. Se il ruolo non esiste già, viene creato.
  - `@ssas_role_member` – L'utente del database SSAS che si sta aggiungendo al ruolo di amministratore.

Creazione di un proxy SSAS

Per pianificare l'elaborazione del database SSAS utilizzando SQL Server Agent, crea una credenziale SSAS e un proxy SSAS. Eseguire queste procedure come utente autenticato da Windows.

Per creare le credenziali SSAS

- Creare le credenziali per il proxy. A tale scopo, è possibile utilizzare SSMS o la seguente istruzione SQL.

```
USE [master]
GO
CREATE CREDENTIAL [SSAS_Credential] WITH IDENTITY = N'mydomain\user_name', SECRET =
 N'mysecret'
GO
```

> **Note**
>
> IDENTITY deve essere un accesso autenticato dal dominio. Sostituire *mysecret* con la password per l'accesso autenticato dal dominio.

Per creare il proxy SSAS

1. Utilizzare l'istruzione SQL seguente per creare il proxy.

```
USE [msdb]
GO
EXEC msdb.dbo.sp_add_proxy
 @proxy_name=N'SSAS_Proxy',@credential_name=N'SSAS_Credential',@description=N''
GO
```

2. Utilizzare l'istruzione SQL seguente per concedere l'accesso al proxy ad altri utenti.

```
USE [msdb]
GO
EXEC msdb.dbo.sp_grant_login_to_proxy
 @proxy_name=N'SSAS_Proxy',@login_name=N'mydomain\user_name'
GO
```

3. Utilizzare la seguente istruzione SQL per dare al sottosistema SSAS l'accesso al proxy.

```
USE [msdb]
GO
EXEC msdb.dbo.rds_sqlagent_proxy
 @task_type='GRANT_SUBSYSTEM_ACCESS',@proxy_name='SSAS_Proxy',@proxy_subsystem='SSAS'
```

```
GO
```

Per visualizzare il proxy e le concessioni sul proxy

1. Utilizzare l'istruzione SQL seguente per visualizzare gli assegnatari del proxy.

```
USE [msdb]
GO
EXEC sp_help_proxy
GO
```

2. Utilizzare l'istruzione SQL seguente per visualizzare i privilegi del sottosistema.

```
USE [msdb]
GO
EXEC msdb.dbo.sp_enum_proxy_for_subsystem
GO
```

Pianificazione dell'elaborazione del database SSAS utilizzando SQL Server Agent

Dopo aver creato le credenziali e il proxy e dopo aver concesso l'accesso SSAS al proxy, è possibile creare un processo Agente SQL Server per pianificare l'elaborazione del database SSAS.

Per pianificare l'elaborazione del database SSAS

- Usa SSMS o T-SQL per creare il processo SQL Server Agent. L'esempio seguente utilizza T-SQL. Puoi configurare ulteriormente la pianificazione dei processi tramite SSMS o T-SQL.
  - Il parametro `@command` delinea il comando XML for Analysis (XMLA) da eseguire tramite il processo SQL Server Agent. Questo esempio configura l'elaborazione del database multidimensionale SSAS.
  - Il parametro `@server` delinea il nome del server SSAS di destinazione del processo SQL Server Agent.

    Per chiamare il servizio SSAS all'interno della stessa istanza database RDS in cui risiede il processo SQL Server Agent, utilizza `localhost:2383`.

    Per chiamare il servizio SSAS dall'esterno dell'istanza database RDS, utilizza l'endpoint RDS. Puoi anche utilizzare l'endpoint Kerberos Active Directory (AD) (*your-DB-instance-*

name.*your-AD-domain-name*) se le istanze database RDS sono unite dallo stesso dominio. Per le istanze database esterne, assicurati di configurare correttamente il gruppo di sicurezza VPC associato all'istanza database RDS per una connessione sicura.

Puoi modificare ulteriormente la query per supportare varie operazioni XMLA. Apporta le modifiche modificando direttamente la query T-SQL o utilizzando l'interfaccia utente SSMS dopo la creazione del processo di SQL Server Agent.

```
USE [msdb]
GO
DECLARE @jobId BINARY(16)
EXEC msdb.dbo.sp_add_job @job_name=N'SSAS_Job',
    @enabled=1,
    @notify_level_eventlog=0,
    @notify_level_email=0,
    @notify_level_netsend=0,
    @notify_level_page=0,
    @delete_level=0,
    @category_name=N'[Uncategorized (Local)]',
    @job_id = @jobId OUTPUT
GO
EXEC msdb.dbo.sp_add_jobserver
    @job_name=N'SSAS_Job',
    @server_name = N'(local)'
GO
EXEC msdb.dbo.sp_add_jobstep @job_name=N'SSAS_Job',
 @step_name=N'Process_SSAS_Object',
    @step_id=1,
    @cmdexec_success_code=0,
    @on_success_action=1,
    @on_success_step_id=0,
    @on_fail_action=2,
    @on_fail_step_id=0,
    @retry_attempts=0,
    @retry_interval=0,
    @os_run_priority=0, @subsystem=N'ANALYSISCOMMAND',
    @command=N'<Batch xmlns="http://schemas.microsoft.com/analysisservices/2003/
engine">
        <Parallel>
            <Process xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```
                xmlns:ddl2="http://schemas.microsoft.com/analysisservices/2003/
engine/2" xmlns:ddl2_2="http://schemas.microsoft.com/analysisservices/2003/
engine/2/2"
                xmlns:ddl100_100="http://schemas.microsoft.com/
analysisservices/2008/engine/100/100" xmlns:ddl200="http://schemas.microsoft.com/
analysisservices/2010/engine/200"
                xmlns:ddl200_200="http://schemas.microsoft.com/
analysisservices/2010/engine/200/200" xmlns:ddl300="http://schemas.microsoft.com/
analysisservices/2011/engine/300"
                xmlns:ddl300_300="http://schemas.microsoft.com/
analysisservices/2011/engine/300/300" xmlns:ddl400="http://schemas.microsoft.com/
analysisservices/2012/engine/400"
                xmlns:ddl400_400="http://schemas.microsoft.com/
analysisservices/2012/engine/400/400" xmlns:ddl500="http://schemas.microsoft.com/
analysisservices/2013/engine/500"
                xmlns:ddl500_500="http://schemas.microsoft.com/
analysisservices/2013/engine/500/500">
                <Object>
                    <DatabaseID>Your_SSAS_Database_ID</DatabaseID>
                </Object>
                <Type>ProcessFull</Type>
                <WriteBackTableCreation>UseExisting</WriteBackTableCreation>
            </Process>
        </Parallel>
    </Batch>',
    @server=N'localhost:2383',
    @database_name=N'master',
    @flags=0,
    @proxy_name=N'SSAS_Proxy'
GO
```

Revoca dell'accesso SSAS dal proxy

È possibile revocare l'accesso al sottosistema SSAS ed eliminare il proxy SSAS utilizzando le seguenti procedure memorizzate.

Per revocare l'accesso ed eliminare il proxy

1. Revocare l'accesso al sottosistema.

```
USE [msdb]
GO
```

```
EXEC msdb.dbo.rds_sqlagent_proxy
 @task_type='REVOKE_SUBSYSTEM_ACCESS',@proxy_name='SSAS_Proxy',@proxy_subsystem='SSAS'
GO
```

2. Revocare le concessioni per la delega.

```
USE [msdb]
GO
EXEC msdb.dbo.sp_revoke_login_from_proxy
 @proxy_name=N'SSAS_Proxy',@name=N'mydomain\user_name'
GO
```

3. Eliminare il proxy.

```
USE [msdb]
GO
EXEC dbo.sp_delete_proxy @proxy_name = N'SSAS_Proxy'
GO
```

Backup di un database SSAS

È possibile creare file di backup del database SSAS solo nella cartella `D:\S3` dell'istanza database. Per spostare i file di backup nel bucket S3, utilizzare Amazon S3.

È possibile eseguire il backup di un database SSAS come segue:

- Un utente di dominio con il ruolo `admin` di un determinato database può utilizzare SSMS per eseguire il backup del database nella cartella `D:\S3`.

  Per ulteriori informazioni, consulta Aggiunta di un utente di dominio come amministratore di database.
- È possibile utilizzare la seguente stored procedure. Questa stored procedure non supporta la crittografia.

```
exec msdb.dbo.rds_msbi_task
@task_type='SSAS_BACKUP_DB',
@database_name='myssasdb',
@file_path='D:\S3\ssas_db_backup.abf',
[@ssas_apply_compression=1],
[@ssas_overwrite_file=1];
```

I parametri seguenti sono obbligatori:

- `@task_type` – Il tipo di attività MSBI, in questo caso `SSAS_BACKUP_DB`.
- `@database_name` – Il nome del database SSAS di cui si esegue il backup.
- `@file_path` – Il percorso del file di backup SSAS. L'estensione `.abf` è obbligatoria.

I parametri seguenti sono facoltativi:

- `@ssas_apply_compression` – Indica se applicare la compressione di backup SSAS. I valori validi sono 1 (Sì) e 0 (No).
- `@ssas_overwrite_file` – Indica se sovrascrivere il file di backup SSAS. I valori validi sono 1 (Sì) e 0 (No).

Ripristino di un database SSAS

Utilizzare la seguente stored procedure per ripristinare un database SSAS da un backup.

Non è possibile ripristinare un database se esiste già un database SSAS con lo stesso nome. La stored procedure per il ripristino non supporta i file di backup crittografati.

```
exec msdb.dbo.rds_msbi_task
@task_type='SSAS_RESTORE_DB',
@database_name='mynewssasdb',
@file_path='D:\S3\ssas_db_backup.abf';
```

I parametri seguenti sono obbligatori:

- `@task_type` – Il tipo di attività MSBI, in questo caso `SSAS_RESTORE_DB`.
- `@database_name` – Il nome del nuovo database SSAS che si sta ripristinando.
- `@file_path` – Il percorso del file di backup SSAS.

Ripristino a un'ora specifica per un'istanza database

P oint-in-time recovery (PITR) non si applica ai database SSAS. Se si esegue PITR, solo i dati SSAS nell'ultimo snapshot prima dell'ora richiesta sono disponibili nell'istanza ripristinata.

Avere database up-to-date SSAS su un'istanza DB ripristinata

1. Eseguire il backup dei database SSAS nella cartella `D:\S3` dell'istanza di origine.

2. Trasferire i file di backup nel bucket S3.
3. Trasferire i file di backup dal bucket S3 alla cartella `D:\S3` sull'istanza ripristinata.
4. Eseguire la stored procedure per ripristinare i database SSAS nell'istanza ripristinata.

   È inoltre possibile rielaborare il progetto SSAS per ripristinare i database.

Modifica della modalità SSAS

Puoi modificare la modalità in cui viene eseguito SSAS, Tabular (Tabulare) o Multidimensional (Multidimensionale). Per cambiare la modalità, usa AWS Management Console o AWS CLI per modificare le impostazioni delle opzioni nell'opzione SSAS.

> **Important**
>
> Puoi utilizzare una sola modalità SSAS alla volta. Assicurati di eliminare tutti i database SSAS prima di cambiare la modalità, altrimenti riceverai un errore.

Console

La seguente procedura della console Amazon RDS cambia la modalità SSAS in Tabular (Tabulare) e imposta il parametro `MAX_MEMORY` al 70 percento.

Per modificare l'opzione SSAS

1. Accedi AWS Management Console e apri la console Amazon RDS all'[indirizzo https://console.aws.amazon.com/rds/](https://console.aws.amazon.com/rds/).
2. Nel pannello di navigazione scegliere Option groups (Gruppi di opzioni).
3. Scegli il gruppo di opzioni con l'opzione `SSAS` che desideri modificare (`ssas-se-2017` negli esempi precedenti).
4. Scegli Modify option (Modifica l'opzione).
5. Modifica le impostazioni delle opzioni:
   a. Per Max memory (Memoria massima), inserisci **`70`**.
   b. Per Mode (Modalità), scegli Tabular (Tabulare).
6. Scegli Modify option (Modifica l'opzione).

AWS CLI

L' AWS CLI esempio seguente modifica la modalità SSAS in Tabular e imposta il MAX_MEMORY parametro al 70 percento.

Affinché il comando CLI funzioni, assicurati di includere tutti i parametri richiesti, anche se non li stai modificando.

Per modificare l'opzione SSAS

- Utilizzare uno dei seguenti comandi.

  Example

  PerLinux, omacOS: Unix

```
aws rds add-option-to-option-group \
    --option-group-name ssas-se-2017 \
    --options
 "OptionName=SSAS,VpcSecurityGroupMemberships=sg-12345e67,OptionSettings=[{Name=MAX_MEMORY,
{Name=MODE,Value=Tabular}]" \
    --apply-immediately
```

  Per Windows:

```
aws rds add-option-to-option-group ^
    --option-group-name ssas-se-2017 ^
    --options
 OptionName=SSAS,VpcSecurityGroupMemberships=sg-12345e67,OptionSettings=[{Name=MAX_MEMORY,V
{Name=MODE,Value=Tabular}] ^
    --apply-immediately
```

Disattivazione di SSAS

Per disattivare SSAS, rimuovi l'opzione SSAS dal relativo gruppo di opzioni.

> **Important**
>
> Prima di rimuovere l'opzione SSAS, eliminare i database SSAS.

Si consiglia di eseguire il backup dei database SSAS prima di eliminarli e rimuovere l'opzione SSAS.

Console

Per rimuovere l'opzione SSAS dal suo gruppo di opzioni

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel pannello di navigazione scegliere Option groups (Gruppi di opzioni).
3. Scegli il gruppo di opzioni con l'opzione SSAS che desideri rimuovere (`ssas-se-2017` negli esempi precedenti).
4. Scegliere Delete option (Elimina opzione).
5. In Deletion options (Opzioni di eliminazione), scegliere SSAS per Options to delete (Opzioni da eliminare).
6. In Apply immediately (Applica immediatamente), scegliere Yes (Sì) per eliminare immediatamente l'opzione oppure No per eliminarla nella finestra di manutenzione successiva.
7. Scegliere Delete (Elimina).

AWS CLI

Per rimuovere l'opzione SSAS dal suo gruppo di opzioni

- Utilizzare uno dei seguenti comandi.

  Example

  Per LinuxmacOS, oUnix:

```
aws rds remove-option-from-option-group \
    --option-group-name ssas-se-2017 \
    --options SSAS \
    --apply-immediately
```

  Per Windows:

```
aws rds remove-option-from-option-group ^
```

```
    --option-group-name ssas-se-2017 ^
    --options SSAS ^
    --apply-immediately
```

Risoluzione dei problemi SSAS

È possibile che si verifichino i seguenti problemi durante l'utilizzo di SSAS.

Problema	Type	Suggerimenti sulla risoluzione dei problemi
Impossibile configurare l'opzione SSAS. La modalità SSAS richiesta è *new_mode*, ma l'attuale istanza database ha *number* database *current_mode* . Elimina i database esistenti prima di passare alla modalità *new_mode*. Per riottenere l'accesso alla modalità *current_mode* per l'eliminazione del database, aggiorna il gruppo di opzioni del database corrente oppure collega un nuovo gruppo di opzioni con %s come valore di impostazione dell'opzione MODE per l'opzione SSAS.	Evento RDS	Non puoi modificare la modalità SSAS se disponi ancora di database SSAS che utilizzano la modalità corrente. Elimina i database SSAS, quindi riprova.
Impossibile rimuovere l'opzione SSAS perché esistono *number* database *mode*. L'opzione SSAS non può essere rimossa finché non vengono eliminati tutti i database SSAS. Aggiungi nuovamente l'opzione SSAS, elimina tutti i database SSAS e riprova.	Evento RDS	Non puoi disattivare SSAS se disponi ancora di database SSAS. Elimina i database SSAS, quindi riprova.
L'opzione SSAS non è abilitata o è in fase di abilitazione. Riprova più tardi.	Stored procedure RDS	Non puoi eseguire stored procedure SSAS quando l'opzione è disattivata o quando è attivata.

Problema	Type	Suggerimenti sulla risoluzione dei problemi
L'opzione SSAS non è configura ta correttamente. Assicurati che lo stato di appartenenza al gruppo di opzioni sia "in-sync" e controlla i log di eventi RDS per i messaggi di errore di configurazione SSAS pertinenti. Dopo questi controlli, riprova. Se gli errori persistono, contatta l' AWS assistenz a.	Stored procedure RDS	Non è possibile eseguire stored procedure SSAS quando l'appartenenza al gruppo di opzioni non è presente nello stato `in-sync`. Questa operazione mette l'opzione SSAS in uno stato di configurazione errato. Se lo stato di appartenenza al gruppo di opzioni cambia in `failed` a causa della modifica dell'opzione SSAS, ci sono due possibili motivi: 1. L'opzione SSAS è stata rimossa senza l'eliminazione dei database SSAS. 2. La modalità SSAS è stata aggiornata da Tabular (Tabulare) a Multidimensional (Multidimensionale) o da Multidime nsional (Multidimensionale) a Tabular (Tabulare), senza che i database SSAS esistenti venissero eliminati. Riconfigura l'opzione SSAS, poiché RDS consente solo una modalità SSAS alla volta e non supporta la rimozione delle opzioni SSAS con i database SSAS presenti. Controlla i log di eventi RDS per verificar e la presenza di errori di configurazione per l'istanza SSAS e risolvere i problemi di conseguenza.

Problema	Type	Suggerimenti sulla risoluzione dei problemi
Implementazione non riuscita. La modifica può essere implementata solo su un server in esecuzione in modalità *deployment_file_mode* . La modalità server attuale è *current_mode* .	Stored procedure RDS	Non puoi implementare un database tabulare su un server multidimensionale o un database multidimensionale su un server tabulare. Assicurati di utilizzare file con la modalità corretta e verifica che l'impostazione dell'opzione `MODE` sia impostata sul valore appropriato.
Ripristino non riuscito. Il file di backup può essere ripristinato solo su un server in esecuzione in modalità *restore_file_mode* . La modalità server attuale è *current_mode* .	Stored procedure RDS	Non puoi ripristinare un database tabulare su un server multidimensionale o un database multidimensionale su un server tabulare. Assicurati di utilizzare file con la modalità corretta e verifica che l'impostazione dell'opzione `MODE` sia impostata sul valore appropriato.
Ripristino non riuscito. Il file di backup e le versioni dell'istanza database RDS non sono compatibili.	Stored procedure RDS	Non puoi ripristinare un database SSAS con una versione incompatibile con la versione dell'istanza di SQL Server. Per ulteriori informazioni, consulta Compatibility levels for tabular models (Livelli di compatibilità per modelli tabulari) e Compatibility level of a multidimensional database (Livello di compatibilità per un database multidimensionale) nella documentazione di Microsoft.

Problema	Type	Suggerimenti sulla risoluzione dei problemi
Ripristino non riuscito. Il file di backup specificato nell'operazione di ripristino è danneggiato o non è un file di backup SSAS. Assicurati che @rds_file_path sia formattato correttamente.	Stored procedure RDS	Non puoi ripristinare un database SSAS con un file danneggiato. Assicurati che il file non sia danneggiato o corrotto. Questo errore può anche essere generato quando `@rds_file_path` non è formattato correttamente (ad esempio, ha doppie barre rovesciate come in `D:\S3\\incorrect_format.abf`).
Ripristino non riuscito. Il nome del database ripristinato non può contenere parole riservate o caratteri non validi: . , ; ' ` : / \\ * | ? \" & % $! + = ( ) [ ] { } < > o avere più di 100 caratteri.	Stored procedure RDS	Il nome del database ripristinato non può contenere parole riservate o caratteri non validi o avere più di 100 caratteri. Per le convenzioni di denominazione degli oggetti SSAS, consulta Object naming rules (Regole di denominazione degli oggetti) nella documentazione di Microsoft.
È stato fornito un nome del ruolo non valido. Il nome del ruolo non può contenere stringhe riservate.	Stored procedure RDS	Il nome del ruolo non può contenere stringhe riservate. Per le convenzioni di denominazione degli oggetti SSAS, consulta Object naming rules (Regole di denominazione degli oggetti) nella documentazione di Microsoft.
È stato fornito un nome del ruolo non valido. Il nome del ruolo non può contenere nessuno dei seguenti caratteri riservati: . , ; ' ` : / \\ * | ? \" & % $! + = ( ) [ ] { } < >	Stored procedure RDS	Il nome del ruolo non può contenere caratteri riservati. Per le convenzioni di denominazione degli oggetti SSAS, consulta Object naming rules (Regole di denominazione degli oggetti) nella documentazione di Microsoft.

Supporto per SQL Server Integration Services in Amazon RDS for SQL Server

Microsoft SQL Server Integration Services (SSIS) è un componente che è possibile utilizzare per eseguire un'ampia gamma di attività di migrazione dei dati. SSIS è una piattaforma per l'integrazione dei dati e le applicazioni di workflow. È dotato di uno strumento di data warehousing utilizzato per l'estrazione, la trasformazione e il caricamento dei dati (ETL). È inoltre possibile utilizzare questo strumento per automatizzare la manutenzione dei database di SQL Server e degli aggiornamenti ai dati del cubo multidimensionale.

I progetti SSIS sono organizzati in pacchetti salvati come file dtsx basati su XML. I pacchetti possono contenere flussi di controllo e flussi di dati. I flussi di dati vengono utilizzati per rappresentare le operazioni ETL. Dopo la distribuzione, i pacchetti vengono archiviati in SQL Server nel database SSISDB. SSISDB è un database OLTP (Online Transaction Processing) in modalità di recupero completo.

Amazon RDS for SQL Server supporta l'esecuzione di SSIS direttamente su istanze database RDS. È possibile abilitare SSIS su un'istanza database esistente o nuova. È installato sulla stessa istanza database del motore di database.

RDS supporta SSIS per SQL Server Standard ed Enterprise Edition nelle seguenti versioni:

- SQL Server 2022, tutte le versioni
- SQL Server 2019, versione 15.00.4043.16.v1 e successive
- SQL Server 2017, versione 14.00.3223.3.v1 e successive
- SQL Server 2016, versione 13.00.5426.0.v1 e successive

Indice

- Abilitazione dell'integrazione di S3
- Autorizzazioni amministrative su SSISDB
  - Configurazione di un utente autenticato da Windows per SSIS
- Distribuzione di un progetto SSIS
- Monitoraggio dello stato di un'attività di distribuzione
- Utilizzo di SSIS
  - Impostazione di gestori di connessioni al database per i progetti SSIS
  - Creazione di un proxy SSIS
  - Pianificazione di un pacchetto SSIS utilizzando SQL Server Agent
  - Revoca dell'accesso SSIS dal proxy
- Disabilitazione di SSIS
- Eliminazione del database SSISDB

Limitazioni e consigli

Le limitazioni e i suggerimenti riportati di seguito si applicano all'esecuzione di SSIS su RDS per SQL Server:

- L'istanza database deve avere un gruppo di parametri associato con il parametro `clr enabled` impostato su 1. Per ulteriori informazioni, consulta Modifica del parametro per SSIS.

  > **Note**
  >
  > Se si abilita il parametro `clr enabled` su SQL Server 2017 o 2019, non è possibile utilizzare il Common Language Runtime (CLR) sull'istanza database. Per ulteriori informazioni, consulta Caratteristiche non supportate e caratteristiche con supporto limitato.

- Sono supportate le seguenti attività del flusso di controllo:
  - Analysis Services esegue task DDL
  - Attività di elaborazione di Analysis Services
  - Attività di inserimento in blocco
  - Verifica dell'attività di integrità del database
  - Attività flusso di dati
  - Attività di query di data mining

  - Attività di profilazione dati
  - Esecuzione di attività del pacchetto
  - Esecuzione di attività del processo agente SQL Server
  - Esecuzione di attività SQL
  - Esecuzione di attività istruzione T-SQL
  - Notifica di attività operatore
  - Ricostruzione attività indice
  - Riorganizzazione attività indice
  - Riduzione attività del database
  - Trasferimento attività database
  - Trasferimento attività processo
  - Trasferimento operazioni di accesso
  - Trasferimento attività oggetti SQL Server
  - Aggiornamento attività statistiche
- È supportata solo la distribuzione del progetto.
- È supportata l'esecuzione di pacchetti SSIS utilizzando SQL Server Agent.
- I record di log SSIS possono essere inseriti solo nei database creati dall'utente.
- Utilizzare solo la cartella `D:\S3` per lavorare con i file. I file inseriti in qualsiasi altra directory vengono eliminati. A questo punto è necessario conoscere alcuni altri dettagli sulla posizione dei file:
  - Inserire i file di input e output del progetto SSIS nella cartella `D:\S3`.
  - Per l'attività Flusso di dati, modificare il percorso per `BLOBTempStoragePath` e `BufferTempStoragePath` su un file all'interno della cartella `D:\S3`. Il percorso del file deve iniziare con `D:\S3\`.
  - Assicurarsi che tutti i parametri, le variabili e le espressioni utilizzate per le connessioni ai file puntino alla cartella `D:\S3`.
  - Nelle istanze Multi-AZ, i file creati da SSIS nella cartella `D:\S3` vengono eliminati dopo un failover. Per ulteriori informazioni, consulta Limitazioni Multi-AZ per l'integrazione S3.
  - Carica i file creati da SSIS nella cartella `D:\S3` nel tuo bucket Amazon S3 per renderli durevoli.
- Le trasformazioni Importa colonne ed Esporta colonne e il componente Script nell'attività Flusso di dati non sono supportati.

- Non è possibile abilitare il dump sull'esecuzione del pacchetto SSIS e non è possibile aggiungere dati ai pacchetti SSIS.
- La funzionalità Scale Out SSIS non è supportata.
- Non è possibile distribuire direttamente i progetti. Forniamo procedure archiviate RDS per farlo. Per ulteriori informazioni, consulta Distribuzione di un progetto SSIS.
- Creare file di progetto SSIS (.ispac) con la modalità di protezione `DoNotSavePasswords` per la distribuzione su RDS.
- SSIS non è supportato nelle istanze Always On con repliche di lettura.
- Non è possibile eseguire il backup del database SSISDB associato all'opzione `SSIS`.
- L'importazione e il ripristino del database SSISDB da altre istanze di SSIS non sono supportati.
- È possibile connettersi ad altre istanze database SQL Server o a un'origine dei dati Oracle. La connessione ad altri motori di database, come MySQL o PostgreSQL, non è supportata per SSIS su RDS per SQL Server. Per ulteriori informazioni sulla connessione a un'origine dei dati Oracle, consulta Server collegati con Oracle OLEDB.

Abilitazione di SSIS

Si abilita SSIS aggiungendo l'opzione SSIS all'istanza database. Utilizzare il seguente processo:

1. Creare un nuovo gruppo di opzioni oppure utilizzare un gruppo di opzioni esistente.
2. Aggiungere l'opzione `SSIS` al gruppo di opzioni.
3. Creare un nuovo gruppo di parametri o scegliere un gruppo di parametri esistente.
4. Modificare il gruppo di parametri per impostare il parametro `clr enabled` su 1 o 2.
5. Associare il gruppo di opzioni e il gruppo di parametri all'istanza database.
6. Abilita l'integrazione Amazon S3.

> **Note**
>
> Se un database con il nome SSISDB o un account di accesso SSIS riservato esiste già nell'istanza database, non è possibile abilitare SSIS sull'istanza.

Creazione del gruppo di opzioni per SSIS

Per utilizzare SSIS, creare un gruppo di opzioni o modificare un gruppo di opzioni che corrisponda all'edizione di SQL Server e alla versione dell'istanza database che si intende utilizzare. A questo scopo, utilizzare la AWS Management Console o AWS CLI.

Console

La seguente procedura crea un gruppo di opzioni per SQL Server Standard Edition 2016.

Per creare il gruppo di opzioni

1. Accedi alla AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel pannello di navigazione scegliere Option groups (Gruppi di opzioni).
3. Seleziona Create group (Crea gruppo).
4. Nella finestra Create option group (Crea gruppo di opzioni) eseguire queste operazioni:
    a. Per Nome, immettere un nome per il gruppo di opzioni che sia univoco all'interno dell'account AWS, ad esempio **`ssis-se-2016`**. Il nome può includere solo lettere, cifre e trattini.
    b. Per Descrizione, immettere una breve descrizione del gruppo di opzioni, ad esempio **`SSIS option group for SQL Server SE 2016`**. La descrizione viene usata per la visualizzazione.
    c. Per Engine (Motore), scegliere sqlserver-se.
    d. Per Versione del motore principale, scegliere 13.00.
5. Scegliere Create (Crea).

CLI

La seguente procedura crea un gruppo di opzioni per SQL Server Standard Edition 2016.

Per creare il gruppo di opzioni

- Eseguire uno dei seguenti comandi.

    Example

    Per LinuxmacOS, oUnix:

```
aws rds create-option-group \
    --option-group-name ssis-se-2016 \
    --engine-name sqlserver-se \
    --major-engine-version 13.00 \
    --option-group-description "SSIS option group for SQL Server SE 2016"
```

Per Windows:

```
aws rds create-option-group ^
    --option-group-name ssis-se-2016 ^
    --engine-name sqlserver-se ^
    --major-engine-version 13.00 ^
    --option-group-description "SSIS option group for SQL Server SE 2016"
```

Aggiunta dell'opzione SSIS al gruppo di opzioni

Utilizzare la AWS Management Console o l'AWS CLI per aggiungere l'opzione `SSIS` al gruppo di opzioni.

Console

Per aggiungere l'opzione SSIS

1. Accedi alla AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione scegliere Option groups (Gruppi di opzioni).
3. Scegliere il gruppo di opzioni appena creato, ssis-se-2016 in questo esempio.
4. Scegliere Add option (Aggiungi opzione).
5. In Dettagli opzione, scegliere SSIS per Nome opzione.
6. In Scheduling (Pianificazione), scegliere se aggiungere l'opzione immediatamente o alla finestra di manutenzione successiva.
7. Scegliere Add option (Aggiungi opzione).

CLI

Per aggiungere l'opzione SSIS

- Aggiungere l'opzione `SSIS` al gruppo di opzioni.

  Example

  Per LinuxmacOS, oUnix:

```
aws rds add-option-to-option-group \
    --option-group-name ssis-se-2016 \
    --options OptionName=SSIS \
    --apply-immediately
```

  Per Windows:

```
aws rds add-option-to-option-group ^
    --option-group-name ssis-se-2016 ^
    --options OptionName=SSIS ^
    --apply-immediately
```

Creazione del gruppo di parametri per SSIS

Creare o modificare un gruppo di parametri per il parametro `clr enabled` corrispondente all'edizione di SQL Server e alla versione dell'istanza database che si pianifica di utilizzare per SSIS.

Console

Nella procedura seguente viene creato un gruppo di parametri per SQL Server Standard Edition 2016.

Per creare il gruppo di parametri

1. Accedi alla AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel pannello di navigazione, scegli Parameter groups (Gruppi di parametri).
3. Scegliere Create parameter group (Crea gruppo di parametri).
4. Nel riquadro Create parameter group (Crea gruppi di parametri), procedi nel modo seguente:

a. Per Famiglia del gruppo di parametri, scegliere sqlserver-se-13.0.

b. Per Group name (Nome gruppo), immettere un identificatore per il gruppo di parametri, ad esempio **`ssis-sqlserver-se-13`**.

c. Per Description (Descrizione), immettere **`clr enabled parameter group`**.

5. Scegliere Create (Crea).

CLI

Nella procedura seguente viene creato un gruppo di parametri per SQL Server Standard Edition 2016.

Per creare il gruppo di parametri

- Eseguire uno dei seguenti comandi.

  Example

  Per LinuxmacOS, oUnix:

```
aws rds create-db-parameter-group \
    --db-parameter-group-name ssis-sqlserver-se-13 \
    --db-parameter-group-family "sqlserver-se-13.0" \
    --description "clr enabled parameter group"
```

  Per Windows:

```
aws rds create-db-parameter-group ^
    --db-parameter-group-name ssis-sqlserver-se-13 ^
    --db-parameter-group-family "sqlserver-se-13.0" ^
    --description "clr enabled parameter group"
```

Modifica del parametro per SSIS

Modifica il parametro `clr enabled` nel gruppo di parametri che corrisponde all'edizione di SQL Server e alla versione dell'istanza database. Per SSIS, impostare il parametro `clr enabled` su 1.

Console

Nella procedura seguente, il gruppo di parametri creato per SQL Server Standard Edition 2016 viene modificato.

Per modificare il gruppo di parametri

1. Accedi alla AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione scegliere Parameter groups (Gruppi di parametri).
3. Scegliete il gruppo di parametri, ad esempio ssis-sqlserver-se-13.
4. In Parameters (Parametri), filtrare l'elenco dei parametri per **`clr`**.
5. Scegliere clr abilitato.
6. Scegliere Edit parameters (Modifica parametri).
7. Da Valori, scegliere 1.
8. Seleziona Save changes (Salva modifiche).

CLI

Nella procedura seguente, il gruppo di parametri creato per SQL Server Standard Edition 2016 viene modificato.

Per modificare il gruppo di parametri

- Eseguire uno dei seguenti comandi.

  Example

  Per LinuxmacOS, oUnix:

```
aws rds modify-db-parameter-group \
    --db-parameter-group-name ssis-sqlserver-se-13 \
    --parameters "ParameterName='clr
 enabled',ParameterValue=1,ApplyMethod=immediate"
```

  Per Windows:

```
aws rds modify-db-parameter-group ^
    --db-parameter-group-name ssis-sqlserver-se-13 ^
```

```
    --parameters "ParameterName='clr
enabled',ParameterValue=1,ApplyMethod=immediate"
```

Associazione del gruppo di opzioni e del gruppo di parametri all'istanza database

Per associare il gruppo di opzioni SSIS e il gruppo di parametri all'istanza database, utilizzare la AWS Management Console o il comando AWS CLI

> **Note**
>
> Se si utilizza un'istanza esistente, è necessario che siano già associati un dominio Active Directory e un ruolo AWS Identity and Access Management (IAM) associato. Se si crea una nuova istanza, specificare un dominio Active Directory esistente e un ruolo IAM. Per ulteriori informazioni, consulta Utilizzo di Active Directory con RDS per SQL Server.

Console

Per completare l'abilitazione di SSIS, associare il gruppo di opzioni SSIS e il gruppo di parametri a un'istanza database nuova o esistente:

- Per una nuova istanza database, associarli all'avvio dell'istanza. Per ulteriori informazioni, consulta Creazione di un'istanza database Amazon RDS.
- Per un'istanza database esistente, associarli modificando l'istanza. Per ulteriori informazioni, consulta Modifica di un'istanza database Amazon RDS.

CLI

Puoi associare il gruppo di opzioni SSIS e il gruppo di parametri a un'istanza database nuova o esistente.

Per creare un'istanza con il gruppo di opzioni SSIS e il gruppo di parametri

- Specificare lo stesso tipo di motore di database e la versione principale utilizzati durante la creazione del gruppo di opzioni.

  Example

  Per LinuxmacOS, oUnix:

```
aws rds create-db-instance \
    --db-instance-identifier myssisinstance \
    --db-instance-class db.m5.2xlarge \
    --engine sqlserver-se \
    --engine-version 13.00.5426.0.v1 \
    --allocated-storage 100 \
    --manage-master-user-password \
    --master-username admin \
    --storage-type gp2 \
    --license-model li \
    --domain-iam-role-name my-directory-iam-role \
    --domain my-domain-id \
    --option-group-name ssis-se-2016 \
    --db-parameter-group-name ssis-sqlserver-se-13
```

Per Windows:

```
aws rds create-db-instance ^
    --db-instance-identifier myssisinstance ^
    --db-instance-class db.m5.2xlarge ^
    --engine sqlserver-se ^
    --engine-version 13.00.5426.0.v1 ^
    --allocated-storage 100 ^
    --manage-master-user-password ^
    --master-username admin ^
    --storage-type gp2 ^
    --license-model li ^
    --domain-iam-role-name my-directory-iam-role ^
    --domain my-domain-id ^
    --option-group-name ssis-se-2016 ^
    --db-parameter-group-name ssis-sqlserver-se-13
```

Per modificare un'istanza database e associare il gruppo di opzioni SSIS e il gruppo di parametri

- Eseguire uno dei seguenti comandi.

  Example

  Per LinuxmacOS, oUnix:

```
aws rds modify-db-instance \
    --db-instance-identifier myssisinstance \
    --option-group-name ssis-se-2016 \
    --db-parameter-group-name ssis-sqlserver-se-13 \
    --apply-immediately
```

Per Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier myssisinstance ^
    --option-group-name ssis-se-2016 ^
    --db-parameter-group-name ssis-sqlserver-se-13 ^
    --apply-immediately
```

Abilitazione dell'integrazione di S3

Per scaricare i file del progetto SSIS (.ispac) sull'host per la distribuzione, utilizzare l'integrazione dei file S3. Per ulteriori informazioni, consulta Integrazione di un'istanza database Amazon RDS for SQL Server con Amazon S3.

Autorizzazioni amministrative su SSISDB

Quando l'istanza viene creata o modificata con l'opzione SSIS, il risultato è un database SSISDB con i ruoli ssis_admin e ssis_logreader concessi all'utente master. L'utente master dispone dei seguenti privilegi in SSISDB:

- modifica del ruolo ssis_admin
- modifica del ruolo ssis_logreader
- modifica di qualsiasi utente

Poiché l'utente principale è un utente autenticato SQL, non è possibile utilizzare l'utente master per l'esecuzione di pacchetti SSIS. L'utente master può utilizzare questi privilegi per creare nuovi utenti SSISDB e aggiungerli ai ruoli ssis_admin e ssis_logreader. Questa operazione è utile per dare accesso agli utenti del dominio per l'utilizzo di SSIS.

Configurazione di un utente autenticato da Windows per SSIS

L'utente master può utilizzare l'esempio di codice riportato di seguito per impostare un accesso autenticato da Windows in SSISDB e concedere le autorizzazioni necessarie per la procedura. In questo modo vengono concesse autorizzazioni all'utente del dominio per distribuire ed eseguire pacchetti SSIS, utilizzare procedure di trasferimento file S3, creare credenziali e lavorare con il proxy Agente SQL Server. Per ulteriori informazioni, consulta Credentials (Database Engine) e Create a SQL Server Agent Proxy nella documentazione di Microsoft.

> **Note**
>
> È possibile concedere alcune o tutte le autorizzazioni seguenti, se necessario, agli utenti autenticati da Windows.

Example

```
-- Create a server-level SQL login for the domain user, if it doesn't already exist
USE [master]
GO
CREATE LOGIN [mydomain\user_name] FROM WINDOWS
GO

-- Create a database-level account for the domain user, if it doesn't already exist

USE [SSISDB]
GO
CREATE USER [mydomain\user_name] FOR LOGIN [mydomain\user_name]

-- Add SSIS role membership to the domain user
ALTER ROLE [ssis_admin] ADD MEMBER [mydomain\user_name]
ALTER ROLE [ssis_logreader] ADD MEMBER [mydomain\user_name]
GO

-- Add MSDB role membership to the domain user
USE [msdb]
GO
CREATE USER [mydomain\user_name] FOR LOGIN [mydomain\user_name]

-- Grant MSDB stored procedure privileges to the domain user
GRANT EXEC ON msdb.dbo.rds_msbi_task TO [mydomain\user_name] with grant option
GRANT SELECT ON msdb.dbo.rds_fn_task_status TO [mydomain\user_name] with grant option
```

```
GRANT EXEC ON msdb.dbo.rds_task_status TO [mydomain\user_name] with grant option
GRANT EXEC ON msdb.dbo.rds_cancel_task TO [mydomain\user_name] with grant option
GRANT EXEC ON msdb.dbo.rds_download_from_s3 TO [mydomain\user_name] with grant option
GRANT EXEC ON msdb.dbo.rds_upload_to_s3 TO [mydomain\user_name] with grant option
GRANT EXEC ON msdb.dbo.rds_delete_from_filesystem TO [mydomain\user_name] with grant
 option
GRANT EXEC ON msdb.dbo.rds_gather_file_details TO [mydomain\user_name] with grant
 option
GRANT EXEC ON msdb.dbo.sp_add_proxy TO [mydomain\user_name] with grant option
GRANT EXEC ON msdb.dbo.sp_update_proxy TO [mydomain\user_name] with grant option
GRANT EXEC ON msdb.dbo.sp_grant_login_to_proxy TO [mydomain\user_name] with grant
 option
GRANT EXEC ON msdb.dbo.sp_revoke_login_from_proxy TO [mydomain\user_name] with grant
 option
GRANT EXEC ON msdb.dbo.sp_delete_proxy TO [mydomain\user_name] with grant option
GRANT EXEC ON msdb.dbo.sp_enum_login_for_proxy to [mydomain\user_name] with grant
 option
GRANT EXEC ON msdb.dbo.sp_enum_proxy_for_subsystem TO [mydomain\user_name]  with grant
 option
GRANT EXEC ON msdb.dbo.rds_sqlagent_proxy TO [mydomain\user_name] WITH GRANT OPTION


-- Add the SQLAgentUserRole privilege to the domain user
USE [msdb]
GO
ALTER ROLE [SQLAgentUserRole] ADD MEMBER [mydomain\user_name]
GO

-- Grant the ALTER ANY CREDENTIAL privilege to the domain user
USE [master]
GO
GRANT ALTER ANY CREDENTIAL TO [mydomain\user_name]
GO
```

Distribuzione di un progetto SSIS

In RDS, non è possibile distribuire progetti SSIS direttamente utilizzando procedure SQL Server Management Studio (SSMS) o SSIS. Per scaricare i file di progetto da Amazon S3 e quindi distribuirli, utilizzare le procedure archiviate di RDS.

Per eseguire le procedure archiviate, accedere come o qualsiasi utente a cui sono state concesse le autorizzazioni di esecuzione per le procedure archiviate. Per ulteriori informazioni, consulta Configurazione di un utente autenticato da Windows per SSIS.

Per distribuire il progetto SSIS

1. Scaricare il file di progetto (.ispac).

```
exec msdb.dbo.rds_download_from_s3
@s3_arn_of_file='arn:aws:s3:::bucket_name/ssisproject.ispac',
[@rds_file_path='D:\S3\ssisproject.ispac'],
[@overwrite_file=1];
```

2. Inviare l'attività di distribuzione, verificando quanto segue:

   - La cartella è presente nel catalogo SSIS.
   - Il nome del progetto corrisponde al nome del progetto utilizzato durante lo sviluppo del progetto SSIS.

```
exec msdb.dbo.rds_msbi_task
@task_type='SSIS_DEPLOY_PROJECT',
@folder_name='DEMO',
@project_name='ssisproject',
@file_path='D:\S3\ssisproject.ispac';
```

Monitoraggio dello stato di un'attività di distribuzione

Per tenere traccia dello stato dell'attività di distribuzione, chiamare la funzione `rds_fn_task_status`. accetta due parametri. Il primo parametro deve essere sempre `NULL` perché non si applica a SSIS. Il secondo parametro accetta un ID attività.

Per visualizzare l'elenco di tutte le attività, imposta il primo parametro su `NULL` e il secondo parametro su `0`, come indicato nell'esempio seguente.

```
SELECT * FROM msdb.dbo.rds_fn_task_status(NULL,0);
```

Per ottenere un'attività specifica, imposta il primo parametro su `NULL` e il secondo parametro sull'ID attività, come indicato nell'esempio seguente.

```
SELECT * FROM msdb.dbo.rds_fn_task_status(NULL,42);
```

La funzione `rds_fn_task_status` restituisce le seguenti informazioni.

Parametro di output	Descrizione
`task_id`	L'ID dell'attività.
`task_type`	`SSIS_DEPLOY_PROJECT`
`database_name`	Non applicabile alle attività SSIS.
`% complete`	L'avanzamento dell'attività espresso come percentuale.
`duration (mins)`	La quantità di tempo dedicato all'attività, in minuti.
`lifecycle`	Lo stato dell'attività. I possibili stati sono i seguenti: • `CREATED` – Dopo aver chiamato la procedura archiviata `msdb.dbo.rds_msbi_task` , viene creata un'attività e lo stato è impostato su `CREATED`. • `IN_PROGRESS` – Dopo l'avvio di un'attività, lo stato viene impostato su `IN_PROGRESS` . Possono essere necessari fino a 5 minuti perché lo stato cambi da `CREATED` in `IN_PROGRESS` . • `SUCCESS` – Al termine di un'attività, lo stato viene impostato su `SUCCESS`. • `ERROR` – Se un'attività non riesce, lo stato viene impostato su `ERROR`. Per ulteriori informazioni sull'errore, consulta la colonna `task_info` . •

Parametro di output	Descrizione
	`CANCEL_REQUESTED` – Quando chiami `rds_cancel_task` , lo stato dell'attività viene impostato su `CANCEL_REQUESTED` . • `CANCELLED` – Dopo che un'attività è stata annullata, lo stato dell'attività viene impostato su `CANCELLED` .
`task_info`	Ulteriori informazioni sull'attività. Se si verifica un errore durante l'elaborazione, questa colonna contiene informazioni sull'errore.
`last_updated`	La data e l'ora dell'ultimo aggiornamento dello stato dell'attività.
`created_at`	La data e l'ora di creazione dell'attività.
`S3_object_arn`	Non applicabile alle attività SSIS.
`overwrite_S3_backup_file`	Non applicabile alle attività SSIS.
`KMS_master_key_arn`	Non applicabile alle attività SSIS.
`filepath`	Non applicabile alle attività SSIS.
`overwrite_file`	Non applicabile alle attività SSIS.
`task_metadata`	Metadati associati all'attività SSIS.

Utilizzo di SSIS

Dopo aver distribuito il progetto SSIS nel catalogo SSIS, è possibile eseguire i pacchetti direttamente da SSMS o pianificarli utilizzando SQL Server Agent. È necessario utilizzare un accesso autenticato

da Windows per l'esecuzione di pacchetti SSIS. Per ulteriori informazioni, consulta Configurazione di un utente autenticato da Windows per SSIS.

Argomenti

- Impostazione di gestori di connessioni al database per i progetti SSIS
- Creazione di un proxy SSIS
- Pianificazione di un pacchetto SSIS utilizzando SQL Server Agent
- Revoca dell'accesso SSIS dal proxy

Impostazione di gestori di connessioni al database per i progetti SSIS

Quando si utilizza un gestore di connessione, è possibile utilizzare i seguenti tipi di autenticazione:

- Per le connessioni al database locale mediante Active Directory gestito da AWS, puoi utilizzare l'autenticazione SQL o l'autenticazione di Windows. Per l'autenticazione di Windows, utilizzare *DB_instance_name*.*fully_qualified_domain_name* come nome del server della stringa di connessione.

  Un esempio è `myssisinstance.corp-ad.example.com`, dove `myssisinstance` è il nome dell'istanza database ed `corp-ad.example.com` è il nome di dominio completo.
- Per le connessioni remote, utilizzare sempre l'autenticazione SQL.
- Per le connessioni al database locale mediante Active Directory gestito dal cliente, puoi utilizzare l'autenticazione SQL o l'autenticazione di Windows. Per l'autenticazione di Windows, utilizza `.` o *`LocalHost`* come nome del server della stringa di connessione.

Creazione di un proxy SSIS

Per pianificare pacchetti SSIS utilizzando SQL Server Agent, creare una credenziale SSIS e un proxy SSIS. Eseguire queste procedure come utente autenticato da Windows.

Per creare le credenziali SSIS

- Creare le credenziali per il proxy. A tale scopo, è possibile utilizzare SSMS o la seguente istruzione SQL.

```
USE [master]
GO
```

```
CREATE CREDENTIAL [SSIS_Credential] WITH IDENTITY = N'mydomain\user_name', SECRET =
 N'mysecret'
GO
```

> **Note**
>
> IDENTITY deve essere un accesso autenticato dal dominio. Sostituire *mysecret* con la password per l'accesso autenticato dal dominio.
> Ogni volta che l'host primario SSISDB viene modificato, modificare le credenziali proxy SSIS per consentire al nuovo host di accedervi.

Per creare il proxy SSIS

1. Utilizzare l'istruzione SQL seguente per creare il proxy.

```
USE [msdb]
GO
EXEC msdb.dbo.sp_add_proxy
 @proxy_name=N'SSIS_Proxy',@credential_name=N'SSIS_Credential',@description=N''
GO
```

2. Utilizzare l'istruzione SQL seguente per concedere l'accesso al proxy ad altri utenti.

```
USE [msdb]
GO
EXEC msdb.dbo.sp_grant_login_to_proxy
 @proxy_name=N'SSIS_Proxy',@login_name=N'mydomain\user_name'
GO
```

3. Utilizzare la seguente istruzione SQL per dare al sottosistema SSIS l'accesso al proxy.

```
USE [msdb]
GO
EXEC msdb.dbo.rds_sqlagent_proxy
 @task_type='GRANT_SUBSYSTEM_ACCESS',@proxy_name='SSIS_Proxy',@proxy_subsystem='SSIS'
GO
```

Per visualizzare il proxy e le concessioni sul proxy

1. Utilizzare l'istruzione SQL seguente per visualizzare gli assegnatari del proxy.

```
USE [msdb]
GO
EXEC sp_help_proxy
GO
```

2. Utilizzare l'istruzione SQL seguente per visualizzare i privilegi del sottosistema.

```
USE [msdb]
GO
EXEC msdb.dbo.sp_enum_proxy_for_subsystem
GO
```

Pianificazione di un pacchetto SSIS utilizzando SQL Server Agent

Dopo aver creato le credenziali e il proxy e dopo aver concesso l'accesso SSIS al proxy, è possibile creare un processo Agente SQL Server per pianificare il pacchetto SSIS.

Per pianificare il pacchetto SSIS

- È possibile utilizzare SSMS o T-SQL per creare il processo Agente SQL Server. L'esempio seguente utilizza T-SQL.

```
USE [msdb]
GO
DECLARE @jobId BINARY(16)
EXEC msdb.dbo.sp_add_job @job_name=N'MYSSISJob',
@enabled=1,
@notify_level_eventlog=0,
@notify_level_email=2,
@notify_level_page=2,
@delete_level=0,
@category_name=N'[Uncategorized (Local)]',
@job_id = @jobId OUTPUT
GO
EXEC msdb.dbo.sp_add_jobserver @job_name=N'MYSSISJob',@server_name=N'(local)'
GO
```

```
EXEC msdb.dbo.sp_add_jobstep
 @job_name=N'MYSSISJob',@step_name=N'ExecuteSSISPackage',
@step_id=1,
@cmdexec_success_code=0,
@on_success_action=1,
@on_fail_action=2,
@retry_attempts=0,
@retry_interval=0,
@os_run_priority=0,
@subsystem=N'SSIS',
@command=N'/ISSERVER "\"\SSISDB\MySSISFolder\MySSISProject\MySSISPackage.dtsx\"" /
SERVER "\"my-rds-ssis-instance.corp-ad.company.com/\""
/Par "\"$ServerOption::LOGGING_LEVEL(Int16)\"";1 /Par
 "\"$ServerOption::SYNCHRONIZED(Boolean)\"";True /CALLERINFO SQLAGENT /REPORTING
 E',
@database_name=N'master',
@flags=0,
@proxy_name=N'SSIS_Proxy'
GO
```

Revoca dell'accesso SSIS dal proxy

È possibile revocare l'accesso al sottosistema SSIS ed eliminare il proxy SSIS utilizzando le seguenti procedure memorizzate.

Per revocare l'accesso ed eliminare il proxy

1. Revocare l'accesso al sottosistema.

```
USE [msdb]
GO
EXEC msdb.dbo.rds_sqlagent_proxy
 @task_type='REVOKE_SUBSYSTEM_ACCESS',@proxy_name='SSIS_Proxy',@proxy_subsystem='SSIS'
GO
```

2. Revocare le concessioni per la delega.

```
USE [msdb]
GO
EXEC msdb.dbo.sp_revoke_login_from_proxy
 @proxy_name=N'SSIS_Proxy',@name=N'mydomain\user_name'
```

```
GO
```

3. Eliminare il proxy.

```
USE [msdb]
GO
EXEC dbo.sp_delete_proxy @proxy_name = N'SSIS_Proxy'
GO
```

Disabilitazione di SSIS

Per disabilitare SSIS, rimuovere l'opzione `SSIS` dal relativo gruppo di opzioni.

> **Important**
>
> La rimozione dell'opzione non elimina il database SSISDB, quindi è possibile rimuovere in modo sicuro l'opzione senza perdere i progetti SSIS.
> È possibile riattivare l'opzione `SSIS` dopo la rimozione per riutilizzare i progetti SSIS precedentemente distribuiti nel catalogo SSIS.

Console

La procedura seguente rimuove l'opzione `SSIS`.

Per rimuovere l'opzione SSIS dal suo gruppo di opzioni

1. Accedi alla AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel pannello di navigazione scegliere Option groups (Gruppi di opzioni).
3. Scegliere il gruppo di opzioni con l'opzione `SSIS` (`ssis-se-2016` negli esempi precedenti).
4. Scegliere Delete option (Elimina opzione).
5. In Opzioni di eliminazione, scegliere SSIS per Opzioni da eliminare.
6. In Apply immediately (Applica immediatamente), scegliere Yes (Sì) per eliminare immediatamente l'opzione oppure No per eliminarla nella finestra di manutenzione successiva.
7. Scegliere Delete (Elimina).

CLI

La procedura seguente rimuove l'opzione `SSIS`.

Per rimuovere l'opzione SSIS dal suo gruppo di opzioni

- Eseguire uno dei seguenti comandi.

  Example

  Per LinuxmacOS, oUnix:

```
aws rds remove-option-from-option-group \
    --option-group-name ssis-se-2016 \
    --options SSIS \
    --apply-immediately
```

  Per Windows:

```
aws rds remove-option-from-option-group ^
    --option-group-name ssis-se-2016 ^
    --options SSIS ^
    --apply-immediately
```

Eliminazione del database SSISDB

Dopo aver rimosso l'opzione SSIS, il database SSISDB non viene eliminato. Per eliminare il database SSISDB, utilizzare la procedura memorizzata `rds_drop_ssis_database` dopo aver rimosso l'opzione SSIS.

Per eliminare il database SSIS

- Utilizzare la seguente stored procedure.

```
USE [msdb]
GO
EXEC dbo.rds_drop_ssis_database
GO
```

Dopo aver eliminato il database SSISDB, se si riattiva l'opzione SSIS si ottiene un nuovo catalogo SSISDB.

Supporto per SQL Server Reporting Services in Amazon RDS for SQL Server

Microsoft SQL Server Reporting Services (SSRS) è un'applicazione basata su server utilizzata per la generazione e la distribuzione di report. Fa parte di una suite di servizi SQL Server che include anche SQL Server Analysis Services (SSAS) e SQL Server Integration Services (SSIS). SSRS è un servizio basato su SQL Server. Puoi utilizzarlo per raccogliere dati da varie origini dati e presentarli in un modo facilmente comprensibile e pronto per l'analisi.

Amazon RDS for SQL Server supporta l'esecuzione di SSRS direttamente su istanze database RDS. È possibile utilizzare SSRS con istanze database nuove o esistenti.

RDS supporta SSRS per SQL Server Standard ed Enterprise Edition nelle seguenti versioni:

- SQL Server 2022, tutte le versioni
- SQL Server 2019, versione 15.00.4043.16.v1 e successive
- SQL Server 2017, versione 14.00.3223.3.v1 e successive
- SQL Server 2016, versione 13.00.5820.21.v1 e successive

Indice

- Configurazione dell'origine dati del report
- Utilizzo di SSRS Email per inviare report
- Revoca delle autorizzazioni a livello di sistema
- Monitoraggio dello stato di un'attività
- Disattivazione di SSRS
- Eliminazione dei database SSRS

Limitazioni e consigli

Le seguenti limitazioni e i suggerimenti riportati di seguito si applicano all'esecuzione di SSRS su RDS per SQL Server:

- Non è possibile utilizzare SSRS su istanze database con repliche di lettura.
- Le istanze devono utilizzare Active Directory autogestito o AWS Directory Service for Microsoft Active Directory per l'autenticazione del portale Web SSRS e del server Web. Per ulteriori informazioni, consulta Utilizzo di Active Directory con RDS per SQL Server.
- Non è possibile eseguire il backup dei database dei server di reporting creati con l'opzione SSRS.
- L'importazione e il ripristino dei database del server di report da altre istanze di SSRS non sono operazioni supportate. Per ulteriori informazioni, consulta Database del server di report.
- Non è possibile configurare SSRS per l'ascolto sulla porta SSL predefinita (443). I valori consentiti sono 1150–49511, eccetto 1234, 1434, 3260, 3343, 3389 e 47001.
- Le sottoscrizioni tramite una condivisione di file di Microsoft Windows non sono supportate.
- L'utilizzo di Reporting Services Configuration Manager non è supportato.
- Le operazioni di creazione e modifica dei ruoli non sono supportate.
- La modifica delle proprietà del server di report non è supportata.
- I ruoli di amministratore di sistema e utente di sistema non sono concessi.
- Non è possibile modificare assegnazioni dei ruoli a livello di sistema tramite il portale Web.

Attivazione di SSRS

Utilizza il seguente processo per attivare SSRS per l'istanza database:

1. Creare un nuovo gruppo di opzioni oppure utilizzare un gruppo di opzioni esistente.

2. Aggiungere l'opzione SSRS al gruppo di opzioni.
3. Associare il gruppo di opzioni a questa istanza database.
4. Consentire l'accesso in ingresso al gruppo di sicurezza Virtual Private Cloud (VPC) per la porta listener SSRS.

Creazione di un gruppo di opzioni per SSRS

Per utilizzare SSRS, creare un gruppo di opzioni che corrisponde al motore SQL Server e alla versione dell'istanza database che si intende utilizzare. A tale scopo, utilizzare o. AWS Management Console AWS CLI

> **Note**
>
> È inoltre possibile utilizzare un gruppo di opzioni esistente se si tratta del motore e della versione di SQL Server corretti.

Console

La seguente procedura crea un gruppo di opzioni per SQL Server Standard Edition 2017.

Per creare il gruppo di opzioni

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel pannello di navigazione scegliere Option groups (Gruppi di opzioni).
3. Seleziona Create group (Crea gruppo).
4. Nella finestra Create option group (Crea gruppo di opzioni) eseguire queste operazioni:
   a. Per Nome, inserisci un nome per il gruppo di opzioni che sia unico all'interno del tuo gruppo Account AWS, ad esempio**`ssrs-se-2017`**. Il nome può includere solo lettere, cifre e trattini.
   b. Per Descrizione, immettere una breve descrizione del gruppo di opzioni, ad esempio **`SSRS option group for SQL Server SE 2017`**. La descrizione viene usata per la visualizzazione.
   c. Per Engine (Motore), scegliere sqlserver-se.
   d. Per Major engine version (Versione principale del motore), scegli 14.00.
5. Scegliere Create (Crea).

CLI

La seguente procedura crea un gruppo di opzioni per SQL Server Standard Edition 2017.

Per creare il gruppo di opzioni

- Eseguire uno dei seguenti comandi.

Example

Per LinuxmacOS, oUnix:

```
aws rds create-option-group \
    --option-group-name ssrs-se-2017 \
    --engine-name sqlserver-se \
    --major-engine-version 14.00 \
    --option-group-description "SSRS option group for SQL Server SE 2017"
```

Per Windows:

```
aws rds create-option-group ^
    --option-group-name ssrs-se-2017 ^
    --engine-name sqlserver-se ^
    --major-engine-version 14.00 ^
    --option-group-description "SSRS option group for SQL Server SE 2017"
```

Aggiunta dell'opzione SSRS al gruppo di opzioni

Quindi, usa AWS Management Console o AWS CLI per aggiungere l'SSRSopzione al tuo gruppo di opzioni.

Console

Per aggiungere l'opzione SSRS

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel pannello di navigazione scegliere Option groups (Gruppi di opzioni).
3. Seleziona il gruppo di opzioni precedentemente creato, quindi scegli Add Option (Aggiungi opzione).

4. In Dettagli opzione, scegliere SSRS per Nome opzione.
5. In Impostazioni delle opzioni, effettuare le seguenti operazioni:

   a. Immettere la porta su cui rimanere in ascolto del servizio SSRS. L'impostazione predefinita è 8443. Per un elenco dei valori consentiti, consulta Limitazioni e consigli.

   b. Immettere un valore per Memoria massima.

      L'opzione Memoria massima specifica la soglia superiore al di sopra della quale non vengono concesse nuove richieste di allocazione di memoria alle applicazioni del server di report. Il numero è una percentuale della memoria totale dell'istanza database. I valori consentiti sono compresi tra 10 e 80.

   c. Per Security groups (Gruppi di sicurezza), scegliere il gruppo di sicurezza VPC da associare all'opzione. Utilizzare lo stesso gruppo di sicurezza associato all'istanza database.

6. Per utilizzare SSRS Email per inviare report, scegli la casella di controllo Configure email delivery options (Configura opzioni di consegna e-mail) nell'area Email delivery in reporting services (Consegna e-mail in Reporting Services) e quindi esegui le seguenti operazioni:

   a. In Sender email address (Indirizzo e-mail mittente), immetti l'indirizzo e-mail da utilizzare nell campo From (Da) dei messaggi inviati da SSRS Email.

      Specifica un account utente con il permesso di inviare posta dal server SMTP.

   b. In SMTP server (Server SMTP), specifica il server o il gateway SMTP da utilizzare.

      Può essere un indirizzo IP, il nome NetBIOS di un computer sulla rete Intranet aziendale o un nome di dominio completo.

   c. In SMTP port (Porta SMTP), immetti la porta da utilizzare per connetterti al server di posta. Il valore predefinito è 25.

   d. Per utilizzare l'autenticazione:

      i. Seleziona la casella di controllo Use authentication (Utilizza autenticazione).

      ii. Per Secret Amazon Resource Name (ARN), inserisci l' AWS Secrets Manager ARN per le credenziali dell'utente.

         Utilizza il seguente formato:

         arn:aws:secretsmanager:*Region*:*AccountId*:secret:*SecretName*-*6RandomChara*

         Ad esempio:

arn:aws:secretsmanager:*us-west-2*:*123456789012*:secret:*MySecret-a1b2c3*

Per ulteriori informazioni sulla creazione del segreto, consulta Utilizzo di SSRS Email per inviare report.

e. Seleziona la casella di controllo Use Secure Sockets Layer (SSL) (Utilizza Secure Sockets Layer (SSL)) per crittografare i messaggi e-mail tramite SSL.

7. In Scheduling (Pianificazione), scegliere se aggiungere l'opzione immediatamente o alla finestra di manutenzione successiva.
8. Scegliere Add option (Aggiungi opzione).

CLI

Per aggiungere l'opzione SSRS

1. Crea un file JSON, ad esempio `ssrs-option.json`.

   a. Imposta i parametri obbligatori seguenti:

   - `OptionGroupName` – Il gruppo di opzioni creato o scelto in precedenza (`ssrs-se-2017` nell'esempio seguente).
   - `Port` – La porta per il servizio SSRS su cui rimanere in ascolto. L'impostazione predefinita è 8443. Per un elenco dei valori consentiti, consulta Limitazioni e consigli.
   - `VpcSecurityGroupMemberships` – Appartenenze ai gruppi di sicurezza VPC per l'istanza database RDS.
   - `MAX_MEMORY` – La soglia superiore al di sopra della quale non vengono concesse nuove richieste di allocazione di memoria alle applicazioni del server di report. Il numero è una percentuale della memoria totale dell'istanza database. I valori consentiti sono compresi tra 10 e 80.

   b. (Facoltativo) Imposta i seguenti parametri per utilizzare SSRS Email:

   - `SMTP_ENABLE_EMAIL`: imposta su `true` per utilizzare SSRS Email. Il valore predefinito è `false`.
   - `SMTP_SENDER_EMAIL_ADDRESS`: l'indirizzo e-mail da utilizzare nel campo From (Da) dei messaggi inviati da SSRS Email. Specifica un account utente con il permesso di inviare posta dal server SMTP.

- SMTP_SERVER: il server o il gateway SMTP da utilizzare. Può essere un indirizzo IP, il nome NetBIOS di un computer sulla rete Intranet aziendale o un nome di dominio completo.
- SMTP_PORT: la porta da utilizzare per la connessione al server di posta. Il valore predefinito è 25.
- SMTP_USE_SSL: imposta su `true` per crittografare i messaggi e-mail tramite SSL. Il valore predefinito è `true`.
- SMTP_EMAIL_CREDENTIALS_SECRET_ARN: l'ARN di Secrets Manager che contiene le credenziali dell'utente. Utilizza il seguente formato:

  arn:aws:secretsmanager:*Region*:*AccountId*:secret:*SecretName*-*6RandomCharact*

  Per ulteriori informazioni sulla creazione del segreto, consulta Utilizzo di SSRS Email per inviare report.
- SMTP_USE_ANONYMOUS_AUTHENTICATION: imposta su `true` e non includere SMTP_EMAIL_CREDENTIALS_SECRET_ARN se non desideri utilizzare l'autenticazione.

  Il valore predefinito è `false` quando SMTP_ENABLE_EMAIL è `true`.

L'esempio seguente include i parametri SSRS Email, utilizzando l'ARN segreto.

```
{
"OptionGroupName": "ssrs-se-2017",
"OptionsToInclude": [
 {
 "OptionName": "SSRS",
 "Port": 8443,
 "VpcSecurityGroupMemberships": ["sg-0abcdef123"],
 "OptionSettings": [
            {"Name": "MAX_MEMORY","Value": "60"},
            {"Name": "SMTP_ENABLE_EMAIL","Value": "true"}
            {"Name": "SMTP_SENDER_EMAIL_ADDRESS","Value": "nobody@example.com"},
            {"Name": "SMTP_SERVER","Value": "email-smtp.us-west-2.amazonaws.com"},
            {"Name": "SMTP_PORT","Value": "25"},
            {"Name": "SMTP_USE_SSL","Value": "true"},
            {"Name": "SMTP_EMAIL_CREDENTIALS_SECRET_ARN","Value":
 "arn:aws:secretsmanager:us-west-2:123456789012:secret:MySecret-a1b2c3"}
            ]
 }],
```

```
"ApplyImmediately": true
}
```

2. Aggiungere l'opzione SSRS al gruppo di opzioni.

   Example

   PerLinux, o: macOS Unix

```
aws rds add-option-to-option-group \
    --cli-input-json file://ssrs-option.json \
    --apply-immediately
```

   Per Windows:

```
aws rds add-option-to-option-group ^
    --cli-input-json file://ssrs-option.json ^
    --apply-immediately
```

Associazione del gruppo di opzioni all'istanza database

Usa AWS Management Console o AWS CLI per associare il tuo gruppo di opzioni all'istanza DB.

Se si utilizza un'istanza database esistente, un dominio Active Directory e un ruolo AWS Identity and Access Management (IAM) devono già essere associati all'istanza. Se si crea una nuova istanza, specificare un dominio Active Directory esistente e un ruolo IAM. Per ulteriori informazioni, consulta Utilizzo di Active Directory con RDS per SQL Server.

Console

È possibile associare il gruppo di opzioni a un'istanza database nuova o esistente:

- Per una nuova istanza database, associare il gruppo di opzioni quando viene avviata l'istanza. Per ulteriori informazioni, consulta Creazione di un'istanza database Amazon RDS.
- Per un'istanza database esistente, modificare l'istanza e associare il nuovo gruppo di opzioni. Per ulteriori informazioni, consulta Modifica di un'istanza database Amazon RDS.

CLI

È possibile associare il gruppo di opzioni a un'istanza database nuova o esistente.

Per creare un'istanza database che utilizza il gruppo di opzioni

- Specificare lo stesso tipo di motore del database e la versione principale utilizzati durante la creazione del gruppo di opzioni.

  Example

  Per LinuxmacOS, oUnix:

```
aws rds create-db-instance \
    --db-instance-identifier myssrsinstance \
    --db-instance-class db.m5.2xlarge \
    --engine sqlserver-se \
    --engine-version 14.00.3223.3.v1 \
    --allocated-storage 100 \
    --manage-master-user-password  \
    --master-username admin \
    --storage-type gp2 \
    --license-model li \
    --domain-iam-role-name my-directory-iam-role \
    --domain my-domain-id \
    --option-group-name ssrs-se-2017
```

  Per Windows:

```
aws rds create-db-instance ^
    --db-instance-identifier myssrsinstance ^
    --db-instance-class db.m5.2xlarge ^
    --engine sqlserver-se ^
    --engine-version 14.00.3223.3.v1 ^
    --allocated-storage 100 ^
    --manage-master-user-password ^
    --master-username admin ^
    --storage-type gp2 ^
    --license-model li ^
    --domain-iam-role-name my-directory-iam-role ^
    --domain my-domain-id ^
    --option-group-name ssrs-se-2017
```

Per modificare un'istanza database per utilizzare il gruppo di opzioni

- Eseguire uno dei seguenti comandi.

  Example

  Per LinuxmacOS, oUnix:

```
aws rds modify-db-instance \
    --db-instance-identifier myssrsinstance \
    --option-group-name ssrs-se-2017 \
    --apply-immediately
```

  Per Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier myssrsinstance ^
    --option-group-name ssrs-se-2017 ^
    --apply-immediately
```

Consentire l'accesso in ingresso al gruppo di sicurezza VPC

Per consentire l'accesso in ingresso al gruppo di sicurezza VPC associato all'istanza database, creare una regola in entrata per la porta listener SSRS specificata. Per ulteriori informazioni sulla configurazione di un gruppo di sicurezza, consulta Fornisci accesso alla istanza database nel VPC creando un gruppo di sicurezza.

Database del server di report

Quando l'istanza database è associata all'opzione SSRS, nell'istanza database vengono creati due nuovi database:

- `rdsadmin_ReportServer`
- `rdsadmin_ReportServerTempDB`

Questi database fungono da database ReportServer e ReportServerTemp DB. SSRS archivia i propri dati nel ReportServer database e memorizza nella cache i dati nel ReportServerTemp database DB. Per ulteriori informazioni, consulta Database del server di report nella documentazione Microsoft.

RDS possiede e gestisce questi database, pertanto le operazioni del database su di essi, come ALTER e DROP, non sono consentite. L'accesso al database `rdsadmin_ReportServerTempDB` non è consentito. Tuttavia, puoi eseguire operazioni di lettura sul database `rdsadmin_ReportServer`.

File di log SSRS

Puoi elencare, visualizzare e scaricare file di log SSRS. *I file di registro SSRS seguono una convenzione di denominazione di _ timestamp .log. ReportServerService* Questi log del server di report si trovano nella directory `D:\rdsdbdata\Log\SSRS`. La directory `D:\rdsdbdata\Log` è anche la directory principale dei log degli errori e dei log di SQL Server Agent. Per ulteriori informazioni, consulta Visualizzazione ed elenco dei file di log del database.

Per le istanze SSRS esistenti, potrebbe essere necessario riavviare il servizio SSRS per accedere ai log del server di report. È possibile riavviare il servizio aggiornando l'opzione `SSRS`.

Per ulteriori informazioni, consulta Utilizzo dei log di Microsoft SQL Server.

Accesso al portale Web SSRS

Utilizzare il seguente processo per accedere al portale Web SSRS:

1. Attiva Secure Sockets Layer (SSL).
2. Concedere l'accesso agli utenti del dominio.
3. Accedere al portale Web utilizzando un browser e le credenziali utente di dominio.

Utilizzo di SSL su RDS

SSRS utilizza il protocollo HTTPS/SSL per le sue connessioni. Per utilizzare questo protocollo, importare un certificato SSL nel sistema operativo Microsoft Windows sul computer client.

Per ulteriori informazioni sui certificati SSL, consulta . Per ulteriori informazioni sull'uso di SSL con SQL Server, consulta Utilizzo di SSL con un'istanza database Microsoft SQL Server.

Concessione dell'accesso agli utenti del dominio

In una nuova attivazione SSRS, non ci sono assegnazioni dei ruoli in SSRS. Per concedere a un utente del dominio o a un gruppo di utenti l'accesso al portale Web, RDS fornisce una stored procedure.

Per concedere l'accesso a un utente del dominio nel portale Web

- Utilizzare la seguente stored procedure.

```
exec msdb.dbo.rds_msbi_task
@task_type='SSRS_GRANT_PORTAL_PERMISSION',
@ssrs_group_or_username=N'AD_domain\user';
```

All'utente del dominio o al gruppo di utenti viene concesso il ruolo di sistema `RDS_SSRS_ROLE`. A questo ruolo sono concesse le seguenti attività a livello di sistema:

- Esecuzione di report
- Gestione dei processi
- Gestione di pianificazioni condivise
- Visualizzazione di pianificazioni condivise

Viene inoltre concesso il ruolo a livello di elemento di `Content Manager` nella cartella root.

Accesso al portale Web

Al termine dell'attività `SSRS_GRANT_PORTAL_PERMISSION`, è possibile accedere al portale utilizzando un browser Web. Il formato dell'URL del portale Web è il seguente.

```
https://rds_endpoint:port/Reports
```

In questo formato, è previsto quanto segue:

- *`rds_endpoint`* – L'endpoint per l'istanza database RDS utilizzata con SSRS.

  L'endpoint è disponibile nella scheda Connettività e sicurezza dell'istanza database. Per ulteriori informazioni, consulta Connessione a un'istanza database che esegua il motore di database di Microsoft SQL Server.
- *`port`* – La porta del listener per SSRS impostata nell'opzione `SSRS`.

Per accedere al portale Web

1. Immettere l'URL del portale Web nel browser.

```
https://myssrsinstance.cg034itsfake.us-east-1.rds.amazonaws.com:8443/Reports
```

2. Accedere con le credenziali per un utente del dominio a cui è stato concesso l'accesso con l'attività `SSRS_GRANT_PORTAL_PERMISSION`.

Distribuzione di report su SSRS

Dopo avere effettuato l'accesso al portale Web, è possibile distribuire report su di esso. È possibile utilizzare lo strumento di caricamento nel portale Web per caricare report o distribuire direttamente da SQL Server Data Tools (SSDT). Quando si esegue la distribuzione da SSDT, verificare quanto segue:

- L'utente che ha avviato SSDT ha accesso al portale Web SSRS.
- Il valore `TargetServerURL` nelle proprietà del progetto SSRS è impostato sull'endpoint HTTPS dell'istanza database RDS con suffisso `ReportServer`, ad esempio:

```
https://myssrsinstance.cg034itsfake.us-east-1.rds.amazonaws.com:8443/ReportServer
```

Configurazione dell'origine dati del report

Dopo aver distribuito un report su SSRS, occorre configurare l'origine dati del report. Durante la configurazione dell'origine dati del report, verifica quanto segue:

- Per le istanze DB di RDS per SQL Server unite a AWS Directory Service for Microsoft Active Directory, utilizza il nome di dominio completo (FQDN) come nome dell'origine dati della stringa di connessione. Un esempio è *`myssrsinstance.corp-ad.example.com`*, dove *`myssrsinstance`* è il nome dell'istanza database ed *`corp-ad.example.com`* è il nome di dominio completo.
- Per le istanze database RDS per SQL Server unite ad Active Directory autogestita, utilizza `.` o *`LocalHost`* come il nome dell'origine dati della stringa di connessione.

Utilizzo di SSRS Email per inviare report

SSRS include l'estensione SSRS Email, che è possibile utilizzare per inviare report agli utenti.

Per configurare SSRS Email, utilizza le impostazioni delle opzioni SSRS. Per ulteriori informazioni, consulta Aggiunta dell'opzione SSRS al gruppo di opzioni.

Dopo aver configurato SSRS Email, è possibile sottoscrivere i report sul server di report. Per ulteriori informazioni, consulta la pagina relativa alla consegna di e-mail in Reporting Services nella documentazione di Microsoft.

L'integrazione con AWS Secrets Manager è necessaria per il funzionamento di SSRS Email su RDS. Per l'integrazione con Secrets Manager, crea un segreto.

> **Note**
>
> Se modifichi il segreto in un secondo momento, devi anche aggiornare l'opzione SSRS nel gruppo di opzioni.

Per creare un segreto per SSRS Email

1. Segui la procedura riportata in Creazione di un segreto nella Guida per l'utente di AWS Secrets Manager .

   a. In Select secret type (Seleziona tipo di segreto), scegliere Other type of secrets (Altro tipo di segreti).
   b. In Key/value pairs (Coppia chiave/valore), immetti quanto segue:

      - **`SMTP_USERNAME`**: immetti un utente con il permesso di inviare posta dal server SMTP.
      - **`SMTP_PASSWORD`** immetti una password per l'utente SMTP.

   c. In Encryption key (Chiave crittografia), non utilizzare la AWS KMS key predefinita. Utilizza una chiave esistente o creane una nuova.

      La policy della chiave del KMS deve consentire l'operazione `kms:Decrypt`, ad esempio:

```
{
    "Sid": "Allow use of the key",
    "Effect": "Allow",
    "Principal": {
        "Service": [
            "rds.amazonaws.com"
        ]
    },
```

```
    "Action": [
        "kms:Decrypt"
    ],
    "Resource": "*"
}
```

2. Segui i passaggi contenuti nella pagina Allegare una policy di autorizzazione a un segreto nella Guida per l'utente di AWS Secrets Manager . La policy delle autorizzazioni fornisce l'operazione `secretsmanager:GetSecretValue` all principale del servizio `rds.amazonaws.com`.

   Si consiglia di utilizzare le condizione `aws:sourceAccount` e `aws:sourceArn` nella policy per evitare problemi di tipo confused deputy. Usa il tuo Account AWS for `aws:sourceAccount` e il gruppo di opzioni ARN per. `aws:sourceArn` Per ulteriori informazioni, consulta Prevenzione del problema "confused deputy" tra servizi.

   Di seguito viene illustrato un esempio di policy di autorizzazione.

```
{
  "Version" : "2012-10-17",
  "Statement" : [ {
    "Effect" : "Allow",
    "Principal" : {
      "Service" : "rds.amazonaws.com"
    },
    "Action" : "secretsmanager:GetSecretValue",
    "Resource" : "*",
    "Condition" : {
      "StringEquals" : {
        "aws:sourceAccount" : "123456789012"
      },
      "ArnLike" : {
        "aws:sourceArn" : "arn:aws:rds:us-west-2:123456789012:og:ssrs-se-2017"
      }
    }
  } ]
}
```

   Per altri esempi, consulta Esempi di policy sulle autorizzazioni per AWS Secrets Manager nella Guida per l'AWS Secrets Manager utente.

Revoca delle autorizzazioni a livello di sistema

Il ruolo di sistema `RDS_SSRS_ROLE` non dispone di autorizzazioni sufficienti per eliminare le assegnazioni di ruolo a livello di sistema. Per rimuovere un utente o un gruppo di utenti da `RDS_SSRS_ROLE`, utilizzare la stessa stored procedure utilizzata per concedere il ruolo, ma utilizzare il tipo di attività `SSRS_REVOKE_PORTAL_PERMISSION`.

Per revocare l'accesso da un utente del dominio per il portale Web

- Utilizzare la seguente stored procedure.

```
exec msdb.dbo.rds_msbi_task
@task_type='SSRS_REVOKE_PORTAL_PERMISSION',
@ssrs_group_or_username=N'AD_domain\user';
```

In questo modo l'utente viene eliminato dal ruolo di sistema `RDS_SSRS_ROLE`. Inoltre, l'utente viene eliminato dal ruolo a livello di elemento `Content Manager`, se uno esiste per l'utente.

Monitoraggio dello stato di un'attività

Per tenere traccia dello stato dell'attività di concessione o revoca, chiamare la funzione `rds_fn_task_status` che accetta due parametri. Il primo parametro deve essere sempre `NULL` perché non si applica a SSRS. Il secondo parametro accetta un ID attività.

Per visualizzare l'elenco di tutte le attività, imposta il primo parametro su `NULL` e il secondo parametro su `0`, come indicato nell'esempio seguente.

```
SELECT * FROM msdb.dbo.rds_fn_task_status(NULL,0);
```

Per ottenere un'attività specifica, imposta il primo parametro su `NULL` e il secondo parametro sull'ID attività, come indicato nell'esempio seguente.

```
SELECT * FROM msdb.dbo.rds_fn_task_status(NULL,42);
```

La funzione `rds_fn_task_status` restituisce le seguenti informazioni.

Parametro di output	Descrizione
`task_id`	L'ID dell'attività.

Parametro di output	Descrizione
`task_type`	Per SSRS, le attività possono avere i seguenti tipi di attività: • `SSRS_GRANT_PORTAL_PERMISSION` • `SSRS_REVOKE_PORTAL_PERMISSION`
`database_name`	Non applicabile alle attività SSRS.
`% complete`	L'avanzamento dell'attività espresso come percentuale.
`duration (mins)`	La quantità di tempo dedicato all'attività, in minuti.

Parametro di output	Descrizione
`lifecycle`	Lo stato dell'attività. I possibili stati sono i seguenti: • `CREATED` – Dopo aver chiamato una delle stored procedure SSRS, viene creata un'attività e lo stato viene impostato su `CREATED`. • `IN_PROGRESS` – Dopo l'avvio di un'attività, lo stato viene impostato su `IN_PROGRESS`. Possono essere necessari fino a 5 minuti perché lo stato cambi da `CREATED` in `IN_PROGRESS`. • `SUCCESS` – Al termine di un'attività, lo stato viene impostato su `SUCCESS`. • `ERROR` – Se un'attività non riesce, lo stato viene impostato su `ERROR`. Per ulteriori informazioni sull'errore, consulta la colonna `task_info`. • `CANCEL_REQUESTED` – Dopo aver chiamato la stored procedure `rds_cancel_task`, lo stato dell'attività viene impostato su `CANCEL_REQUESTED`. • `CANCELLED` – Dopo che un'attività è stata annullata, lo stato dell'attività viene impostato su `CANCELLED`.
`task_info`	Ulteriori informazioni sull'attività. Se si verifica un errore durante l'elaborazione, questa colonna contiene informazioni sull'errore.

Parametro di output	Descrizione
`last_updated`	La data e l'ora dell'ultimo aggiornamento dello stato dell'attività.
`created_at`	La data e l'ora di creazione dell'attività.
`S3_object_arn`	Non applicabile alle attività SSRS.
`overwrite_S3_backup_file`	Non applicabile alle attività SSRS.
`KMS_master_key_arn`	Non applicabile alle attività SSRS.
`filepath`	Non applicabile alle attività SSRS.
`overwrite_file`	Non applicabile alle attività SSRS.
`task_metadata`	Metadati associati all'attività SSRS.

Disattivazione di SSRS

Per disattivare SSRS, rimuovi l'opzione `SSRS` dal relativo gruppo di opzioni. La rimozione dell'opzione non elimina i database SSRS. Per ulteriori informazioni, consulta Eliminazione dei database SSRS.

È possibile riattivare SSRS aggiungendo nuovamente l'opzione `SSRS`. Se sono stati eliminati anche i database SSRS, la nuova aggiunta dell'opzione sulla stessa istanza database crea nuovi database del server di report.

Console

Per rimuovere l'opzione SSRS dal suo gruppo di opzioni

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel pannello di navigazione scegliere Option groups (Gruppi di opzioni).
3. Scegliere il gruppo di opzioni con l'opzione `SSRS` (`ssrs-se-2017` negli esempi precedenti).
4. Scegliere Delete option (Elimina opzione).
5. In Opzioni di eliminazione, scegliere SSRS per Opzioni da eliminare.

6. In Apply immediately (Applica immediatamente), scegliere Yes (Sì) per eliminare immediatamente l'opzione oppure No per eliminarla nella finestra di manutenzione successiva.
7. Scegliere Delete (Elimina).

CLI

Per rimuovere l'opzione SSRS dal suo gruppo di opzioni

- Eseguire uno dei seguenti comandi.

  Example

  Per LinuxmacOS, oUnix:

```
aws rds remove-option-from-option-group \
    --option-group-name ssrs-se-2017 \
    --options SSRS \
    --apply-immediately
```

  Per Windows:

```
aws rds remove-option-from-option-group ^
    --option-group-name ssrs-se-2017 ^
    --options SSRS ^
    --apply-immediately
```

Eliminazione dei database SSRS

La rimozione dell'opzione SSRS non elimina i database del server di report. Per eliminarli, utilizzare la stored procedure seguente.

Per eliminare i database del server di report, assicurarsi di rimuovere prima l'opzione SSRS.

Per eliminare i database SSRS

- Utilizzare la seguente stored procedure.

```
exec msdb.dbo.rds_drop_ssrs_databases
```

Supporto per Microsoft Distributed Transaction Coordinator in RDS per SQL Server

Una transazione distribuita è una transazione di database in cui sono coinvolti due o più host di rete. RDS per SQL Server supporta transazioni distribuite tra host, in cui un singolo host può essere uno dei seguenti:

- Istanza database RDS per SQL Server
- Host SQL Server locale
- Host Amazon EC2 con SQL Server installato
- Qualsiasi altro host EC2 o istanza database RDS con un motore del database che supporta le transazioni distribuite

In RDS, a partire da SQL Server 2012 (versione 11.00.5058.0.v1 e successive), tutte le edizioni di RDS per SQL Server supportano le transazioni distribuite. Il supporto viene fornito utilizzando Microsoft Distributed Transaction Coordinator (MSDTC). Per informazioni dettagliate su MSDTC, consulta Distributed Transaction Coordinator nella documentazione Microsoft.

Indice

Limitazioni

Le seguenti limitazioni si applicano all'utilizzo di MSDTC su RDS per SQL Server:

- MSDTC non è supportato nelle istanze che utilizzano il mirroring del database SQL Server. Per ulteriori informazioni, consulta Transactions - availability groups and database mirroring.
- Il parametro `in-doubt xact resolution` deve essere impostato su 1 o 2. Per ulteriori informazioni, consulta Modifica del parametro per MSDTC.
- MSDTC richiede che tutti i nomi host che partecipano alle transazioni distribuite siano risolvibili utilizzando i nomi host. RDS mantiene automaticamente questa funzionalità per le istanze aggiunte al dominio. Tuttavia, per le istanze standalone assicurarsi di configurare manualmente il server DNS.
- le transazioni XA di Java Database Connectivity (JDBC) sono supportate per SQL Server 2017 versione 14.00.3223.3 e successive e SQL Server 2019.
- Le transazioni distribuite che dipendono dalle DLL client nelle istanze RDS non sono supportate.
- L'utilizzo di librerie a collegamento dinamico XA personalizzate non è supportato.

Abilitazione di MSDTC

Utilizzare il seguente processo per abilitare MSDTC per l'istanza database:

1. Creare un nuovo gruppo di opzioni oppure utilizzare un gruppo di opzioni esistente.
2. Aggiungere l'opzione `MSDTC` al gruppo di opzioni.
3. Creare un nuovo gruppo di parametri o scegliere un gruppo di parametri esistente.
4. Modificare il gruppo di parametri per impostare il parametro `in-doubt xact resolution` su 1 o 2.
5. Associare il gruppo di opzioni e il gruppo di parametri all'istanza database.

Creazione del gruppo di opzioni per MSDTC

Utilizzare la AWS Management Console o AWS CLI per creare un gruppo di opzioni che corrisponde al motore SQL Server e alla versione dell'istanza database.

> **Note**
>
> È inoltre possibile utilizzare un gruppo di opzioni esistente se si tratta del motore e della versione di SQL Server corretti.

Console

La seguente procedura crea un gruppo di opzioni per SQL Server Standard Edition 2016.

Per creare il gruppo di opzioni

1. Accedi alla AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel pannello di navigazione scegliere Option groups (Gruppi di opzioni).
3. Seleziona Create group (Crea gruppo).
4. Nella finestra Create option group (Crea gruppo di opzioni) eseguire queste operazioni:
   a. Per Nome, immettere un nome per il gruppo di opzioni che sia univoco all'interno dell'account AWS, ad esempio **`msdtc-se-2016`**. Il nome può includere solo lettere, cifre e trattini.
   b. Per Descrizione, immettere una breve descrizione del gruppo di opzioni, ad esempio **`MSDTC option group for SQL Server SE 2016`**. La descrizione viene usata per la visualizzazione.
   c. Per Engine (Motore), scegliere sqlserver-se.
   d. Per Versione del motore principale, scegliere 13.00.
5. Scegliere Create (Crea).

CLI

Nell'esempio seguente viene creato un gruppo di opzioni per SQL Server Standard Edition 2016.

Per creare il gruppo di opzioni

- Utilizzare uno dei seguenti comandi.

Example

Per LinuxmacOS, oUnix:

```
aws rds create-option-group \
    --option-group-name msdtc-se-2016 \
    --engine-name sqlserver-se \
    --major-engine-version 13.00 \
    --option-group-description "MSDTC option group for SQL Server SE 2016"
```

Per Windows:

```
aws rds create-option-group ^
    --option-group-name msdtc-se-2016 ^
    --engine-name sqlserver-se ^
    --major-engine-version 13.00 ^
    --option-group-description "MSDTC option group for SQL Server SE 2016"
```

Aggiunta dell'opzione MSDTC al gruppo di opzioni

Utilizzare la AWS Management Console o l'AWS CLI per aggiungere l'opzione `MSDTC` al gruppo di opzioni.

Sono richieste le seguenti impostazioni delle opzioni:

- Porta – La porta utilizzata per accedere a MSDTC. I valori consentiti sono compresi tra 1150 e 49151, ad eccezione di 1234, 1434, 3260, 3343, 3389 e 47001. Il valore predefinito è 5000.

  Assicurarsi che la porta che si desidera utilizzare sia abilitata nelle regole del firewall. Assicurarsi, inoltre, che, se necessario, questa porta sia abilitata nelle regole in entrata e in uscita per il gruppo di sicurezza associato all'istanza database. Per ulteriori informazioni, consulta Impossibile connettersi all'istanza database di Amazon RDS.
- Security groups (Gruppi di sicurezza): le appartenenze a gruppi di sicurezza VPC per l'istanza database RDS.
- Tipo di autenticazione – La modalità di autenticazione tra gli host. Sono supportati i seguenti tipi di autenticazione:

  - Reciproco – Le istanze RDS vengono autenticate reciprocamente l'una con l'altra utilizzando l'autenticazione integrata. Se questa opzione è selezionata, tutte le istanze associate a questo gruppo di opzioni devono essere aggiunte al dominio.
  - Nessuna – Nessuna autenticazione viene eseguita tra gli host. Non è consigliabile utilizzare questa modalità in ambienti di produzione.
- Dimensione del log delle transazioni – La dimensione del log delle transazioni MSDTC. I valori consentiti sono compresi tra 4 e 1024 MB. La dimensione predefinita è 4 MB.

Le seguenti impostazioni delle opzioni sono facoltative:

- Abilitazione delle connessioni in entrata – Indica se consentire connessioni MSDTC in entrata alle istanze associate a questo gruppo di opzioni.
- Abilitazione delle connessioni in uscita – Indica se consentire connessioni MSDTC in uscita dalle istanze associate a questo gruppo di opzioni.
- Abilita XA – Indica se consentire transazioni XA. Per ulteriori informazioni sul protocollo XA, consulta XA Specification.
- Abilita LU SNA – Indica se consentire l'utilizzo del protocollo LU SNA per le transazioni distribuite. Per ulteriori informazioni sul supporto del protocollo LU SNA, consulta Managing IBM CICS LU 6.2 Transactions nella documentazione Microsoft.

Console

Per aggiungere l'opzione MSDTC

1. Accedi alla AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel pannello di navigazione scegliere Option groups (Gruppi di opzioni).
3. Scegliere il gruppo di opzioni appena creato.
4. Scegliere Add option (Aggiungi opzione).
5. In Dettagli opzione, scegliere MSDTC per Nome opzione.
6. In Impostazioni delle opzioni:
   a. Per Porta, immettere il numero di porta per accedere a MSDTC. L'impostazione predefinita è 5000.

b. Per Security groups (Gruppi di sicurezza), scegliere il gruppo di sicurezza VPC da associare all'opzione.

c. Per Tipo di autenticazione, scegliere Reciproco o Nessuna.

d. Per Dimensioni del log delle transazioni, immettere un valore compreso tra 4 e 1024. Il valore di default è 4.

7. In Configurazione aggiuntiva, eseguire le operazioni seguenti:

a. Per Connessioni, se necessario, scegliere Abilitazione delle connessioni in entrata e Abilitazione delle connessioni in uscita.

b. Per Protocolli consentiti, se necessario, scegliere Abilita XA e Abilita LU SNA.

8. In Scheduling (Pianificazione), scegliere se aggiungere l'opzione immediatamente o alla finestra di manutenzione successiva.

9. Scegliere Add option (Aggiungi opzione).

Per aggiungere questa opzione, non è richiesto alcun riavvio.

CLI

Per aggiungere l'opzione MSDTC

1. Creare un file JSON, ad esempio `msdtc-option.json`, con i seguenti parametri obbligatori.

```
{
"OptionGroupName":"msdtc-se-2016",
"OptionsToInclude": [
 {
 "OptionName":"MSDTC",
 "Port":5000,
 "VpcSecurityGroupMemberships":["sg-0abcdef123"],
 "OptionSettings":[{"Name":"AUTHENTICATION","Value":"MUTUAL"},
{"Name":"TRANSACTION_LOG_SIZE","Value":"4"}]
 }],
"ApplyImmediately": true
}
```

2. Aggiungere l'opzione `MSDTC` al gruppo di opzioni.

Example

Per LinuxmacOS, oUnix:

```
aws rds add-option-to-option-group \
    --cli-input-json file://msdtc-option.json \
    --apply-immediately
```

Per Windows:

```
aws rds add-option-to-option-group ^
    --cli-input-json file://msdtc-option.json ^
    --apply-immediately
```

Non è richiesto alcun riavvio.

Creazione del gruppo di parametri per MSDTC

Creare o modificare un gruppo di parametri per il parametro `in-doubt xact resolution` corrispondente all'edizione di SQL Server e alla versione dell'istanza database.

Console

Nell'esempio seguente viene creato un gruppo di parametri per SQL Server Standard Edition 2016.

Per creare il gruppo di parametri

1. Accedi alla AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel pannello di navigazione, scegli Parameter groups (Gruppi di parametri).
3. Scegliere Create parameter group (Crea gruppo di parametri).
4. Nel riquadro Create parameter group (Crea gruppi di parametri), procedi nel modo seguente:
    a. Per Famiglia del gruppo di parametri, scegliere sqlserver-se-13.0.
    b. Per Group name (Nome gruppo), immettere un identificatore per il gruppo di parametri, ad esempio **`msdtc-sqlserver-se-13`**.
    c. Per Description (Descrizione), immettere **`in-doubt xact resolution`**.
5. Scegliere Create (Crea).

CLI

Nell'esempio seguente viene creato un gruppo di parametri per SQL Server Standard Edition 2016.

Per creare il gruppo di parametri

- Utilizzare uno dei seguenti comandi.

  Example

  Per LinuxmacOS, oUnix:

```
aws rds create-db-parameter-group \
    --db-parameter-group-name msdtc-sqlserver-se-13 \
    --db-parameter-group-family "sqlserver-se-13.0" \
    --description "in-doubt xact resolution"
```

  Per Windows:

```
aws rds create-db-parameter-group ^
    --db-parameter-group-name msdtc-sqlserver-se-13 ^
    --db-parameter-group-family "sqlserver-se-13.0" ^
    --description "in-doubt xact resolution"
```

Modifica del parametro per MSDTC

Modifica il parametro `in-doubt xact resolution` nel gruppo di parametri che corrisponde all'edizione di SQL Server e alla versione dell'istanza database.

Per MSDTC, impostare il parametro `in-doubt xact resolution` su una delle seguenti opzioni:

- 1 - `Presume commit`. Si ipotizza che sia stato eseguito il commit di tutte le transazioni dubbie MSDTC.
- 2 - `Presume abort`. Si ipotizza che tutte le transazioni dubbie MSDTC siano state interrotte.

Per ulteriori informazioni, consulta in-doubt xact resolution Server Configuration Option nella documentazione Microsoft.

Console

Nell'esempio seguente il gruppo di parametri creato per SQL Server Standard Edition 2016 viene modificato.

Per modificare il gruppo di parametri

1. Accedi alla AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione scegliere Parameter groups (Gruppi di parametri).
3. Scegliete il gruppo di parametri, ad esempio msdtc-sqlserver-se-13.
4. In Parameters (Parametri), filtrare l'elenco dei parametri per **`xact`**.
5. Scegliere in-doubt xact resolution.
6. Scegliere Edit parameters (Modifica parametri).
7. Immetti **1** o **2**.
8. Seleziona Save changes (Salva modifiche).

CLI

Nell'esempio seguente il gruppo di parametri creato per SQL Server Standard Edition 2016 viene modificato.

Per modificare il gruppo di parametri

- Utilizzare uno dei seguenti comandi.

  Example

  Per LinuxmacOS, oUnix:

```
aws rds modify-db-parameter-group \
    --db-parameter-group-name msdtc-sqlserver-se-13 \
    --parameters "ParameterName='in-doubt xact
 resolution',ParameterValue=1,ApplyMethod=immediate"
```

  Per Windows:

```
aws rds modify-db-parameter-group ^
```

```
    --db-parameter-group-name msdtc-sqlserver-se-13 ^
    --parameters "ParameterName='in-doubt xact
resolution',ParameterValue=1,ApplyMethod=immediate"
```

Associazione del gruppo di opzioni e del gruppo di parametri all'istanza database

Puoi utilizzare la AWS Management Console o AWS CLI per associare il gruppo di opzioni MSDTC e il gruppo di parametri all'istanza database.

Console

Puoi associare il gruppo di opzioni MSDTC e il gruppo di parametri a un'istanza database nuova o esistente.

- Per una nuova istanza database, associarli all'avvio dell'istanza. Per ulteriori informazioni, consulta Creazione di un'istanza database Amazon RDS.
- Per un'istanza database esistente, associarli modificando l'istanza. Per ulteriori informazioni, consulta Modifica di un'istanza database Amazon RDS.

> **Note**
>
> Se utilizzi un'istanza database esistente aggiunta la dominio, ad essa devono già essere associati un dominio Active Directory e un ruolo AWS Identity and Access Management (IAM). Se crei una nuova istanza aggiunta la dominio, specifica un dominio Active Directory e un ruolo IAM esistenti. Per ulteriori informazioni, consulta Utilizzo di Active Directory gestito da AWS con RDS per SQL Server.

CLI

Puoi associare il gruppo di opzioni MSDTC e il gruppo di parametri a un'istanza database nuova o esistente.

> **Note**
>
> Se utilizzi un'istanza database aggiunta la dominio esistente, ad essa devono già essere associati un dominio Active Directory e un ruolo IAM. Se crei una nuova istanza aggiunta

la dominio, specifica un dominio Active Directory e un ruolo IAM esistenti. Per ulteriori informazioni, consulta Utilizzo di Active Directory gestito da AWS con RDS per SQL Server.

Per creare un'istanza database con il gruppo di opzioni MSDTC e il gruppo di parametri

- Specificare lo stesso tipo di motore del database e la versione principale utilizzati durante la creazione del gruppo di opzioni.

  Example

  Per LinuxmacOS, oUnix:

```
aws rds create-db-instance \
    --db-instance-identifier mydbinstance \
    --db-instance-class db.m5.2xlarge \
    --engine sqlserver-se \
    --engine-version 13.00.5426.0.v1 \
    --allocated-storage 100 \
    --manage-master-user-password \
    --master-username admin \
    --storage-type gp2 \
    --license-model li \
    --domain-iam-role-name my-directory-iam-role \
    --domain my-domain-id \
    --option-group-name msdtc-se-2016 \
    --db-parameter-group-name msdtc-sqlserver-se-13
```

  Per Windows:

```
aws rds create-db-instance ^
    --db-instance-identifier mydbinstance ^
    --db-instance-class db.m5.2xlarge ^
    --engine sqlserver-se ^
    --engine-version 13.00.5426.0.v1 ^
    --allocated-storage 100 ^
    --manage-master-user-password ^
    --master-username admin ^
    --storage-type gp2 ^
    --license-model li ^
    --domain-iam-role-name my-directory-iam-role ^
    --domain my-domain-id ^
```

```
    --option-group-name msdtc-se-2016 ^
    --db-parameter-group-name msdtc-sqlserver-se-13
```

Per modificare un'istanza database e associare il gruppo di opzioni MSDTC e il gruppo di parametri

- Utilizzare uno dei seguenti comandi.

  Example

  Per LinuxmacOS, oUnix:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --option-group-name msdtc-se-2016 \
    --db-parameter-group-name msdtc-sqlserver-se-13 \
    --apply-immediately
```

  Per Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --option-group-name msdtc-se-2016 ^
    --db-parameter-group-name msdtc-sqlserver-se-13 ^
    --apply-immediately
```

Utilizzo di transazioni distribuite

In Amazon RDS for SQL Server, le transazioni distribuite vengono eseguite allo stesso modo delle transazioni distribuite eseguite in locale:

- Utilizzando le transazioni promuovibili `System.Transactions` di .NET Framework, che ottimizza le transazioni distribuite posticipandone la creazione fino a quando non sono necessarie.

  In questo caso, la promozione è automatica e non richiede alcun intervento. Se all'interno della transazione è presente un solo gestore risorse, non viene eseguita alcuna promozione. Per ulteriori informazioni sugli ambiti di transazioni implicite, consulta Implementing an Implicit Transaction using Transaction Scope nella documentazione Microsoft.

  Le transazioni promuovibili sono supportate con queste implementazioni .NET:

- A partire da ADO.NET 2.0, `System.Data.SqlClient` supporta le transazioni promuovibili con SQL Server. Per ulteriori informazioni, consulta System.Transactions Integration with SQL Server nella documentazione Microsoft.
- ODP.NET supporta `System.Transactions`. Viene creata una transazione locale per la prima connessione nell'ambito `TransactionsScope` aperta a Oracle Database 11g release 1 (versione 11.1) e successive. Quando viene aperta una seconda connessione, questa transazione viene automaticamente promossa a una transazione distribuita. Per ulteriori informazioni sul supporto di transazioni distribuite in ODP.NET, consulta Microsoft Distributed Transaction Coordinator Integration nella documentazione Microsoft.

- Utilizzando l'istruzione `BEGIN DISTRIBUTED TRANSACTION`. Per ulteriori informazioni, consulta BEGIN DISTRIBUTED TRANSACTION (Transact-SQL) nella documentazione Microsoft.

Utilizzo di transazioni XA

A partire da RDS per SQL Server 2017 versione 14.00.3223.3, è possibile controllare transazioni distribuite utilizzando JDBC. Quando si imposta l'opzione `Enable XA` su `true` nell'opzione `MSDTC`, RDS abilita automaticamente le transazioni JDBC e concede il ruolo `SqlJDBCXAUser` all'utente `guest`. Ciò consente di eseguire transazioni distribuite tramite JDBC. Per ulteriori informazioni, tra cui un codice di esempio, consulta Comprendere le transazioni XA nella documentazione di Microsoft.

Utilizzo del tracciamento delle transazioni

RDS supporta il controllo delle tracce delle transazioni MSDTC e il loro download dall'istanza database RDS per la risoluzione dei problemi. Puoi controllare le sessioni di tracciamento delle transazioni eseguendo la seguente stored procedure di RDS.

```
exec msdb.dbo.rds_msdtc_transaction_tracing 'trace_action',
[@traceall='0|1'],
[@traceaborted='0|1'],
[@tracelong='0|1'];
```

Il parametro seguente è obbligatorio:

- `trace_action` – L'operazione di tracciamento. Può essere `START`, `STOP` o `STATUS`.

I parametri seguenti sono facoltativi:

- `@traceall` – Impostare su 1 per tracciare tutte le transazioni distribuite. Il valore predefinito è 0.
- `@traceaborted` – Impostare su 1 per tracciare le transazioni distribuite annullate. Il valore predefinito è 0.
- `@tracelong` – Impostare su 1 per tracciare le transazioni distribuite di lunga durata. Il valore predefinito è 0.

Example dell'operazione di tracciamento START

Per avviare una nuova sessione di tracciamento delle transazioni, eseguire l'istruzione di esempio seguente.

```
exec msdb.dbo.rds_msdtc_transaction_tracing 'START',
@traceall='0',
@traceaborted='1',
@tracelong='1';
```

> **Note**
>
> Può essere attiva una sola sessione di tracciamento delle transazioni alla volta. Se viene emesso un nuovo comando START di sessione di tracciamento mentre una sessione di tracciamento è attiva, viene restituito un errore e la sessione di tracciamento attiva rimane invariata.

Example di operazione di tracciamento STOP

Per interrompere una sessione di tracciamento delle transazioni, eseguire l'istruzione seguente.

```
exec msdb.dbo.rds_msdtc_transaction_tracing 'STOP'
```

Questa istruzione interrompe la sessione di tracciamento delle transazioni attiva e salva i dati di tracciamento delle transazioni nella directory di log nell'istanza database RDS. La prima riga dell'output contiene il risultato complessivo dell'esecuzione e le righe seguenti indicano i dettagli dell'operazione.

Di seguito è riportato un esempio di interruzione della sessione di tracciamento.

```
OK: Trace session has been successfully stopped.
```

```
Setting log file to: D:\rdsdbdata\MSDTC\Trace\dtctrace.log
Examining D:\rdsdbdata\MSDTC\Trace\msdtctr.mof for message formats,  8 found.
Searching for TMF files on path: (null)
Logfile D:\rdsdbdata\MSDTC\Trace\dtctrace.log:
 OS version    10.0.14393  (Currently running on 6.2.9200)
 Start Time    <timestamp>
 End Time      <timestamp>
 Timezone is   @tzres.dll,-932 (Bias is 0mins)
 BufferSize            16384 B
 Maximum File Size     10 MB
 Buffers  Written      Not set (Logger may not have been stopped).
 Logger Mode Settings (11000002) ( circular paged
 ProcessorCount         1
Processing completed   Buffers: 1, Events: 3, EventsLost: 0 :: Format Errors: 0,
 Unknowns: 3
Event traces dumped to d:\rdsdbdata\Log\msdtc_<timestamp>.log
```

Puoi utilizzare le informazioni dettagliate per eseguire query sul nome del file di log generato. Per ulteriori informazioni sul download dei file di log dall'istanza database RDS, consulta Monitoraggio dei file di log di Amazon RDS.

I log delle sessioni di traccia rimangono sull'istanza per 35 giorni. Tutti i log delle sessioni di traccia più vecchi vengono eliminati automaticamente.

Example dell'operazione di tracciamento STATUS

Per tracciare lo stato di una sessione di tracciamento delle transazioni, eseguire l'istruzione seguente.

```
exec msdb.dbo.rds_msdtc_transaction_tracing 'STATUS'
```

Questa istruzione restituisce quanto segue come righe separate del set di risultati.

```
OK
SessionStatus: <Started|Stopped>
TraceAll: <True|False>
TraceAborted: <True|False>
TraceLongLived: <True|False>
```

La prima riga indica il risultato complessivo dell'operazione: `OK` o `ERROR` con i dettagli, se applicabile. Le righe successive indicano i dettagli sullo stato della sessione di tracciamento:

- `SessionStatus`Il valore di può essere uno dei seguenti:

- `Started` se una sessione di tracciamento è in esecuzione.
- `Stopped` se nessuna sessione di tracciamento è in esecuzione.

- I flag della sessione di tracciamento possono essere `True` o `False` a seconda di come sono stati impostati nel comando `START`.

Modifica dell'opzione MSDTC

Dopo aver abilitato l'opzione `MSDTC`, puoi modificarne le impostazioni. Per informazioni su come modificare le impostazioni dell'opzione, consulta Modifica di un'impostazione di un'opzione.

> **Note**
>
> Alcune modifiche alle impostazioni delle opzioni MSDTC richiedono il riavvio del servizio MSDTC. Questo requisito può influenzare l'esecuzione di transazioni distribuite.

Disabilitazione di MSDTC

Per disabilitare MSDTC, rimuovere l'opzione `MSDTC` dal relativo gruppo di opzioni.

Console

Per rimuovere l'opzione MSDTC dal suo gruppo di opzioni

1. Accedi alla AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel pannello di navigazione scegliere Option groups (Gruppi di opzioni).
3. Scegliere il gruppo di opzioni con l'opzione `MSDTC` (`msdtc-se-2016` negli esempi precedenti).
4. Scegliere Delete option (Elimina opzione).
5. In Opzioni di eliminazione, scegliere MSDTC per Opzioni da eliminare.
6. In Apply immediately (Applica immediatamente), scegliere Yes (Sì) per eliminare immediatamente l'opzione oppure No per eliminarla nella finestra di manutenzione successiva.
7. Scegliere Delete (Elimina).

CLI

Per rimuovere l'opzione MSDTC dal suo gruppo di opzioni

- Utilizzare uno dei seguenti comandi.

  Example

  Per LinuxmacOS, oUnix:

```
aws rds remove-option-from-option-group \
    --option-group-name msdtc-se-2016 \
    --options MSDTC \
    --apply-immediately
```

  Per Windows:

```
aws rds remove-option-from-option-group ^
    --option-group-name msdtc-se-2016 ^
    --options MSDTC ^
    --apply-immediately
```

Risoluzione dei problemi relativi a MSDTC per RDS for SQL Server

In alcuni casi, potrebbero verificarsi problemi durante il tentativo di stabilire una connessione tra MSDTC in esecuzione su un computer client e il servizio MSDTC in esecuzione in un'istanza database RDS for SQL Server. In tal caso, assicurati che siano soddisfatte le seguenti condizioni:

- Le regole in entrata per il gruppo di sicurezza associato all'istanza database sono configurate correttamente. Per ulteriori informazioni, consulta Impossibile connettersi all'istanza database di Amazon RDS.
- Il computer client è configurato correttamente.
- Le regole del firewall MSDTC sul computer client sono abilitate.

Per configurare il computer client

1. Aprire Servizi di componenti.

   Oppure, in Server Manager, scegliere Strumenti e quindi Servizi di componenti.

2. Espandere Servizi di componenti, espandere Computer, espandere Risorse del computer e quindi espandere Distributed Transaction Coordinator.
3. Aprire il menu contestuale (pulsante destro del mouse) per DTC locale e scegliere Proprietà.
4. Scegliere la scheda Sicurezza .
5. Scegliere tutte le opzioni seguenti:

   - Accesso DTC di rete
   - Consenti in entrata
   - Consenti in uscita
6. Assicurarsi di scegliere la modalità di autenticazione corretta:

   - Autenticazione reciproca obbligatoria – Il computer client viene aggiunto allo stesso dominio di altri nodi che partecipano alla transazione distribuita oppure esiste una relazione di attendibilità configurata tra domini.
   - Nessuna autenticazione richiesta – Tutti gli altri casi.
7. Scegliere OK per salvare le modifiche.
8. Se viene richiesto di riavviare il servizio, scegliere Sì.

Per abilitare le regole firewall MSDTC

1. Aprire Windows Firewall, quindi scegliere Impostazioni avanzate.

   Oppure, in Server Manager, scegliere Strumenti, quindi selezionare Windows Firewall con sicurezza avanzata.

   > **Note**
   >
   > A seconda del sistema operativo in uso, Windows Firewall potrebbe essere chiamato Windows Defender Firewall.
2. Scegliere Regole in entrata nel riquadro sinistro.
3. Abilitare le regole del firewall riportate di seguito, se non sono già abilitate:

   - Distributed Transaction Coordinator (RPC)
   - Distributed Transaction Coordinator (RPC)-EPMAP
   - Distributed Transaction Coordinator (TCP-In)

4. Chiudere Windows Firewall.

Attività DBA frequenti per Microsoft SQL Server

In questa sezione vengono descritte le implementazioni, specifiche per Amazon RDS, di alcune attività DBA frequenti per le istanze database che eseguono il motore di database Microsoft SQL Server. Per fornire un'esperienza di servizio gestito, Amazon RDS non fornisce accesso shell alle istanze database e limita l'accesso a certe procedure e tabelle di sistema che richiedono privilegi avanzati.

> **Note**
>
> Quando utilizzi un'istanza database SQL Server, puoi eseguire script per modificare un database appena creato, ma non puoi modificare il database [model], ossia quello che serve da modello per nuovi database.

Argomenti

- Utilizzo di file di traccia e file dump

Accesso al database tempdb sulle istanze database Microsoft SQL Server su Amazon RDS

Puoi accedere al database `tempdb` sulle tue istanze database Microsoft SQL Server su Amazon RDS. Puoi eseguire il codice su `tempdb` servendoti di Transact-SQL tramite Microsoft SQL Server Management Studio (SSMS) o qualsiasi altra applicazione client SQL standard. Per ulteriori informazioni sulla connessione alla tua istanza database, consulta Connessione a un'istanza database che esegua il motore di database di Microsoft SQL Server.

L'utente master per la tua istanza database riceve l'accesso `CONTROL` a `tempdb` affinché possa modificare le opzioni del database `tempdb`. L'utente master non è il proprietario del database `tempdb`. Se necessario, l'utente master può concedere l'accesso `CONTROL` ad altri utenti affinché anch'essi possano modificare le opzioni del database `tempdb`.

> **Note**
>
> Non puoi eseguire i comandi Database Console Commands (DBCC) nel database `tempdb`.

Modifica delle opzioni del database tempdb

Puoi modificare le opzioni di database nel database `tempdb` sulle tue istanze database Amazon RDS. Per ulteriori informazioni sulle opzioni che puoi modificare, consulta Database tempdb nella documentazione di Microsoft.

Opzioni di database come, ad esempio, le opzioni per le dimensioni massime dei file, permangono dopo il riavvio dell'istanza database. Puoi modificare le opzioni di database per ottimizzare le prestazioni durante l'importazione dei dati e per evitare di esaurire lo spazio di storage.

Ottimizzazione delle prestazioni durante l'importazione dei dati

Per ottimizzare le prestazioni durante l'importazione di grandi quantità di dati nella tua istanza database, imposta le proprietà `SIZE` e `FILEGROWTH` del database tempdb su un numero grande. Per ulteriori informazioni su come ottimizzare `tempdb`, consulta Ottimizzazione delle prestazioni di tempdb nella documentazione di Microsoft.

L'esempio seguente mostra come impostare la dimensione su 100 GB e la crescita file su 10 percento.

```
alter database[tempdb] modify file (NAME = N'templog', SIZE=100GB, FILEGROWTH = 10%)
```

Come evitare problemi di storage

Per evitare che il database `tempdb` utilizzi tutto lo spazio su disco disponibile, imposta la proprietà `MAXSIZE`. L'esempio seguente mostra come impostare la proprietà su 2048 MB.

```
alter database [tempdb] modify file (NAME = N'templog', MAXSIZE = 2048MB)
```

Riduzione del database tempdb

Puoi scegliere tra due modi per ridurre il database `tempdb` sulla tua istanza database Amazon RDS. Puoi utilizzare la procedura `rds_shrink_tempdbfile` oppure impostare la proprietà `SIZE`.

Utilizzo della procedura rds_shrink_tempdbfile

La procedura `msdb.dbo.rds_shrink_tempdbfile` Amazon RDS permette di ridurre il database `tempdb`. Puoi chiamare `rds_shrink_tempdbfile` soltanto se disponi dell'accesso `CONTROL` a `tempdb`. Quando chiami `rds_shrink_tempdbfile` non si verifica alcun tempo di inattività per la tua istanza database.

La procedura `rds_shrink_tempdbfile` include i seguenti parametri.

Nome del parametro	Tipo di dati	Default	Campo obbligatorio	Descrizione
`@temp_filename`	SYSNAME	—	obbligatorio	Il nome logico del file da ridurre.
`@target_size`	int	nullo	facoltativo	La nuova dimensione del file in megabyte.

Nell'esempio seguente vengono ottenuti i nomi dei file per il database `tempdb`.

```
use tempdb;
GO
```

```
select name, * from sys.sysfiles;
GO
```

Nell'esempio seguente le dimensioni del file del database `tempdb` denominato `test_file` vengono ridotte e viene richiesta una nuova dimensione di `10` MB:

```
exec msdb.dbo.rds_shrink_tempdbfile @temp_filename = N'test_file', @target_size = 10;
```

Impostazione della proprietà SIZE

Puoi anche ridurre il database `tempdb` impostando la proprietà `SIZE` e riavviando l'istanza database. Per ulteriori informazioni sul riavvio dell'istanza database, consulta Riavvio di un'istanza database.

L'esempio seguente mostra come impostare la proprietà `SIZE` su 1024 MB.

```
alter database [tempdb] modify file (NAME = N'templog', SIZE = 1024MB)
```

Configurazione TempDB per implementazioni Multi-AZ

Se l'istanza DB di RDS per SQL Server si trova in una distribuzione Multi-AZ utilizzando Database Mirroring (DBM) o Always On Availability Groups (AG), tieni presenti le seguenti considerazioni per l'utilizzo del database. `tempdb`

Non è possibile replicare `tempdb` i dati dall'istanza DB principale all'istanza DB secondaria. Quando esegui il failover su un'istanza DB secondaria, `tempdb` su quell'istanza DB secondaria sarà vuota.

È possibile sincronizzare la configurazione delle opzioni del `tempdb` database, comprese le impostazioni relative al dimensionamento dei file e alla crescita automatica, dall'istanza DB principale all'istanza DB secondaria. La sincronizzazione della `tempDB` configurazione è supportata in tutte le versioni di RDS per SQL Server. È possibile attivare la sincronizzazione automatica della `tempdb` configurazione utilizzando la seguente procedura memorizzata:

```
EXECUTE msdb.dbo.rds_set_system_database_sync_objects @object_types = 'TempDbFile';
```

> **Important**
>
> Prima di utilizzare la `rds_set_system_database_sync_objects` stored procedure, assicurati di aver impostato la `tempdb` configurazione preferita sull'istanza DB principale, anziché sull'istanza DB secondaria. Se hai apportato la modifica alla configurazione

sull'istanza DB secondaria, la `tempdb` configurazione preferita potrebbe essere eliminata quando attivi la sincronizzazione automatica.

È possibile utilizzare la seguente funzione per confermare se la sincronizzazione automatica della `tempdb` configurazione è attivata:

```
SELECT * from msdb.dbo.rds_fn_get_system_database_sync_objects();
```

Quando la sincronizzazione automatica della `tempdb` configurazione è attivata, verrà restituito un valore per il `object_class` campo. Quando è disattivato, non viene restituito alcun valore.

È possibile utilizzare la seguente funzione per trovare l'ultima volta che gli oggetti sono stati sincronizzati, in ora UTC:

```
SELECT * from msdb.dbo.rds_fn_server_object_last_sync_time();
```

Ad esempio, se hai modificato la `tempdb` configurazione all'01:00 e poi esegui la `rds_fn_server_object_last_sync_time` funzione, il valore restituito `last_sync_time` dovrebbe essere successivo alle 01:00, a indicare che è avvenuta una sincronizzazione automatica.

Se si utilizza anche la replica dei lavori di SQL Server Agent, è possibile abilitare la replica sia per i lavori di SQL Agent che per la `tempdb` configurazione fornendoli nel parametro: `@object_type`

```
EXECUTE msdb.dbo.rds_set_system_database_sync_objects @object_types =
 'SQLAgentJob,TempDbFile';
```

Per ulteriori informazioni sulla replica dei processi di SQL Server Agent, vedere. Attivazione della replica di processo SQL Server Agent

In alternativa all'utilizzo della `rds_set_system_database_sync_objects` stored procedure per garantire che le modifiche alla `tempdb` configurazione vengano sincronizzate automaticamente, è possibile utilizzare uno dei seguenti metodi manuali:

> **Note**
>
> Si consiglia di attivare la sincronizzazione automatica della `tempdb` configurazione utilizzando la `rds_set_system_database_sync_objects` stored procedure. L'utilizzo

della sincronizzazione automatica evita la necessità di eseguire queste attività manuali ogni volta che si modifica la `tempdb` configurazione.

- Modifica innanzitutto la tua istanza database e disattiva Multi-AZ, quindi modifica tempdb e infine riattiva Multi-AZ. Questo sistema non causa alcun tempo di inattività.

  Per ulteriori informazioni, consulta Modifica di un'istanza database Amazon RDS.
- Modifica dapprima `tempdb` nell'istanza primaria originale, quindi effettua il failover manualmente e infine modifica `tempdb` nella nuova istanza primaria. Questo sistema causa un tempo di inattività.

  Per ulteriori informazioni, consulta Riavvio di un'istanza database.

Analisi del carico di lavoro del database su un'istanza database Amazon RDS for SQL Server con Tuning Advisor motore di database

Database Engine Tuning Advisor è un'applicazione client fornita da Microsoft che analizza il carico di lavoro dei database e suggerisce un insieme ottimale di indici per i database Microsoft SQL Server in base al tipo di query eseguite. Come SQL Server Management Studio, Tuning Advisor viene eseguito da un computer client che si connette all'istanza database Amazon RDS che esegue SQL Server. Il computer client può essere un computer eseguito in locale nella tua rete o può essere un'istanza Amazon EC2 Windows in esecuzione nella stessa regione dell'istanza database Amazon RDS.

Questa sezione mostra come acquisire un carico di lavoro affinché Tuning Advisor lo analizzi. Questa è la procedura consigliata per acquisire un carico di lavoro perché Amazon RDS limita l'accesso host all'istanza SQL Server. Per ulteriori informazioni, consulta Database Engine Tuning Advisor nella documentazione Microsoft.

Per utilizzare Tuning Advisor, occorre fornire ad Advisor ciò che chiamiamo "carico di lavoro". Un carico di lavoro è un insieme di istruzioni Transact-SQL che vengono eseguite su uno o più database che desideri ottimizzare. Durante l'ottimizzazione dei database, Database Engine Tuning Advisor si serve di file e tabelle di traccia, script Transact-SQL o file XML come input del carico di lavoro. Quando usi Amazon RDS, un carico di lavoro può essere un file in un computer client o una tabella di database in un database Amazon RDS for SQL Server accessibile al computer client. Il file o la tabella devono contenere query sul database che desideri ottimizzare in un formato riproducibile.

Per ottenere la massima efficacia di Tuning Advisor, i carichi di lavoro dovrebbero essere il più possibile realistici. Puoi generare un file o una tabella del carico di lavoro creando una traccia

dell'istanza database. Quando una traccia è in esecuzione, puoi simulare un carico sull'istanza database oppure eseguire le applicazioni con carico normale.

Esistono due tipi di tracce: lato client e lato server. Le tracce lato client sono più facili da configurare e puoi osservare in tempo reale gli eventi di traccia che vengono acquisiti in SQL Server Profiler. Una traccia lato server è più complicata da configurare e richiede una certa quantità di scripting Transact-SQL. Inoltre, la traccia occupa spazio di storage perché viene trascritta in un file sull'istanza database in Amazon RDS. È importante monitorare quanto spazio di storage viene utilizzato da una traccia in esecuzione lato server perché l'istanza database potrebbe acquisire lo stato di storage completo e non sarebbe più disponibile se lo spazio di storage venisse esaurito.

Per le tracce lato client, dopo che una quantità sufficiente di dati di traccia è stata acquisita in SQL Server Profiler, puoi generare il file del carico di lavoro salvando la traccia in un file sul computer locale o in una tabella di database su un'istanza database accessibile dal computer client. Il principale svantaggio dell'utilizzo di una traccia lato client consiste nel fatto che la traccia potrebbe non essere in grado di acquisire tutte le query in condizioni di carico intenso. Ciò potrebbe rendere meno efficace l'analisi eseguita da Database Engine Tuning Advisor. Se devi eseguire una traccia con carichi intensi e desideri fare in modo che tale traccia acquisisca ogni query di una sessione di traccia, è preferibile utilizzare una traccia lato server.

Per le tracce lato server, devi memorizzare i file di traccia sull'istanza database in un file del carico di lavoro idoneo oppure puoi salvare la traccia in una tabella sull'istanza database dopo il suo completamento. Puoi utilizzare SQL Server Profiler per salvare la traccia in un file sul computer o fare in modo che Tuning Advisor legga la tabella di traccia sull'istanza database.

Esecuzione di una traccia lato client su un'istanza database SQL Server

Per eseguire una traccia lato client su un'istanza database SQL Server

1. Avvia SQL Server Profiler. SQL Server Profiler è installato nella sottocartella Performance Tools della cartella della tua istanza SQL Server. Per avviare una traccia lato client, devi caricare o definire un modello di definizione di traccia.
2. Nel menu SQL Server Profiler File (File di SQL Server Profiler), fai clic su New Trace (Nuova traccia). Nella casella di dialogo Connect to Server (Connessione al server), immetti l'endpoint dell'istanza database, la porta, il nome utente e la password master per il database per cui desideri eseguire la traccia.
3. Nella casella di dialogo Trace Properties (Proprietà traccia), immetti un nome per la traccia e scegli un modello di definizione della traccia. Un modello predefinito, TSQL_Replay, viene fornito

con l'applicazione. Puoi modificare questo modello per definire la tua traccia. Modifica gli eventi e le relative informazioni nella scheda Events Selection (Selezione eventi) della casella di dialogo Trace Properties.

Per ulteriori informazioni sui modelli di definizione della traccia e sull'utilizzo di SQL Server Profiler per specificare una traccia lato client, consulta Database Engine Tuning Advisor nella documentazione Microsoft..

4. Avvia la traccia lato client e osserva in tempo reale le query SQL che vengono eseguite sull'istanza database.
5. Selezionare Stop Trace (Arresta traccia) dal menu File (File) quando hai completato la traccia. Salva i risultati in un file o come tabella di traccia sull'istanza database.

Esecuzione di una traccia lato server su un'istanza database SQL Server

Scrivere script per la creazione di una traccia lato server può essere complicato e non rientra nell'ambito di questo documento. Questa sezione contiene script che puoi utilizzare come esempio. Come per le tracce lato client, l'obiettivo è creare un file del carico di lavoro o una tabella di traccia che puoi aprire con Database Engine Tuning Advisor.

Di seguito è riportato uno script sintetico di esempio per l'avvio di una traccia lato server e l'acquisizione dei dettagli in un file del carico di lavoro. All'inizio dello script la traccia viene salvata nel file RDSTrace.trc nella directory D:\RDSDBDATA\Log e viene eseguito il rollover ogni 100 MB, pertanto i file di traccia sequenziali sono denominati RDSTrace_1.trc, RDSTrace_2.trc, ecc.

```
DECLARE @file_name NVARCHAR(245) = 'D:\RDSDBDATA\Log\RDSTrace';
DECLARE @max_file_size BIGINT = 100;
DECLARE @on BIT = 1
DECLARE @rc INT
DECLARE @traceid INT

EXEC @rc = sp_trace_create @traceid OUTPUT, 2, @file_name, @max_file_size
IF (@rc = 0) BEGIN
   EXEC sp_trace_setevent @traceid, 10, 1, @on
   EXEC sp_trace_setevent @traceid, 10, 2, @on
   EXEC sp_trace_setevent @traceid, 10, 3, @on
 . . .
   EXEC sp_trace_setfilter @traceid, 10, 0, 7, N'SQL Profiler'
   EXEC sp_trace_setstatus @traceid, 1
   END
```

L'esempio seguente consiste in uno script per l'arresto di una traccia. La traccia creata dallo script precedente continua a essere eseguita finché non la arresti esplicitamente o il processo esaurisce lo spazio su disco.

```
DECLARE @traceid INT
SELECT @traceid = traceid FROM ::fn_trace_getinfo(default)
WHERE property = 5 AND value = 1 AND traceid <> 1

IF @traceid IS NOT NULL BEGIN
   EXEC sp_trace_setstatus @traceid, 0
   EXEC sp_trace_setstatus @traceid, 2
END
```

Puoi salvare i risultati di una traccia lato server in una tabella di database e servirti di questa tabella come carico di lavoro per Tuning Advisor utilizzando la funzione fn_trace_gettable. Con i comandi seguenti i risultati di tutti i file denominati RDSTrace.trc presenti nella directory D:\rdsdbdata\Log, inclusi tutti i file di rollover come RDSTrace_1.trc, vengono caricati in una tabella denominata RDSTrace nel database corrente.

```
SELECT * INTO RDSTrace
FROM fn_trace_gettable('D:\rdsdbdata\Log\RDSTrace.trc', default);
```

Per salvare uno specifico file di rollover in una tabella, ad esempio il file RDSTrace_1.trc, specifica il nome del file e sostituisci il valore predefinito con 1 come ultimo parametro di fn_trace_gettable.

```
SELECT * INTO RDSTrace_1
FROM fn_trace_gettable('D:\rdsdbdata\Log\RDSTrace_1.trc', 1);
```

Esecuzione di Tuning Advisor con una traccia

Una volta creata una traccia come file locale o tabella di database, puoi eseguire Tuning Advisor sull'istanza database. Per utilizzare Tuning Advisor con Amazon RDS si segue la stessa procedura adottata quando si lavora con un'istanza SQL Server remota standalone. Puoi utilizzare l'interfaccia utente di Tuning Advisor sulla macchina client oppure l'utilità dta.exe dalla riga di comando. In entrambi i casi, devi connetterti all'istanza database Amazon RDS utilizzando il relativo endpoint e fornire il tuo nome utente e la tua password master quando utilizzi Tuning Advisor.

L'esempio di codice seguente mostra l'uso dell'utilità a riga di comando dta.exe su un'istanza database Amazon RDS con un endpoint **`dta.cnazcmklsdei.us-east-1.rds.amazonaws.com`**. Nell'esempio sono inclusi il nome utente master **`admin`** e la password utente master **`test`**, il

database di esempio da sintonizzare è denominato dal computer **C:\RDSTrace.trc**. Il codice della riga di comando di esempio specifica una sessione di traccia denominata **RDSTrace1** e i file di output nel computer locale denominati **RDSTrace.sql** per lo script di output SQL, **RDSTrace.txt** per un file dei risultati e **RDSTrace.xml** per un file XML dell'analisi. Nel database RDSDTA è anche specificata una tabella degli errori denominata **RDSTraceErrors**.

```
dta -S dta.cnazcmklsdei.us-east-1.rds.amazonaws.com -U admin -P test -D RDSDTA -
if C:\RDSTrace.trc -s RDSTrace1 -of C:\ RDSTrace.sql -or C:\ RDSTrace.txt -ox C:\
 RDSTrace.xml -e RDSDTA.dbo.RDSTraceErrors
```

Ecco il codice della riga di comando di esempio, ma il carico di lavoro di input qui è una tabella nell'istanza Amazon RDS remota denominata **RDSTrace**, che si trova nel database **RDSDTA**.

```
dta -S dta.cnazcmklsdei.us-east-1.rds.amazonaws.com -U admin -P test -D RDSDTA -it
 RDSDTA.dbo.RDSTrace -s RDSTrace1 -of C:\ RDSTrace.sql -or C:\ RDSTrace.txt -ox C:\
 RDSTrace.xml -e RDSDTA.dbo.RDSTraceErrors
```

Per un elenco completo dei parametri della riga di comando dell'utilità dta, consulta dta Utility nella documentazione di Microsoft.

Modifica di **db_owner** nell'account **rdsa** per il database

Quando crei o ripristini un database in un'istanza database di RDS per SQL Server, Amazon RDS imposta il proprietario del database su `rdsa`. Se disponi di un'implementazione multi-AZ che utilizza mirroring del database di SQL Server (DBM) o i gruppi di disponibilità Always On (AG), Amazon RDS imposta il proprietario del database sull'istanza database secondaria su `NT AUTHORITY\SYSTEM`. Il proprietario del database secondario non può essere modificato finché l'istanza database secondaria non viene promossa al ruolo principale. Nella maggior parte dei casi, impostare il proprietario del database su `NT AUTHORITY\SYSTEM` non è problematico durante l'esecuzione delle query, tuttavia può generare errori durante l'esecuzione di stored procedure nel sistema, come `sys.sp_updatestats` che richiedono autorizzazioni elevate per l'esecuzione.

È possibile utilizzare la seguente query per identificare il proprietario dei database di proprietà di `NT AUTHORITY\SYSTEM`:

```
SELECT name FROM sys.databases WHERE SUSER_SNAME(owner_sid) = 'NT AUTHORITY\SYSTEM';
```

Puoi utilizzare la stored procedure `rds_changedbowner_to_rdsa` di Amazon RDS per cambiare il proprietario del database in `rdsa`. Non è consentito utilizzare i seguenti

database con `rds_changedbowner_to_rdsa`: `master, model, msdb, rdsadmin, rdsadmin_ReportServer, rdsadmin_ReportServerTempDB, SSISDB`.

Per modificare il proprietario del database in`rdsa`, chiama la `rds_changedbowner_to_rdsa` stored procedure e fornisci il nome del database.

Example di utilizzo:

```
exec msdb.dbo.rds_changedbowner_to_rdsa 'TestDB1';
```

Il parametro seguente è obbligatorio:

- `@db_name`: il nome del database in cui modificare il proprietario del database in `rdsa`.

Regole di confronto e set di caratteri per Microsoft SQL Server

SQL Server supporta regole di confronto a più livelli. Quando crei l'istanza database, imposti le regole di confronto del server predefinite. Puoi ignorare le regole di confronto nel database, nella tabella o a livello di colonna.

Argomenti

- Regola di confronto a livello di server per Microsoft SQL Server
- Regola di confronto a livello di database per Microsoft SQL Server

Regola di confronto a livello di server per Microsoft SQL Server

Quando crei un'istanza database di Microsoft SQL Server, puoi impostare le regole di confronto del server che desideri utilizzare. Se non scegli delle regole di confronto diverse, per impostazione predefinita la regola di confronto a livello di server sarà SQL_Latin1_General_CP1_CI_AS. Le regole di confronto del server vengono applicate per impostazione predefinita a tutti i database e agli oggetti di database.

> **Note**
>
> Non è possibile modificare le regole di confronto quando si esegue il ripristino da uno snapshot DB.

Amazon RDS al momento supporta le seguenti regole di confronto del server:

Collation (Regola di confronto)	Descrizione
Arabic_CI_AS	Arabo, senza distinzione tra maiuscole e minuscole, con distinzione dei caratteri accentati, senza distinzione tra tipi di kana, senza distinzione della larghezza
Chinese_PRC_BIN2	Chinese-RPC, ordinamento dei punti di codice binario
Chinese_PRC_CI_AS	Chinese-PRC, case-insensitive, accent-se nsitive, kanatype-insensitive, width-insensitive
Chinese_Taiwan_Stroke_CI_AS	Chinese-Taiwan-Stroke, case-insensitive, accent-sensitive, kanatype-insensitive, width-insensitive
Danish_Norwegian_CI_AS	Danish-Norwegian, case-insensitive, accent-se nsitive, kanatype-insensitive, width-insensitive
Finnish_Swedish_CI_AS	Finnish, Swedish, and Swedish (Finland), case-insensitive, accent-sensitive, kanatype-insensiti ve, width-insensitive
French_CI_AS	French, case-insensitive, accent-sensitive, kanatype-insensitive, width-insensitive
Hebrew_BIN	Hebrew, binary sort
Hebrew_CI_AS	Ebraico, non sensibile al maiuscolo/minuscolo, sensibile ai caratteri accentati, non sensibile al kana, non sensibile alla larghezza
Japanese_BIN	Giapponese, ordinamento binario
Japanese_CI_AS	Japanese, case-insensitive, accent-sensitive, kanatype-insensitive, width-insensitive

Collation (Regola di confronto)	Descrizione
Japanese_CS_AS	Giapponese, non sensibile al maiuscolo/minuscolo, sensibile ai caratteri accentati, non sensibile al kana, non sensibile alla larghezza
Japanese_XJIS_140_CI_AS	Giapponese, senza distinzione tra maiuscole e minuscole, con distinzione dei caratteri accentati, senza distinzione tra tipi di kana, senza distinzione della larghezza, caratteri supplementari, senza distinzione della selezione di variazione
Japanese_XJIS_140_CI_AS_KS_VSS	Giapponese, senza distinzione tra maiuscole e minuscole, con distinzione dei caratteri accentati, con distinzione tra tipi di kana, senza distinzione della larghezza, caratteri supplementari, con distinzione della selezione di variazione
Japanese_XJIS_140_CI_AS_VSS	Giapponese, senza distinzione tra maiuscole e minuscole, con distinzione dei caratteri accentati, senza distinzione tra tipi di kana, senza distinzione della larghezza, caratteri supplementari, con distinzione della selezione di variazione
Japanese_XJIS_140_CS_AS_KS_WS	Giapponese, con distinzione tra maiuscole e minuscole, con distinzione dei caratteri accentati, con distinzione tra tipi di kana, con distinzione della larghezza, caratteri supplementari, senza distinzione della selezione di variazione
Korean_Wansung_CI_AS	Korean-Wansung, case-insensitive, accent-sensitive, kanatype-insensitive, width-insensitive
Latin1_General_100_BIN	Latin1-General-100, ordinamento binario

Collation (Regola di confronto)	Descrizione
Latin1_General_100_BIN2	Latin1-General-100, ordinamento dei punti di codice binario
Latin1_General_100_BIN2_UTF8	Latin1-General-100, ordinamento dei punti di codice binario, con codifica UTF-8
Latin1_General_100_CI_AS	Latin1-General-100, case-insensitive, accent-sensitive, kanatype-insensitive, width-insensitive
Latin1_General_100_CI_AS_SC_UTF8	Latin1-General-100, senza distinzione tra maiuscole e minuscole, con distinzione dei caratteri accentati, caratteri supplementari, con codifica UTF-8
Latin1_General_BIN	Latin1-General, binary sort
Latin1_General_BIN2	Latin1-General, ordinamento dei punti di codice binario
Latin1_General_CI_AI	Latin1-General, case-insensitive, accent-insensitive, kanatype-insensitive, width-insensitive
Latin1_General_CI_AS	Latin1-General, case-insensitive, accent-sensitive, kanatype-insensitive, width-insensitive
Latin1_General_CI_AS_KS	Latin1-General, non sensibile al maiuscolo/minuscolo, sensibile ai caratteri accentati, non sensibile al kana, non sensibile alla larghezza
Latin1_General_CS_AS	Latin1-General, con distinzione tra maiuscole e minuscole, con distinzione tra caratteri accentati, senza distinzione kana e senza distinzione larghezza
Modern_Spanish_CI_AS	Modern-Spanish, case-insensitive, accent-sensitive, kanatype-insensitive, width-insensitive

Collation (Regola di confronto)	Descrizione
Polish_CI_AS	Polacco, senza distinzione tra maiuscole e minuscole, con distinzione dei caratteri accentati, senza distinzione tra tipi di kana, senza distinzione della larghezza
SQL_1xCompat_CP850_CI_AS	Latin1-General, non sensibile al maiuscolo/minuscolo, sensibile ai caratteri accentati, non sensibile al kana, non sensibile alla larghezza per dati Unicode, SQL Server Ordinamento 49 su codepage 850 per dati non-Unicode
SQL_Latin1_General_CP1_CI_AI	Latin1-General, case-insensitive, accent-insensitive, kanatype-insensitive, width-ins ensitive per dati Unicode Data, SQL Server Sort Order 54 on Code Page 1252 per dati non-Unicode
SQL_Latin1_General_CP1_CI_AS (default)	Latin1-General, case-insensitive, accent-se nsitive, kanatype-insensitive, width-insensitive per dati Unicode, SQL Server Sort Order 52 on Code Page 1252 per dati non-Unicode
SQL_Latin1_General_CP1_CS_AS	Latin1-General, case-sensitive, accent-se nsitive, kanatype-insensitive, width-insensitive per dati Unicode, SQL Server Sort Order 51 on Code Page 1252 per dati non-Unicode
SQL_Latin1_General_CP437_CI_AI	Latin1-General, case-insensitive, accent-insensitive, kanatype-insensitive, width-ins ensitive per dati Unicode Data, SQL Server Sort Order 34 on Code Page 437 per dati non-Unicode
SQL_Latin1_General_CP850_BIN	Latin1-General, ordinamento binario per dati Unicode, SQL Server Ordinamento 40 su codepage 850 per dati non-Unicode

Collation (Regola di confronto)	Descrizione
SQL_Latin1_General_CP850_BIN2	Latin1-General, ordinamento dei punti di codice binario per dati Unicode, SQL Server Sort Order 40 on Code Page 850 per dati non-Unico de
SQL_Latin1_General_CP850_CI_AI	Latin1-General, senza distinzione tra maiuscole e minuscole, con distinzione dei caratteri accentati, senza distinzione tra tipi di kana, senza distinzione della larghezza per dati Unicode, SQL Server Ordinamento 44 su codepage 850 per dati non-Unicode
SQL_Latin1_General_CP850_CI_AS	Latin1-General, case-insensitive, accent-se nsitive, kanatype-insensitive, width-insensitive per dati Unicode, SQL Server Sort Order 42 on Code Page 850 per dati non-Unicode
SQL_Latin1_General_CP1256_CI_AS	Latin1-General, non sensibile al maiuscolo/ minuscolo, sensibile ai caratteri accentati, non sensibile al kana, non sensibile alla larghezza per dati Unicode, SQL Server Ordinamento 146 su codepage 1256 per dati non Unicode
Thai_CI_AS	Thai, case-insensitive, accent-sensitive, kanatype-insensitive, width-insensitive
Turkish_CI_AS	Turco, senza distinzione tra maiuscole e minuscole, con distinzione dei caratteri accentati, senza distinzione tra tipi di kana, senza distinzione della larghezza

Per scegliere le regole di confronto:

- Se utilizzi la console Amazon RDS, quando crei una nuova istanza database scegli Additional configuration (Configurazione aggiuntiva), quindi immetti le regole di confronto nel campo Collation

(Regole di confronto). Per ulteriori informazioni, consulta Creazione di un'istanza database Amazon RDS.

- Se utilizzi AWS CLI, scegli l'opzione `--character-set-name` con il comando `create-db-instance`. Per ulteriori informazioni, consulta create-db-instance.
- Se utilizzi l'API Amazon RDS, scegli il parametro `CharacterSetName` con l'operazione `CreateDBInstance`. Per ulteriori informazioni, consulta CreateDBInstance.

Regola di confronto a livello di database per Microsoft SQL Server

Puoi cambiare la collazione predefinita a livello di database, tabella o colonna sovrascrivendola durante la creazione di un nuovo database o oggetto di database. Ad esempio, se la regola di confronto è SQL_Latin1_General_CP1_CI_AS, puoi modificarla in Mohawk_100_CI_AS per il supporto della regola di confronto Mohawk. Può essere eseguito il cast del tipo anche per gli argomenti di una query, in modo tale da utilizzare una collazione diversa, se necessario.

Ad esempio, la query seguente consente di cambiare la collazione predefinita per la colonna AccountName in Mohawk_100_CI_AS

```
CREATE TABLE [dbo].[Account]
 (
     [AccountID] [nvarchar](10) NOT NULL,
     [AccountName] [nvarchar](100) COLLATE Mohawk_100_CI_AS NOT NULL
 ) ON [PRIMARY];
```

Il motore di database di Microsoft SQL Server supporta lo standard Unicode tramite i tipi di dati incorporati NCHAR, NVARCHAR e NTEXT. Se ad esempio hai bisogno del supporto CJK, utilizzerai i tipi di dati Unicode per l'archiviazione di caratteri e sovrascriverai la collazione server predefinita durante la creazione di tuoi database e tabelle. Ecco alcuni collegamenti Microsoft a pagine che trattano della collazione e del supporto Unicode per SQL Server:

- Utilizzo delle collazioni
- Collazione e terminologia internazionale
- Utilizzo delle collazioni per SQL Server
- Considerazioni di carattere internazionale per i database e i motori di database

Creazione di un utente di database

Puoi creare un utente di database per l'istanza database di Amazon RDS for Microsoft SQL Server eseguendo uno script T-SQL come nell'esempio di seguito. Utilizza un'applicazione come SQL Server Management Suite (SSMS). Accedi all'istanza database come l'utente principale creato quando è stata creata l'istanza database.

```
--Initially set context to master database
USE [master];
GO
--Create a server-level login named theirname with password theirpassword
CREATE LOGIN [theirname] WITH PASSWORD = 'theirpassword';
GO
--Set context to msdb database
USE [msdb];
GO
--Create a database user named theirname and link it to server-level login theirname
CREATE USER [theirname] FOR LOGIN [theirname];
GO
```

Per un esempio di aggiunta di un utente di database a un ruolo, consulta Aggiungere un utente al ruolo SQL AgentUser .

> **Note**
>
> Se si ottengono errori di autorizzazione durante l'aggiunta di un utente, è possibile ripristinare i privilegi modificando la password per l'utente principale dell'istanza database. Per ulteriori informazioni, consultare Reimpostazione della password del ruolo db_owner.

Individuazione di un modello di ripristino per il tuo database Microsoft SQL Server

In Amazon RDS, modello di ripristino, periodo di retention e stato del database sono correlati.

È importante comprendere le conseguenze prima di apportare una modifiche a una di queste impostazioni. Ogni impostazione può influenzare le altre. Ad esempio:

- Se modifichi il modello di ripristino del database in SIMPLE o BULK_LOGGED quando è abilitata la retention dei backup, Amazon RDS reimposta il modello di ripristino su FULL entro cinque minuti

dalla modifica. Questo comporta anche l'acquisizione di uno snapshot dell'istanza database da parte di RDS.

- Se imposti la retention dei backup su `0`, RDS reimposta la modalità di ripristino su SIMPLE.
- Se modifichi il modello di ripristino del database da SIMPLE a una qualsiasi altra opzione quando la retention dei backup è impostata su `0` giorni, RDS reimposta il modello di ripristino nuovamente su SIMPLE.

> **Important**
>
> Non effettuare mai il passaggio del modello di ripristino alle istanze Multi-AZ, sebbene sembri un'operazione che puoi eseguire, —ad esempio, tramite ALTER DATABASE. La retention dei backup e quindi il modello di ripristino su "FULL" (Completo) sono necessari per Multi-AZ. Se modifichi il modello di ripristino, RDS lo reimposta immediatamente su "FULL" (Completo). Questa reimpostazione automatica forza RDS a ricreare completamente la replica. Durante il processo di ricreazione, la disponibilità del database viene ridotta per circa 30-90 minuti finché il mirroring non è pronto per il failover. Anche l'istanza database subisce un calo delle prestazioni nello stesso modo in cui avviene durante una conversione da Single-AZ a Multi-AZ. La durata di questo calo delle prestazioni dipende dalle dimensioni di storage —del database: più grande è lo storage archiviato, più a lungo durerà il calo.

Per ulteriori informazioni sui modelli di ripristino di SQL Server, consulta Modelli di ripristino (SQL Server) nella documentazione di Microsoft.

Determinazione dell'ora dell'ultimo failover

Per determinare l'ora dell'ultimo failover, utilizza la seguente stored procedure:

```
execute msdb.dbo.rds_failover_time;
```

Questa procedura restituisce le seguenti informazioni.

Parametro di output	Descrizione
errorlog_available_from	Mostra l'ora in cui i log degli errori sono disponibili nella directory dei log.

Parametro di output	Descrizione
recent_failover_time	Mostra l'ora dell'ultimo failover, se è disponibile nei log degli errori. In caso contrario mostra `null`.

> **Note**
>
> La stored procedure esegue la ricerca di tutti i log degli errori di SQL Server disponibili nella directory di log per recuperare l'ora del failover più recente. Se i messaggi di failover sono stati sovrascritti da SQL Server, la procedura non recupera l'ora di failover.

Example di nessun failover recente

Questo esempio mostra l'output quando il log degli errori non contiene alcun failover recente. Nessun failover si è verificato dal 29-04-2020 alle 23:59:00.01.

errorlog_available_from	recent_failover_time
2020-04-29 23:59:00.0100000	null

Example di failover recente

Questo esempio mostra l'output quando il log degli errori contiene un failover. Il failover più recente è stato il 05-05-2020 alle 18:57:51.89.

errorlog_available_from	recent_failover_time
2020-04-29 23:59:00.0100000	2020-05-05 18:57:51.8900000

Disattivazione degli inserti rapidi durante il caricamento in blocco

A partire da SQL Server 2016, gli inserimenti rapidi sono abilitati per impostazione predefinita. Gli inserti rapidi sfruttano la registrazione minima che si verifica mentre il database si trova nel modello

di recupero con registrazione semplice o in blocco per ottimizzare le prestazioni di inserimento. Con inserti rapidi, ogni batch di carico di massa acquisisce nuove estensioni, ignorando la ricerca di allocazione per le estensioni esistenti con spazio libero disponibile per ottimizzare le prestazioni dell'inserto.

Tuttavia, con carichi di massa di inserti rapidi con piccole dimensioni batch possono portare a un aumento dello spazio inutilizzato consumato dagli oggetti. Se non è possibile aumentare la dimensione del batch, l'abilitazione del flag di traccia 692 può contribuire a ridurre lo spazio riservato inutilizzato, ma a scapito delle prestazioni. L'attivazione di questo flag di traccia disabilita gli inserti rapidi durante il caricamento di massa dei dati in heap o indici cluster.

È possibile attivare il flag di traccia 692 come parametro di avvio utilizzando gruppi di parametri DB. Per ulteriori informazioni, consulta Utilizzo di gruppi di parametri.

Il flag di traccia 692 è supportato per Amazon RDS su SQL Server 2016 e versioni successive. Per ulteriori informazioni sui flag di traccia, vedere DBCC TRACEON - Trace Flags nella documentazione di Microsoft.

Rimozione di un database Microsoft SQL Server

Puoi rimuovere un database su un'istanza database Amazon RDS che esegue Microsoft SQL Server in un'implementazione Single-AZ o Multi-AZ. Per rimuovere il database, utilizzare il seguente comando:

```
--replace your-database-name with the name of the database you want to drop
EXECUTE msdb.dbo.rds_drop_database  N'your-database-name'
```

> **Note**
>
> Utilizza virgolette singole diritte nel comando. In caso contrario, si verifica un errore.

Dopo aver utilizzato questa procedura per la rimozione del database, Amazon RDS rimuove tutte le connessioni al database esistenti e la cronologia dei backup del database.

Ridenominazione di un database Microsoft SQL Server in un'implementazione Multi-AZ

Per assegnare un nuovo nome a un'istanza database Microsoft SQL Server che utilizza Multi-AZ, utilizzare la procedura seguente:

1. Innanzitutto, disattivare Multi-AZ per l'istanza database.
2. Rinomina il database eseguendo `rdsadmin.dbo.rds_modify_db_name`.
3. Quindi, attiva il mirroring Multi-AZ o i gruppi di disponibilità AlwaysON per l'istanza database, per riportarla nel suo stato originario.

Per ulteriori informazioni, consulta Aggiunta di Multi-AZ a un'istanza database di Microsoft SQL Server.

> **Note**
>
> Se l'istanza non utilizza Multi-AZ, non è necessario modificare alcuna impostazione prima o dopo aver eseguito `rdsadmin.dbo.rds_modify_db_name`.

Esempio: nell'esempio seguente la stored procedure `rdsadmin.dbo.rds_modify_db_name` rinomina un database da **`MOO`** a **`ZAR`**. Ciò equivale all'esecuzione dell'istruzione `DDL ALTER DATABASE [MOO] MODIFY NAME = [ZAR]`.

```
EXEC rdsadmin.dbo.rds_modify_db_name N'MOO', N'ZAR'
GO
```

Reimpostazione della password del ruolo **`db_owner`**

Se ti escludi dal ruolo `db_owner` sul tuo database Microsoft SQL Server, puoi ripristinare la password del ruolo `db_owner` modificando la password master dell'istanza database. In questo modo puoi riottenere l'accesso all'istanza database, accedere ai database utilizzando la password così modificata per `db_owner` e ripristinare i privilegi del ruolo `db_owner` che potresti avere revocato per errore. Puoi modificare la password dell'istanza database utilizzando la console Amazon RDS, il comando AWS CLI modify-db-instance oppure l'operazione ModifyDBInstance. Per ulteriori

informazioni sulla modifica di un'istanza database, consulta Modifica di un'istanza database Amazon RDS.

Ripristino di istanze database terminate in base alla licenza

Microsoft ha richiesto che alcuni clienti Amazon RDS che non hanno segnalato le proprie informazioni sulla mobilità delle licenze Microsoft interrompano la propria istanza database. Amazon RDS acquisisce snapshot di queste istanze database ed è possibile eseguire il ripristino dallo snapshot in una nuova istanza database con il modello con licenza inclusa.

Puoi eseguire il ripristino da una snapshot di Standard Edition creando un'istanza Standard Edition o Enterprise Edition.

Puoi eseguire il ripristino da una snapshot di Enterprise Edition creando un'istanza Standard Edition o Enterprise Edition.

Per eseguire il ripristino da una snapshot SQL Server dopo che Amazon RDS ha creato una snapshot finale dell'istanza

1. Accedi alla AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, selezionare Snapshots (Snapshot).
3. Selezionare lo snapshot dell'istanza database SQL Server. Amazon RDS crea uno snapshot finale dell'istanza database. Il nome della snapshot dell'istanza terminata è nel formato `instance_name-final-snapshot`. Ad esempio, se il nome dell'istanza database è **`mytest.cdxgahslksma.us-east-1.rds.com`**, lo snapshot finale viene chiamato **`mytest-final-snapshot`** e si trova nella stessa regione AWS dell'istanza database originale.
4. Per Actions (Operazioni), selezionare Restore Snapshot (Ripristina snapshot).

   Viene visualizzata la pagina Restore DB Instance (Ripristina istanza database).
5. Per License Model (Modello di licenza), scegliere license-included (licenza inclusa).
6. Scegliere il motore di database di SQL Server che si desidera utilizzare.
7. Per DB Instance Identifier (Identificatore istanze database), inserire il nome per l'istanza database ripristinata.
8. Selezionare Restore DB Instance (Ripristina istanza database).

Per ulteriori informazioni sul ripristino da una snapshot, consulta Ripristino da uno snapshot database.

Transizione di un database Microsoft SQL Server da OFFLINE a ONLINE

Puoi far passare il database Microsoft SQL Server in un'istanza database Amazon RDS da `OFFLINE` a `ONLINE`.

Metodo SQL Server	Metodo Amazon RDS
ALTER DATABASE *db_name* SET ONLINE;	EXEC rdsadmin.dbo.rds_set_database_online *db_name*

Uso di Change Data Capture

Amazon RDS supporta Change Data Capture (CDC) per le istanze di database che eseguono Microsoft SQL Server. CDC "cattura" le modifiche effettuate ai dati delle tue tabelle. Memorizza i metadati di ogni modifica, ai quali potrai accedere successivamente. Per ulteriori informazioni sul funzionamento di CDC, consulta Change Data Capture nella documentazione di Microsoft.

Prima di usare CDC con le istanze database Amazon RDS, abilitalo nel database eseguendo `msdb.dbo.rds_cdc_enable_db`. Devi avere i privilegi dell'utente master per abilitare CDC nell'istanza database Amazon RDS. Dopo l'abilitazione di CDC, qualsiasi utente che sia `db_owner` del database interessato può abilitare o disabilitare CDC sulle tabelle di tale database.

> **Important**
>
> Durante i ripristini, CDC verrà disabilitata. Tutti i metadati correlati saranno rimossi automaticamente dal database. Ciò vale per il ripristino di snapshot e point-in-time e per il ripristino nativo di SQL Server da parte di S3. Dopo l'esecuzione di uno di questi tipi di ripristino, puoi riabilitare CDC e specificare di nuovo le tabelle da monitorare.

Per abilitare CDC per un'istanza DB, esegui la procedura archiviata `msdb.dbo.rds_cdc_enable_db`.

```
exec msdb.dbo.rds_cdc_enable_db 'database_name'
```

Per disabilitare CDC per un'istanza DB, esegui la procedura archiviata `msdb.dbo.rds_cdc_disable_db`.

```
exec msdb.dbo.rds_cdc_disable_db 'database_name'
```

Argomenti

- Monitoraggio di tabelle con Change Data Capture
- Processi di Change Data Capture
- Change Data Capture per istanze Multi-AZ

Monitoraggio di tabelle con Change Data Capture

Dopo l'abilitazione di CDC sul database, puoi iniziare a monitorare specifiche tabelle. Puoi scegliere le tabelle da monitorare eseguendo sys.sp_cdc_enable_table.

```
--Begin tracking a table
exec sys.sp_cdc_enable_table
   @source_schema           = N'source_schema'
,  @source_name             = N'source_name'
,  @role_name               = N'role_name'

--The following parameters are optional:

--, @capture_instance       = 'capture_instance'
--, @supports_net_changes   = supports_net_changes
--, @index_name             = 'index_name'
--, @captured_column_list   = 'captured_column_list'
--, @filegroup_name         = 'filegroup_name'
--, @allow_partition_switch = 'allow_partition_switch'
;
```

Per visualizzare la configurazione CDC per le tue tabelle, esegui sys.sp_cdc_help_change_data_capture.

```
--View CDC configuration
exec sys.sp_cdc_help_change_data_capture

--The following parameters are optional and must be used together.
--  'schema_name', 'table_name'
```

```
;
```

Per ulteriori informazioni sulle tabelle, funzioni e procedure memorizzate di CDC riportate nella documentazione di SQL Server, consulta le sezioni seguenti:

- Procedure memorizzate di Change Data Capture (Transact-SQL)
- Funzioni di Change Data Capture (Transact-SQL)
- Tabelle di Change Data Capture (Transact-SQL)

Processi di Change Data Capture

Quando abiliti CDC, SQL Server crea i relativi processi. I proprietari del database (`db_owner`) possono visualizzare, creare, modificare ed eliminare i processi di CDC. Tuttavia, l'account di sistema di RDS è il proprietario di tali processi. Pertanto, i processi non sono visibili per le viste e le procedure native o da SQL Server Management Studio.

Per controllare il comportamento di CDC in un database, utilizza procedure SQL Server native come sp_cdc_enable_table e sp_cdc_start_job . Per modificare i parametri di un'attività di CDC, ad esempio `maxtrans` e `maxscans`, puoi utilizzare sp_cdc_change_jobs. .

Per ulteriori informazioni sui processi di CDC, puoi interrogare le seguenti viste a gestione dinamica:

- sys.dm_cdc_errors
- sys.dm_cdc_log_scan_sessions
- sysjobs
- sysjobhistory

Change Data Capture per istanze Multi-AZ

Se utilizzi CDC su un'istanza Multi-AZ, assicurati che la configurazione del processo di CDC del server mirror corrisponda a quella del server principale. I processi di CDC sono mappati in `database_id`. Se gli ID database sul secondario sono diversi da quelli del principale, i processi non verranno associati al database corretto. Per prevenire gli errori dopo un failover, RDS elimina e ricrea i processi sul nuovo server principale. I processi così ricreati utilizzano i parametri registrati dal server principale prima del failover.

Anche se questa procedura si svolge rapidamente, può sempre accadere che i processi di CDC vengano eseguiti prima che RDS possa correggerli. Di seguito sono descritti tre modi per forzare i parametri affinché siano coerenti tra le repliche principali e secondarie:

- Usa gli stessi parametri di processo per tutti i database con CDC abilitata.
- Prima di modificare la configurazione di un processo di CDC, converti l'istanza Multi-AZ in Single-AZ.
- Trasferisci i parametri manualmente ogni volta che li modifichi sul server principale.

Per visualizzare e definire i parametri CDC utilizzati per ricreare i processi di CDC dopo un failover, utilizza `rds_show_configuration` e `rds_set_configuration`.

L'esempio seguente restituisce il valore impostato per `cdc_capture_maxtrans`. RDS configura automaticamente il valore di qualsiasi parametro impostato su `RDS_DEFAULT`.

```
-- Show configuration for each parameter on either primary and secondary replicas.
exec rdsadmin.dbo.rds_show_configuration 'cdc_capture_maxtrans';
```

Per impostare la configurazione sul secondario, eseguire `rdsadmin.dbo.rds_set_configuration`. Questa procedura imposta i valori del parametro per tutti i database sul server secondario. Queste impostazioni vengono utilizzate solo dopo un failover. Nell'esempio seguente, `maxtrans` viene impostato su *`1000`* per tutti i processi di acquisizione di CDC:

```
--To set values on secondary. These are used after failover.
exec rdsadmin.dbo.rds_set_configuration 'cdc_capture_maxtrans', 1000;
```

Per impostare i parametri di un'attività di CDC sul server principale, utilizza sys.sp_cdc_change_job.

Uso di SQL Server Agent

Con Amazon RDS puoi utilizzare SQL Server Agent su un'istanza database che esegue Microsoft SQL Server Enterprise Edition, Standard Edition o Web Edition. SQL Server Agent è un servizio di Microsoft Windows che esegue attività pianificate di amministrazione, dette processi. Puoi utilizzare SQL Server Agent per eseguire processi T-SQL che ricostruiscono indici, eseguono controlli anticorruzione e aggregano i dati in un'istanza database SQL Server.

Quando crei un'istanza database SQL Server, l'utente master è iscritto nel ruolo `SQLAgentUserRole`.

SQL Server Agent può eseguire un processo pianificato, in risposta a un evento specifico, oppure su richiesta. Per ulteriori informazioni, consulta SQL Server Agent nella documentazione Microsoft.

> **Note**
>
> Evita di pianificare i processi da eseguire durante le finestre di manutenzione e backup per l'istanza DB. I processi di manutenzione e backup avviati da AWS potrebbero interrompere un processo o causarne l'annullamento.
> Nelle distribuzioni Multi-AZ, i processi di SQL Server Agent vengono replicati dall'host principale all'host secondario quando la funzionalità di replica del processo è attivata. Per ulteriori informazioni, consulta Attivazione della replica di processo SQL Server Agent.
> Le implementazioni multi-AZ hanno un limite di 10.000 processi di SQL Server Agent. Se hai bisogno di un limite più alto, richiedi un aumento contattando. AWS Support Aprire la pagina del Centro di supporto AWS Support effettuando l'accesso se necessario, quindi selezionare Crea caso. Selezionare Service limit increase (Aumento limiti del servizio). Compilare e inviare il modulo.

Per visualizzare la cronologia di uno specifico processo di SQL Server Agent in SQL Server Management Studio (SSMS), apri Object Explorer, fai clic con il pulsante destro del mouse sul processo e seleziona View History (Visualizza cronologia).

Poiché SQL Server Agent è in esecuzione su un host gestito in un'istanza DB, alcune azioni non sono supportate:

- L'esecuzione di processi di replica e l'esecuzione di script da riga di comando utilizzando ActiveX, la shell dei comandi di Windows o Windows non sono supportate. PowerShell
- Non è possibile avviare, arrestare o riavviare manualmente SQL Server Agent.
- Le notifiche e-mail tramite SQL Server Agent non sono disponibili da un'istanza database.
- Gli avvisi e gli operatori di SQL Server Agent non sono supportati.
- L'utilizzo di SQL Server Agent per creare backup non è supportato. Utilizza Amazon RDS per il backup dell'istanza database.
- Attualmente, RDS per SQL Server non supporta l'uso di token SQL Server Agent.

Attivazione della replica di processo SQL Server Agent

È possibile attivare la replica dei processi SQL Server Agent utilizzando la seguente stored procedure:

```
EXECUTE msdb.dbo.rds_set_system_database_sync_objects @object_types = 'SQLAgentJob';
```

È possibile eseguire la stored procedure su tutte le versioni di SQL Server supportate da Amazon RDS for SQL Server. I processi vengono replicati nelle seguenti categorie:

- [Senza categoria (locale)]
- [Senza categoria (multi-server)]
- [Senza categoria]
- Raccoglitore di dati
- Tuning Advisor del motore del database
- Manutenzione database
- Full text

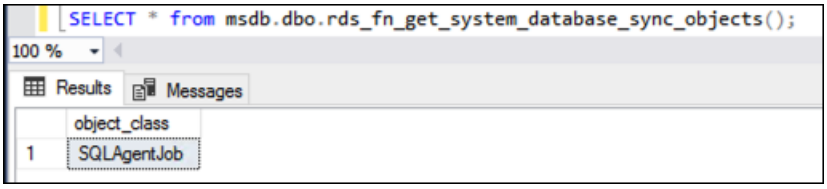
Vengono replicati solo i processi che utilizzano i passaggi del processo T-SQL. I lavori con tipi di passaggi come SQL Server Integration Services (SSIS), SQL Server Reporting Services (SSRS), Replication e non vengono replicati. PowerShell I processi che utilizzano Database Mail e gli oggetti a livello di server non vengono replicati.

> **Important**
>
> L'host principale è l'origine della replica. Prima di attivare la replica di processo, assicurarsi che i processi SQL Server Agent siano sul principale. In caso contrario, è possibile che i processi SQL Server Agent vengano eliminati se la funzionalità viene attivata quando i processi più recenti si trovano sull'host secondario.

Utilizza la seguente funzione per verificare se la replica è attivata.

```
SELECT * from msdb.dbo.rds_fn_get_system_database_sync_objects();
```

Se i processi di SQL Server Agent sono in fase di replica, la query T-SQL restituisce quanto segue. Se i processi non si stanno replicando, non restituisce nulla per `object_class`.

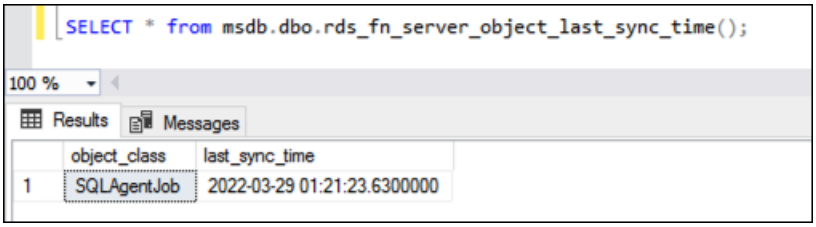


È possibile utilizzare la seguente funzione per trovare l'ultima volta che gli oggetti sono stati sincronizzati in base al fuso orario UTC.

```
SELECT * from msdb.dbo.rds_fn_server_object_last_sync_time();
```

Supponi, ad esempio, di modificare un processo SQL Server Agent all'01:00. Prevedi che l'orario di sincronizzazione più recente sia dopo l'01:00, il che suggerisce che la sincronizzazione è avvenuta.

Dopo la sincronizzazione, i valori restituiti per `date_created` e `date_modified` sul nodo secondario dovrebbero corrispondere.



Se si utilizza anche la `tempdb` replica, è possibile abilitare la replica sia per i job di SQL Agent che per la `tempdb` configurazione fornendoli nel parametro: `@object_type`

```
EXECUTE msdb.dbo.rds_set_system_database_sync_objects @object_types =
 'SQLAgentJob,TempDbFile';
```

Per ulteriori informazioni sulla `tempdb` replica, vedere. Configurazione TempDB per implementazioni Multi-AZ

Aggiungere un utente al ruolo SQL AgentUser

Per consentire un ulteriore accesso o utilizzo utente di SQL Server Agent, accedi come utente master ed esegui le operazioni seguenti.

1. Creazione di un altro login a livello di server con il comando `CREATE LOGIN`.
2. Creazione di un utente in `msdb` con il comando `CREATE USER` e collegamento di questo utente alle credenziali di accesso create nella fase precedente.

3. Aggiunta dell'utente al `SQLAgentUserRole` utilizzando la procedura memorizzata di sistema `sp_addrolemember`.

Supponi, ad esempio, che il tuo nome utente master sia **`admin`** e di voler concedere l'accesso a SQL Server Agent a un utente denominato **`theirname`** con una password **`theirpassword`**. In tal caso, puoi utilizzare la seguente procedura.

Per aggiungere un utente al AgentUser ruolo SQL

1. Accedi come utente master.
2. Esegui i comandi seguenti:

```
--Initially set context to master database
USE [master];
GO
--Create a server-level login named theirname with password theirpassword
CREATE LOGIN [theirname] WITH PASSWORD = 'theirpassword';
GO
--Set context to msdb database
USE [msdb];
GO
--Create a database user named theirname and link it to server-level login
 theirname
CREATE USER [theirname] FOR LOGIN [theirname];
GO
--Added database user theirname in msdb to SQLAgentUserRole in msdb
EXEC sp_addrolemember [SQLAgentUserRole], [theirname];
```

Eliminazione di un processo SQL Server Agent

Puoi utilizzare la procedura archiviata `sp_delete_job` per eliminare i processi di SQL Server Agent su Amazon RDS for Microsoft SQL Server.

Non è possibile utilizzare SSMS per eliminare i processi di SQL Server Agent. Se provi a farlo, riceverai un messaggio di errore simile al seguente:

```
The EXECUTE permission was denied on the object 'xp_regread', database
 'mssqlsystemresource', schema 'sys'.
```

Come servizio gestito, a RDS viene impedita l'esecuzione di procedure che accedono al registro di Windows. Quando si utilizza SSMS, tenta di eseguire un processo (`xp_regread`) per il quale RDS non è autorizzato.

> **Note**
>
> In RDS per SQL Server, solo i membri del ruolo sysadmin possono aggiornare o eliminare i processi di proprietà di un account di accesso diverso.

Per eliminare un processo di SQL Server Agent

- Eseguire la seguente istruzione T-SQL:

```
EXEC msdb..sp_delete_job @job_name = 'job_name';
```

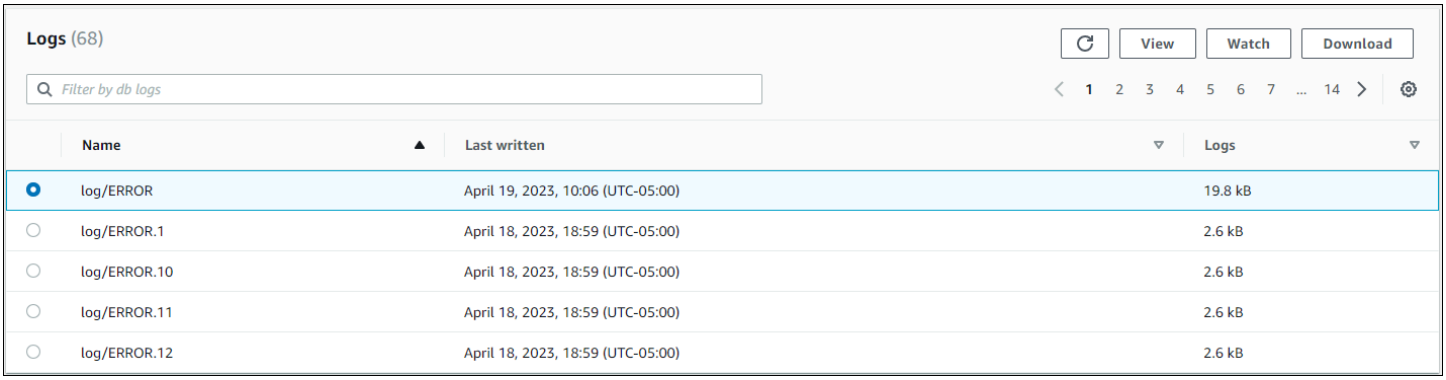
Utilizzo dei log di Microsoft SQL Server

Puoi utilizzare la console Amazon RDS per visualizzare, monitorare e scaricare i log di SQL Server Agent, Microsoft SQL Server e SQL Server Reporting Services (SSRS).

Monitoraggio dei file di log

Se visualizzi un log nella console Amazon RDS, puoi visualizzarne i contenuti così come si presentano in quel momento. I log monitorati nella console vengono aperti in una modalità dinamica che ti consente di visualizzarne gli aggiornamenti in tempo quasi reale.

Solo l'ultimo log può essere monitorato. Supponiamo, ad esempio, che ti vengano mostrati i log seguenti:



Solo il log/ERROR viene aggiornato attivamente, essendo il più recente. Puoi scegliere di monitorare altri log, ma questi sono statici e non si aggiornano.

Archiviazione dei file di log

La console Amazon RDS mostra i log relativi all'ultima settimana fino alla giornata corrente. Puoi scaricare e archiviare i log per conservarli come riferimento dopo tale periodo. Uno dei modi per archiviare i log consiste nel caricarli in un bucket Amazon S3. Per istruzioni su come configurare un bucket Amazon S3 e caricare un file, consulta le nozioni di base su Amazon S3 nella Guida alle operazioni di base di Amazon Simple Storage Service e fai clic su Get Started (Inizia).

Visualizzazione dei log dell'agente e degli errori

Per visualizzare i log dell'agente e degli errori di Microsoft SQL Server, usa la stored procedure `rds_read_error_log` in Amazon RDS con i parametri seguenti:

- **`@index`** – Versione del log da recuperare. Il valore predefinito è 0 (recupero del log degli errori corrente). Specifica 1 per recuperare il log precedente, 2 per recuperare il log ancora precedente e così via.
- **`@type`** – Tipo del log da recuperare. Specifica 1 per recuperare un log degli errori. Specifica 2 per recuperare un log degli agenti.

Example

Nell'esempio seguente viene richiesto il log degli errori corrente.

```
EXEC rdsadmin.dbo.rds_read_error_log @index = 0, @type = 1;
```

Per ulteriori informazioni sugli errori di SQL Server, consulta Errori del motore di database nella documentazione Microsoft.

Utilizzo di file di traccia e file dump

In questa sezione viene descritto l'utilizzo dei file di traccia e dei file dump per le istanze database Amazon RDS che eseguono Microsoft SQL Server.

Generazione di una query SQL di traccia

```
declare @rc int
```

```
declare @TraceID int
declare @maxfilesize bigint

set @maxfilesize = 5

exec @rc = sp_trace_create @TraceID output,  0, N'D:\rdsdbdata\log\rdstest',
 @maxfilesize, NULL
```

Visualizzazione di una traccia aperta

```
select * from ::fn_trace_getinfo(default)
```

Visualizzazione dei contenuti della traccia

```
select * from ::fn_trace_gettable('D:\rdsdbdata\log\rdstest.trc', default)
```

Impostazione del periodo di retention dei file di traccia e dei file dump

I file di traccia e i file dump possono accumularsi e occupare spazio su disco. Per impostazione predefinita, Amazon RDS elimina i file di traccia e i file dump che risalgono a più di sette giorni prima.

Per visualizzare il periodo di retention corrente dei file di traccia e dei file dump, utilizza la procedura `rds_show_configuration`, come illustrato nell'esempio seguente.

```
exec rdsadmin..rds_show_configuration;
```

Per modificare il periodo di retention dei file di traccia, utilizza la procedura `rds_set_configuration` e imposta `tracefile retention` in minuti. L'esempio seguente imposta il periodo di retention dei file di traccia su 24 ore.

```
exec rdsadmin..rds_set_configuration 'tracefile retention', 1440;
```

Per modificare il periodo di retention dei file dump, utilizza la procedura `rds_set_configuration` e imposta `dumpfile retention` in minuti. L'esempio seguente imposta il periodo di retention del file dump su 3 giorni.

```
exec rdsadmin..rds_set_configuration 'dumpfile retention', 4320;
```

Per motivi di sicurezza, non puoi eliminare una traccia o un file dump specifici su un'istanza database SQL Server. Per modificare tutti i file di traccia o i file dump inutilizzati, imposta il periodo di retention dei file su 0.

Amazon RDS per MySQL

Amazon RDS supporta le istanze database che eseguono le seguenti versioni di MySQL:

- MySQL 8.0
- MySQL 5.7

Per ulteriori informazioni sul supporto delle versioni secondarie, consulta Versioni di MySQL in Amazon RDS.

Creare un'istanza database di Amazon RDS per MySQL, utilizza gli strumenti di gestione o le interfacce di Amazon RDS. A questo punto puoi effettuare le seguenti operazioni:

- Ridimensionare l'istanza database
- Autorizzare le connessioni all'istanza database
- Creare e ripristinare da backup o snapshot
- Creare istanze secondarie Multi-AZ
- Creare repliche di lettura
- Monitorare le prestazioni dell'istanza database

Per archiviare e accedere ai dati nell'istanza database, utilizza le utilità e le applicazioni MySQL standard.

Amazon RDS for MySQL è conforme a molti standard di settore. Ad esempio, puoi utilizzare i database RDS per MySQL per creare applicazioni conformi a HIPAA. Puoi utilizzare database RDS per MySQL per archiviare informazioni sanitarie, inclusi dati sanitari protetti (PHI), in base a un Contratto di società in affari (BAA) con AWS. Amazon RDS per MySQL soddisfa inoltre i requisiti di sicurezza Federal Risk and Authorization Management Program (FedRAMP). Inoltre, Amazon RDS per MySQL ha ricevuto una certificazione FedRAMP Joint Authorization Board (JAB) Provisional Authority to Operate (P-ATO) alla FedRAMP HIGH Baseline all'interno delle AWS GovCloud (US). Per ulteriori informazioni sugli standard di conformità supportati, consulta Conformità di AWS Cloud.

Per ulteriori informazioni sulle caratteristiche in ogni versione di MySQL, consulta la pagina relativa alle caratteristiche principali di MySQL nella documentazione di MySQL.

Prima di creare un'istanza database, completa i passaggi in Configurazione di Amazon RDS. Quando crei un'istanza database, l'utente principale di RDS ottiene privilegi DBA, con alcune limitazioni. Utilizzare questo account per attività amministrative, ad esempio la creazione di account di database aggiuntivi.

Puoi creare:

- Istanze DB
- Snapshot DB
- P oint-in-time ripristina
- Backup automatizzati
- Backup manuali

Puoi utilizzare istanze database che eseguono MariaDB all'interno di un cloud privato virtuale (VPC) basato su Amazon VPC. Inoltre, puoi attivare varie opzioni per aggiungere altre funzionalità all'istanza database MySQL. Amazon RDS supporta le implementazioni multi-AZ per MySQL come soluzione failover a elevata disponibilità.

> **Important**
>
> Per offrire un'esperienza di servizio gestito, Amazon RDS non fornisce accesso shell alle istanze database. Limita anche l'accesso ad alcune procedure di sistema e tabelle che richiedono privilegi avanzati. Puoi accedere al database utilizzando i client SQL standard come il client mysql. Tuttavia, non è possibile accedere direttamente all'host utilizzando Telnet o Secure Shell (SSH).

Argomenti

- Supporto delle funzionalità MySQL su Amazon RDS
- Versioni di MySQL in Amazon RDS
- Connessione a un'istanza database che esegue il motore di database di MySQL
- Protezione delle connessioni di istanze database MySQL
- Prestazioni delle query migliorate per RDS per MySQL con Amazon RDS Optimized Reads
- Prestazioni di scrittura migliorate con Scritture ottimizzate per Amazon RDS per MySQL
- Aggiornamento del motore di database MySQL

- Aggiornamento di una versione del motore di snapshot MySQL DB
- Importazione di dati in un'istanza database MySQL
- Uso della replica MySQL in Amazon RDS
- Configurazione di cluster active-active per RDS for MySQL
- Esportazione di dati da un'istanza database MySQL tramite la replica
- Opzioni per le istanze database MySQL
- Parametri per MySQL
- Attività di log DBA comuni per istanze database MySQL
- Fuso orario locale per le istanze database MySQL
- Problemi e limitazioni note per Amazon RDS for MySQL
- Riferimento delle stored procedure RDS per MySQL

Supporto delle funzionalità MySQL su Amazon RDS

RDS per MySQL supporta la maggior parte delle caratteristiche e delle funzionalità di MySQL. Alcune funzionalità potrebbero avere un supporto o privilegi limitati.

Please change to "Puoi filtrare le nuove funzionalità Amazon RDS alla pagina Quali sono le novità del database?. Per Prodotti, scegli Amazon RDS. Quindi esegui la ricerca utilizzando parole chiave come **`MySQL 2022`**.

> **Note**
>
> I seguenti elenchi non sono esaustivi.

Argomenti

Motori di storage supportati per RDS for MySQL

MySQL supporta più motori di storage con funzionalità diverse, ma non tutti sono ottimizzati per il recovery e la durabilità dei dati. Amazon RDS supporta completamente il motore di archiviazione InnoDB per le istanze database MySQL. Le funzionalità Ripristino point-in-time e Ripristino di snapshot di Amazon RDS richiedono un motore di storage che supporti il recupero da arresto anomalo e sono disponibili solo per il motore di storage InnoDB. Per ulteriori informazioni, consulta Supporto per memcached MySQL.

Il motore di storage Federated non è attualmente supportato da Amazon RDS for MySQL.

Per gli schemi creati dall'utente, il motore di storage MyISAM non supporta il ripristino in modo affidabile e può causare la perdita o il danneggiamento dei dati quando si riavvia MySQL dopo un ripristino, impedendo il corretto funzionamento del ripristino point-in-time o del ripristino da uno snapshot. Tuttavia, se scegli comunque di utilizzare MyISAM con Amazon RDS gli snapshot possono essere utili in alcune situazioni.

> **Note**
>
> Le tabelle del sistema nello schema `mysql` possono trovarsi nello storage MyISAM.

Per convertire le tabelle MyISAM esistenti in tabelle InnoDB, è possibile utilizzare il comando `ALTER TABLE`, ad esempio `alter table TABLE_NAME engine=innodb;`. Tieni presente che MyISAM e InnoDB hanno diversi punti di forza e di debolezza, quindi è necessario valutare a fondo le conseguenze di questa modifica sulle applicazioni, prima di eseguirla.

MySQL 5.1, 5.5 e 5.6 non sono più supportati in Amazon RDS. Puoi tuttavia ripristinare gli snapshot di MySQL 5.1, 5.5 e 5.6 esistenti. Quando ripristini uno snapshot di MySQL 5.1, 5.5 o 5.6, l'istanza database viene aggiornata automaticamente a MySQL 5.7.

Uso di memcached e altre opzioni con MySQL su Amazon RDS

La maggior parte dei motori di database Amazon RDS supporta gruppi di opzioni che permettono di selezionare caratteristiche aggiuntive per l'istanza database. Le istanze di database RDS per MySQL supportano l'opzione `memcached`, una cache semplice basata su chiave. Per ulteriori informazioni su `memcached` e altre opzioni, consulta Opzioni per le istanze database MySQL. Per ulteriori informazioni sull'utilizzo di gruppi di opzioni, consulta Uso di gruppi di opzioni.

Precaricamento della cache InnoDB per MySQL su Amazon RDS

Il precaricamento della cache InnoDB può offrire vantaggi in termini di prestazioni per l'istanza database MySQL salvando lo stato corrente del pool di buffer quando l'istanza database viene arrestata e quindi ricaricando il pool di buffer con le informazioni salvate quando l'istanza database si avvia. Ciò evita la necessità di preparare il pool di buffer dal normale utilizzo del database e permette invece di precaricare il pool di buffer con le pagine per le query comuni note. Il file che archivia le informazioni del pool di buffer salvato archivia solo i metadati per le pagine che si trovano nel pool di buffer e non le pagine stesse. Di conseguenza, il file non richiede molto spazio di storage. La dimensione del file equivale circa allo 0,2% della dimensione della cache. Ad esempio, per una cache da 64 GiB, il file di precaricamento della cache è di 128 MiB. Per ulteriori informazioni sul precaricamento della cache InnoDB, consulta la pagina relativa al salvataggio e ripristino dello stato del pool di buffer nella documentazione di MySQL.

Le istanze database RDS per MySQL supportano il precaricamento della cache InnoDB. Per abilitare il precaricamento della cache InnoDB, imposta i parametri

`innodb_buffer_pool_dump_at_shutdown` e `innodb_buffer_pool_load_at_startup` su 1 nel gruppo di parametri per l'istanza database. La modifica dei valori di questi parametri in un gruppo di parametri ha effetto in tutte le istanze database MySQL che utilizzano tale gruppo di parametri. Per abilitare il precaricamento della cache InnoDB per istanze database MySQL specifiche, devi creare un nuovo gruppo di parametri per tali istanze. Per informazioni sui gruppi di parametri, consulta Utilizzo di gruppi di parametri.

Il precaricamento della cache InnoDB fornisce principalmente un vantaggio in termini di prestazioni per le istanze database che utilizzano lo storage standard. Se utilizzi lo storage PIOPS non riscontrerai generalmente un vantaggio significativo in termini di prestazioni.

Important

Se l'istanza database MySQL non si arresta normalmente, ad esempio durante un failover, lo stato del pool di buffer non viene salvato nel disco. In questo caso, al riavvio dell'istanza database, MySQL carica il file del pool di buffer disponibile. Ciò non comporta alcun problema, ma il pool di buffer ripristinato potrebbe non riflettere lo stato più recente del pool di buffer prima del riavvio. Per fare in modo che sia disponibile uno stato recente del pool di buffer per precaricare la cache InnoDB all'avvio, è consigliabile eseguire periodicamente un dump del pool di buffer "on demand".
Puoi creare un evento per eseguire il dump del pool di buffer automaticamente e a intervalli regolari. L'istruzione seguente crea ad esempio un evento denominato `periodic_buffer_pool_dump` che esegue il dump del pool di buffer ogni ora.

```
CREATE EVENT periodic_buffer_pool_dump
ON SCHEDULE EVERY 1 HOUR
DO CALL mysql.rds_innodb_buffer_pool_dump_now();
```

Per ulteriori informazioni sugli eventi MySQL, consulta la sintassi degli eventi nella documentazione di MySQL.

Dump e caricamento del pool di buffer on demand

Puoi salvare e caricare la cache InnoDB "on demand".

- Per eseguire il dump dello stato corrente del pool di buffer su disco, chiama la stored procedure mysql.rds_innodb_buffer_pool_dump_now.

- Per caricare lo stato salvato del pool di buffer dal disco, chiama la stored procedure mysql.rds_innodb_buffer_pool_load_now.
- Per annullare un'operazione di caricamento in corso, chiama la stored procedure mysql.rds_innodb_buffer_pool_load_abort.

Caratteristiche di MySQL non supportate da Amazon RDS

Al momento, Amazon RDS non supporta le caratteristiche seguenti di MySQL:

- Plug-in di autenticazione
- Registrazione degli errori nel log di sistema
- Crittografia di spazi tabelle InnoDB
- Plug-in per la complessità della password
- Variabili di sistema persistenti
- Plugin di riscrittura Rewriter Query
- Replica semi-sincrona
- Spazio di tabella trasportabile
- Plug-in IDE

> **Note**
>
> Gli ID transazione globale sono supportati per tutte le versioni di RDS per MySQL 5.7 e per RDS per MySQL 8.0.26 e versioni successive alla 8.0.

Per offrire un'esperienza di servizio gestito, Amazon RDS non fornisce accesso shell alle istanze database. Limita anche l'accesso ad alcune procedure di sistema e tabelle che richiedono privilegi avanzati. Amazon RDS supporta l'accesso ai database in un'istanza database con qualsiasi applicazione client SQL standard. Amazon RDS non consente l'accesso host diretto a un'istanza database tramite Telnet, Secure Shell (SSH) o Windows Remote Desktop Connection. Quando crei un'istanza DB, vieni assegnato come db_owner per tutti i database su quell'istanza e disponi di tutte le autorizzazioni a livello di database ad eccezione di quelle utilizzate per i backup. Amazon RDS gestisce i backup per tuo conto.

Versioni di MySQL in Amazon RDS

Per MySQL, i numeri di versione sono organizzati come versione X.Y.Z. Nella terminologia di Amazon RDS; X.Y indica la versione principale e Z è il numero di versione secondaria. Per le implementazioni di Amazon RDS, una modifica di versione è considerata principale se cambia il numero di versione principale—ad esempio nel caso di un passaggio dalla versione 5.7 alla 8.0. Una modifica di versione è considerata secondaria se cambia solo il numero della versione secondaria, ad esempio se si passa dalla versione 8.0.32 alla 8.0.34.

Argomenti

Versioni secondarie di MySQL supportate in Amazon RDS

Attualmente Amazon RDS supporta le versioni secondarie di MySQL seguenti.

> **Note**
>
> Le date con solo un mese e un anno sono approssimative e vengono aggiornate con una data esatta quando nota.
> Amazon RDS Extended Support non è disponibile per le versioni minori.

Versione del motore MySQL	Data di rilascio nella community	Data di rilascio per RDS	Data di fine del supporto standard RDS
8.0			

Versione del motore MySQL	Data di rilascio nella community	Data di rilascio per RDS	Data di fine del supporto standard RDS
8,0,36	16 gennaio 2024	12 febbraio 2024	Marzo 2025
8.0.35	25 ottobre 2023	9 novembre 2023	Marzo 2025
8,0,34	18 luglio 2023	9 agosto 2023	Settembre 2024
8,0,33	18 aprile 2023	15 giugno 2023	Settembre 2024
8,0,32	17 gennaio 2023	7 febbraio 2023	Settembre 2024
5.7			
5,7.44*	25 ottobre 2023	2 novembre 2023	29 febbraio 2024

* Questa versione secondaria continuerà a essere disponibile quando la versione principale sarà disponibile in Amazon RDS Extended Support. Per ulteriori informazioni, consulta Utilizzo dell'estensione del supporto per Amazon RDS.

Le versioni secondarie possono raggiungere la fine del supporto standard prima che lo facciano le versioni principali. Ad esempio, la versione secondaria 8.0.28 ha raggiunto la fine della data di supporto standard il 28 marzo 2024, mentre la versione principale 8.0 raggiungerà tale data il 31 luglio 2026. RDS supporterà versioni secondarie 8.0.* aggiuntive che la comunità MySQL rilascerà tra queste date.

Quando crei una nuova istanza database, puoi specificare qualsiasi versione di MySQL attualmente supportata. Puoi specificare la versione principale MySQL 5.7 e qualsiasi versione secondaria supportata per la versione principale specificata. Se non viene specificata alcuna versione, Amazon RDS utilizza per impostazione predefinita una versione supportata, in genere la più recente. Se viene specificata una versione principale ma non una secondaria, per impostazione predefinita Amazon RDS utilizza una release recente della versione principale specificata. Per visualizzare un elenco delle versioni supportate, nonché le impostazioni predefinite per le istanze DB appena create, usa il comando. `describe-db-engine-versions` AWS CLI

Ad esempio, per visualizzare l'elenco delle versioni di motori supportate per RDS per MySQL. esegui il comando CLI seguente:

```
aws rds describe-db-engine-versions --engine mysql --query "*[].
{Engine:Engine,EngineVersion:EngineVersion}" --output text
```

La versione predefinita di MySQL potrebbe variare in base alla Regione AWS. Per creare un'istanza DB con una versione secondaria specifica, specifica la versione secondaria durante la creazione dell'istanza DB. È possibile determinare la versione secondaria predefinita di an Regione AWS utilizzando il seguente comando: AWS CLI

```
aws rds describe-db-engine-versions --default-only --engine mysql
 --engine-version major-engine-version --region region --query "*[].
{Engine:Engine,EngineVersion:EngineVersion}" --output text
```

Sostituisci *major-engine-version*con la versione principale del motore e sostituisci la *regione* con Regione AWS. Ad esempio, il AWS CLI comando seguente restituisce la versione del motore secondario MySQL predefinita per la versione principale 5.7 e gli Stati Uniti occidentali (Oregon) (us-west-2): Regione AWS

```
aws rds describe-db-engine-versions --default-only --engine mysql --engine-version 5.7
 --region us-west-2 --query "*[].{Engine:Engine,EngineVersion:EngineVersion}" --output
 text
```

Con Amazon RDS, puoi controllare quando eseguire l'aggiornamento dell'istanza MySQL a una nuova versione supportata da Amazon RDS. Puoi mantenere la compatibilità con versioni specifiche di MySQL, testare le nuove versioni con l'applicazione prima di distribuirle in produzione e aggiornare le versioni quando è più appropriato in base alla tua pianificazione.

Quando l'aggiornamento automatico della versione secondaria è abilitato, l'istanza DB verrà aggiornata automaticamente alle nuove versioni secondarie di MySQL in quanto sono supportate da Amazon RDS. L'applicazione di patch avviene durante la finestra di manutenzione pianificata. È possibile modificare un'istanza DB per abilitare o disabilitare gli aggiornamenti automatici delle versioni secondarie.

Se annulli gli aggiornamenti automatici pianificati, puoi eseguire manualmente l'aggiornamento a una versione secondaria supportata seguendo la stessa procedura utilizzata per l'aggiornamento di una versione principale. Per informazioni, consulta Aggiornamento della versione del motore di un'istanza database.

Amazon RDS attualmente supporta gli aggiornamenti della versione principale da MySQL 5.6 alla versione 5.7 e da MySQL 5.7 alla versione 8.0. Poiché gli aggiornamenti della versione principale

prevedono alcuni rischi di compatibilità, non vengono eseguiti automaticamente, ma devi inviare una richiesta di modifica dell'istanza database. Testa in modo approfondito qualsiasi aggiornamento prima di applicarlo alle istanze di produzione. Per informazioni sull'aggiornamento di un'istanza database MySQL, consulta Aggiornamento del motore di database MySQL.

Puoi testare una nuova versione per un'istanza database prima di eseguire l'aggiornamento creando uno snapshot DB dell'istanza database esistente, eseguendo il ripristino dallo snapshot DB per creare una nuova istanza database e quindi avviando un aggiornamento della versione per la nuova istanza database. Potrai quindi effettuare le prove che desideri sul clone aggiornato dell'istanza database prima di decidere se aggiornare o meno l'istanza database originale.

Versioni principali di MySQL supportate in Amazon RDS

Il supporto standard delle versioni principali di RDS per MySQL resta disponibile almeno fino alla fine del ciclo di vita della community per la versione della community corrispondente. Puoi continuare a utilizzare una versione principale dopo la data di fine del supporto RDS standard a pagamento. Per ulteriori informazioni, consulta Utilizzo dell'estensione del supporto per Amazon RDS e Prezzi di Amazon RDS per MySQL.

È possibile utilizzare le date seguenti per pianificare i cicli di test e aggiornamento.

> **Note**
>
> Le date con solo un mese e un anno sono approssimative e vengono aggiornate con una data esatta quando nota.

Versione principale di MySQL	Data di rilascio nella community	Data di rilascio per RDS	Data di fine vita nella community	Data di fine del supporto standard RDS	Data di inizio validità dei prezzi per l'estensione del supporto RDS (1 anno)	Data di inizio validità dei prezzi per l'estensione del supporto RDS (3 anni)	Data di fine dell'estensione del supporto RDS
MySQL 8.0	19 aprile 2018	23 ottobre 2018	Aprile 2026	31 luglio 2026	1 agosto 2026	1 agosto 2028	31 luglio 2029
MySQL 5.7*	21 ottobre 2015	22 febbraio 2016	Ottobre 2023	29 febbraio 2024	1 marzo 2024	1 marzo 2026	28 febbraio 2027

* MySQL 5.7 è ora disponibile solo con RDS Extended Support. Per ulteriori informazioni, consulta Utilizzo dell'estensione del supporto per Amazon RDS.

Versioni Amazon RDS Extended Support per RDS per MySQL

Il contenuto seguente elenca tutte le versioni delle versioni di RDS Extended Support for RDS for MySQL.

Rilasci

- RDS Extended Support per RDS per MySQL versione 5.7.44-RDS.20240408

RDS Extended Support per RDS per MySQL versione 5.7.44-RDS.20240408

È disponibile RDS Extended Support for RDS for MySQL versione 5.7.44-RDS.20240408.

Questa versione contiene patch per i seguenti CVE:

- CVE-2024-20963

Utilizzo dell'ambiente di anteprima del database

Nel luglio 2023, Oracle ha annunciato un nuovo modello di rilascio per MySQL. Questo modello include due tipi di rilasci: di innovazione e LTS. Amazon RDS rende disponibili i rilasci di innovazione MySQL nell'ambiente di anteprima RDS. Per ulteriori informazioni sui rilasci di innovazione MySQL, consulta Introducing MySQL Innovation and Long-Term Support (LTS) versions.

Le istanze database RDS per MySQL nell'ambiente di anteprima del database sono funzionalmente simili alle altre istanze database RDS per MySQL. Non è tuttavia possibile utilizzare l'ambiente di anteprima del database per carichi di lavoro di produzione.

Gli ambienti di anteprima presentano le seguenti limitazioni:

- Amazon RDS elimina tutte le istanze database 60 giorni dopo la creazione, insieme a eventuali backup e snapshot.
- Puoi utilizzare solo lo storage General Purpose (SSD) e Provisioned IOPS (SSD).
- Non puoi ricevere assistenza con le istanze DB. AWS Support Puoi invece pubblicare le tue domande nella community di domande e risposte AWS gestita, re:POST.AWS
- Non puoi copiare uno snapshot di un'istanza database in un ambiente di produzione.

Le seguenti opzioni sono supportate dall'anteprima.

- È possibile creare istanze database utilizzando le classi di istanza database db.m6i, db.m6g, db.m5, db.t3, db.r6g e db.r5. Per ulteriori informazioni sulle classi delle istanze RDS, consulta Classi di istanze database.
- È possibile utilizzare distribuzioni AZ singola e Multi-AZ.
- È possibile utilizzare le funzioni di dump e caricamento standard di MySQL per esportare database da o importare database nell'ambiente di anteprima database.

Funzionalità non supportate nell'ambiente di anteprima del database

Le seguenti funzionalità non sono disponibili nell'ambiente di anteprima del database:

- Copia di snapshot tra regioni diverse
- Repliche di lettura tra regioni diverse

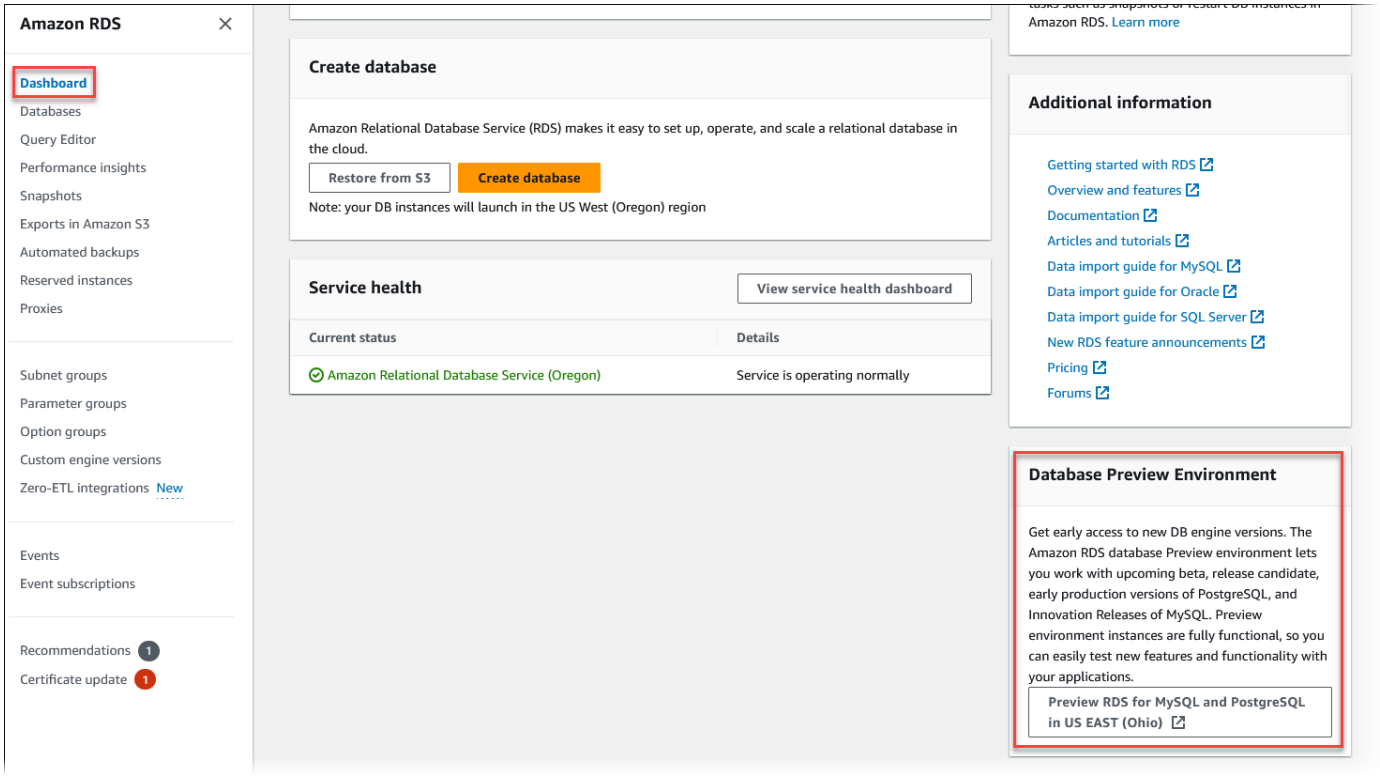
Creazione di una nuova istanza database nell'ambiente di anteprima del database

È possibile creare un'istanza DB nell'ambiente Database Preview utilizzando AWS Management Console, AWS CLI o l'API RDS.

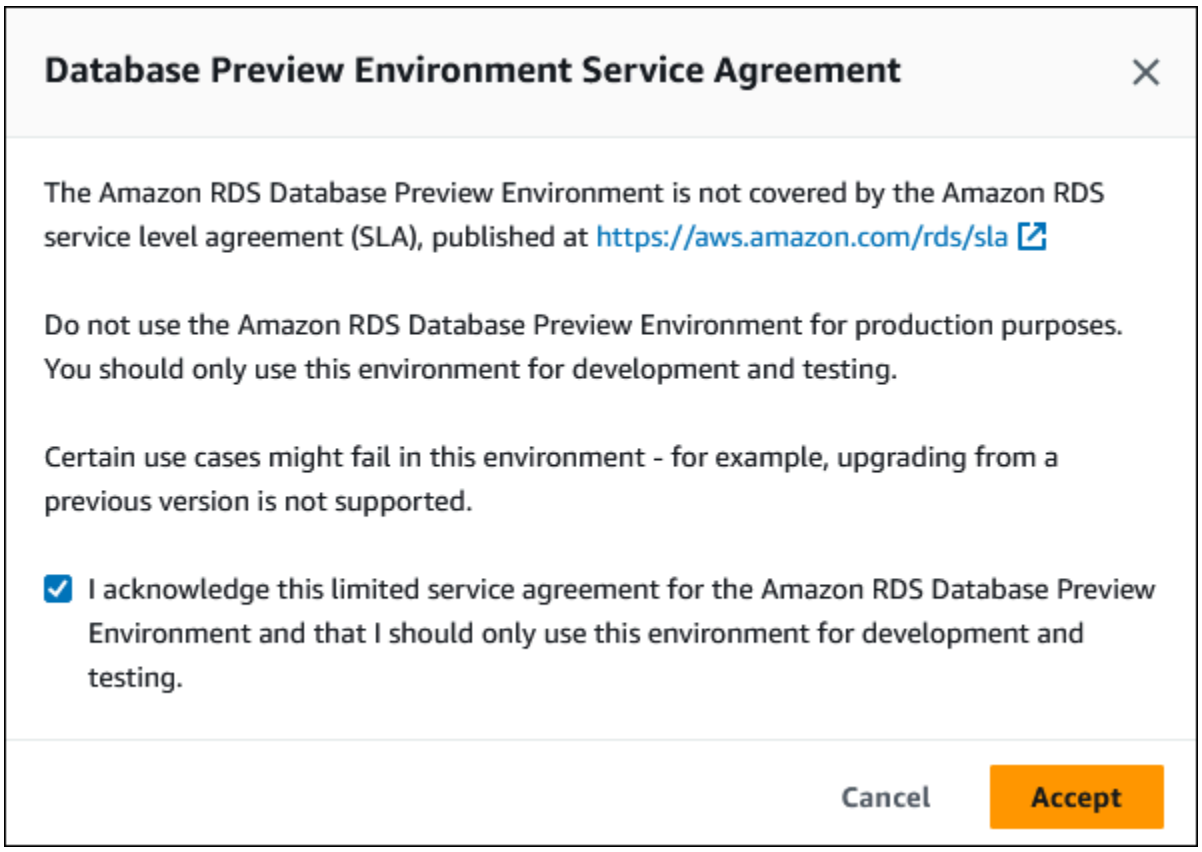
Console

Per creare un'istanza database nell'ambiente di anteprima del database

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Scegliere Dashboard (Pannello di controllo) nel pannello di navigazione.
3. Nella pagina Pannello di controllo individua la sezione Ambiente di anteprima del database, come mostrato nell'immagine seguente.



L'ambiente di anteprima del database è accessibile direttamente. Prima di poter procedere, è necessario capire e accettare le limitazioni.



4. Per creare l'istanza database RDS per MySQL, segui la stessa procedura utilizzata per creare qualsiasi istanza database Amazon RDS. Per ulteriori informazioni, consulta la procedura Console in Creazione di un'istanza database.

AWS CLI

Per creare un'istanza database nell'ambiente di anteprima del database utilizzando la AWS CLI, usa il seguente endpoint.

```
rds-preview.us-east-2.amazonaws.com
```

Per creare l'istanza database RDS per MySQL, segui la stessa procedura utilizzata per creare qualsiasi istanza database Amazon RDS. Per ulteriori informazioni, consulta la procedura AWS CLI in Creazione di un'istanza database.

API RDS

Per creare un'istanza database nell'ambiente di anteprima del database utilizzando l'API RDS, usa il seguente endpoint.

```
rds-preview.us-east-2.amazonaws.com
```

Per creare l'istanza database RDS per MySQL, segui la stessa procedura utilizzata per creare qualsiasi istanza database Amazon RDS. Per ulteriori informazioni, consulta la procedura API RDS in Creazione di un'istanza database.

MySQL versione 8.3 nell'ambiente Database Preview

La versione 8.3 di MySQL è ora disponibile nell'ambiente Amazon RDS Database Preview. La versione 8.3 di MySQL contiene diversi miglioramenti descritti in Modifiche a MySQL 8.3.0.

Per informazioni sull'ambiente di anteprima del database, consulta the section called “ Ambiente di anteprima del database”. Per accedere all'ambiente di anteprima dalla console, selezionare https://console.aws.amazon.com/rds-preview/.

MySQL versione 8.2 nell'ambiente Database Preview

La versione 8.2 di MySQL è ora disponibile nell'ambiente Amazon RDS Database Preview. La versione 8.2 di MySQL contiene diversi miglioramenti descritti in Modifiche a MySQL 8.2.0.

Per informazioni sull'ambiente di anteprima del database, consulta the section called “ Ambiente di anteprima del database”. Per accedere all'ambiente di anteprima dalla console, selezionare https://console.aws.amazon.com/rds-preview/.

MySQL versione 8.1 nell'ambiente di anteprima del database

MySQL versione 8.1 è ora disponibile nell'ambiente di anteprima del database Amazon RDS. MySQL versione 8.1 include vari miglioramenti, descritti in Changes in MySQL 8.1.0.

Per informazioni sull'ambiente di anteprima del database, consulta the section called “ Ambiente di anteprima del database”. Per accedere all'ambiente di anteprima dalla console, selezionare https://console.aws.amazon.com/rds-preview/.

Versioni obsolete per Amazon RDS for MySQL

Le versioni 5.1, 5.5 e 5.6 di Amazon RDS per MySQL sono obsoleti.

Per informazioni sulla policy di deprecazione di Amazon RDS for MySQL, consulta la pagina Domande frequenti su Amazon RDS.

Connessione a un'istanza database che esegue il motore di database di MySQL

Prima di eseguire la connessione a un'istanza database che esegue il motore di database di MySQL, devi creare un'istanza database. Per informazioni, consulta Creazione di un'istanza database Amazon RDS. Dopo che Amazon RDS ha fornito l'istanza database, puoi utilizzare una qualsiasi applicazione client o utilità MySQL standard per connetterti all'istanza. Nella stringa di connessione devi specificare l'indirizzo DNS dell'endpoint dell'istanza database come parametro host e specificare il numero di porta dell'endpoint dell'istanza database come parametro port.

Per autenticarti sulla tua istanza DB RDS, puoi utilizzare uno dei metodi di autenticazione per MySQL e AWS Identity and Access Management l'autenticazione del database (IAM):

- Per istruzioni su come eseguire l'autenticazione a MySQL utilizzando uno dei metodi di autenticazione per MySQL, consulta Metodo di autenticazione nella documentazione di MySQL.
- Per informazioni su come eseguire l'autenticazione a MySQL utilizzando l'autenticazione del database IAM, consulta Autenticazione del database IAM per MariaDB, MySQL e PostgreSQL.

Puoi eseguire la connessione a un'istanza database MySQL utilizzando strumenti come il cliente della riga di comando MySQL. Per ulteriori informazioni sull'utilizzo del client della riga di comando MySQL, vai alla sezione mysql - Il client della riga di comando di MySQL nella documentazione di MySQL. Un'applicazione basata su GUI che puoi utilizzare per la connessione è MySQL Workbench. Per ulteriori informazioni, consulta la pagina Download MySQL Workbench. Per informazioni sull'installazione di MySQL (compreso il client della riga di comando MySQL), consulta Installazione e aggiornamento di MySQL.

Per connettersi a un'istanza DB dall'esterno Amazon VPC, l'istanza DB deve essere accessibile pubblicamente, l'accesso deve essere concesso utilizzando le regole in ingresso del gruppo di sicurezza dell'istanza DB e devono essere soddisfatti altri requisiti. Per ulteriori informazioni, consulta Impossibile connettersi all'istanza database di Amazon RDS.

Puoi utilizzare la crittografia Secure Sockets Layer (SSL) o Transport Layer Security (TLS) sulle connessioni a un'istanza database MySQL. Per informazioni, consulta Utilizzo di SSL/TLS con un'istanza database MySQL. Se utilizzi l'autenticazione del database AWS Identity and Access Management (IAM), assicurati di utilizzare una connessione SSL/TLS. Per informazioni, consulta Autenticazione del database IAM per MariaDB, MySQL e PostgreSQL.

Puoi inoltre connetterti a un'istanza database da un server Web. Per ulteriori informazioni, consulta Tutorial: creazione di un server Web e un'istanza database Amazon RDS.

> **Note**
>
> Per informazioni sulla connessione a un'istanza database MariaDB, consulta Connessione a un'istanza database che esegue il motore di database MariaDB.

Indice

Ricerca delle informazioni di connessione per un'istanza DB RDS for MySQL

Le informazioni di connessione per un'istanza database includono l'endpoint, la porta e un utente di database valido, ad esempio l'utente master. Si supponga, ad esempio, che un valore endpoint sia `mydb.123456789012.us-east-1.rds.amazonaws.com`. In questo caso, il valore della porta è `3306` e l'utente del database è `admin`. Date queste informazioni, è possibile specificare i seguenti valori in una stringa di connessione:

- Per host, nome host o nome DNS, specifica `mydb.123456789012.us-east-1.rds.amazonaws.com`.
- Per la porta, specific `3306`.
- Per l'utente, specifica `admin`.

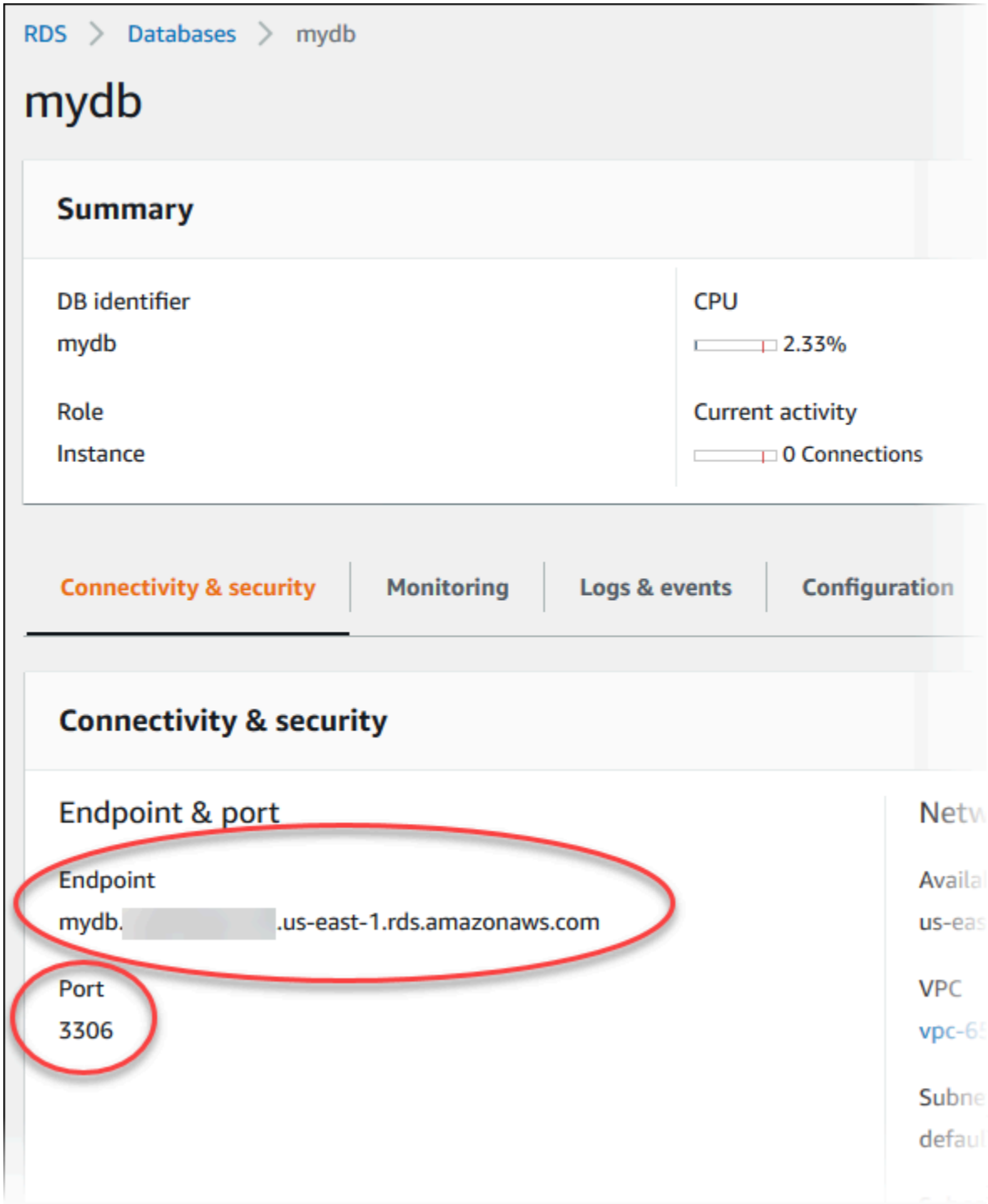
Per connettersi a un'istanza database, utilizzare qualsiasi client per un motore di database MySQL. Ad esempio, è possibile utilizzare il client a riga di comando MySQL o MySQL Workbench.

Per trovare le informazioni di connessione per un'istanza DB, puoi utilizzare il AWS CLI describe-db-instancescomando AWS Management Console, o l'operazione DescribedBInstances API di Amazon RDS per elencarne i dettagli.

Console

Per trovare le informazioni di connessione per un'istanza DB nel AWS Management Console

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di spostamento scegliere Database per visualizzare un elenco delle istanze database.
3. Scegliere il nome dell'istanza database MySQL per visualizzarne i dettagli.
4. Nella scheda Connectivity & security (Connettività e sicurezza), copiare l'endpoint. Annotare anche il numero di porta. L'endpoint e il numero di porta sono necessari per la connessione all'istanza database.



5. Se è necessario trovare il nome utente master, scegliere la scheda Configurazione e visualizzare il valore del nome utente principale .

AWS CLI

Per trovare le informazioni di connessione per un'istanza DB MySQL utilizzando AWS CLI il, chiamate il comando. describe-db-instances Nella chiamata, eseguire una query per l'ID istanza database, l'endpoint, la porta e il nome utente master.

PerLinux, omacOS: Unix

```
aws rds describe-db-instances \
  --filters "Name=engine,Values=mysql" \
  --query "*[].[DBInstanceIdentifier,Endpoint.Address,Endpoint.Port,MasterUsername]"
```

Per Windows:

```
aws rds describe-db-instances ^
  --filters "Name=engine,Values=mysql" ^
  --query "*[].[DBInstanceIdentifier,Endpoint.Address,Endpoint.Port,MasterUsername]"
```

L'output visualizzato dovrebbe essere simile al seguente.

```
[
    [
        "mydb1",
        "mydb1.123456789012.us-east-1.rds.amazonaws.com",
        3306,
        "admin"
    ],
    [
        "mydb2",
        "mydb2.123456789012.us-east-1.rds.amazonaws.com",
        3306,
        "admin"
    ]
]
```

API RDS

Per trovare le informazioni di connessione per un'istanza database utilizzando l'API Amazon RDS, richiamare l'operazione DescribeDBInstances. Nell'output, individuare i valori per l'indirizzo dell'endpoint, la porta dell'endpoint e il nome utente master.

Installazione del client da riga di comando MySQL

La maggior parte delle distribuzioni Linux include il client MariaDB invece del client Oracle MySQL. Per installare il client della linea di comando MySQL su Amazon Linux 2023, esegui il comando seguente:

```
sudo dnf install mariadb105
```

Per installare il client della linea di comando MySQL su Amazon Linux 2, esegui il comando seguente:

```
sudo yum install mariadb
```

Per installare il client della riga di comando MySQL sulla maggior parte delle distribuzioni Linux basate su DEB, emettere il comando seguente:

```
apt-get install mariadb-client
```

Per controllare la versione del client a riga di comando MySQL, emettere il seguente comando.

```
mysql --version
```

Per leggere la documentazione MySQL per la versione corrente del client, emettere il comando seguente:

```
man mysql
```

Connessione dal client a riga di comando MySQL (non crittografato)

> **Important**
>
> Utilizzare una connessione MySQL non crittografata solo quando il client e il server sono nello stesso VPC e la rete è attendibile. Per ulteriori informazioni sull'uso di connessioni crittografate, consulta Connessione dal client a riga di comando MySQL con SSL/TLS (crittografato).

Per connetterti a un'istanza database utilizzando il client della riga di comando MySQL, inserisci il seguente comando al prompt dei comandi. Per il parametro -h, sostituisci il nome DNS (endpoint) per la tua istanza database. Per il parametro -P, sostituisci la porta per la tua istanza database. Per il parametro -u, sostituire il nome utente di un utente di database valido, ad esempio l'utente master. Immetti la password dell'utente master quando richiesto.

```
mysql -h mysql-instance1.123456789012.us-east-1.rds.amazonaws.com -P 3306 -
u mymasteruser -p
```

Dopo aver immesso la password per l'utente, l'output dovrebbe essere analogo a quanto mostrato di seguito.

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9738
Server version: 8.0.28 Source distribution

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql>
```

Connessione da MySQL Workbench

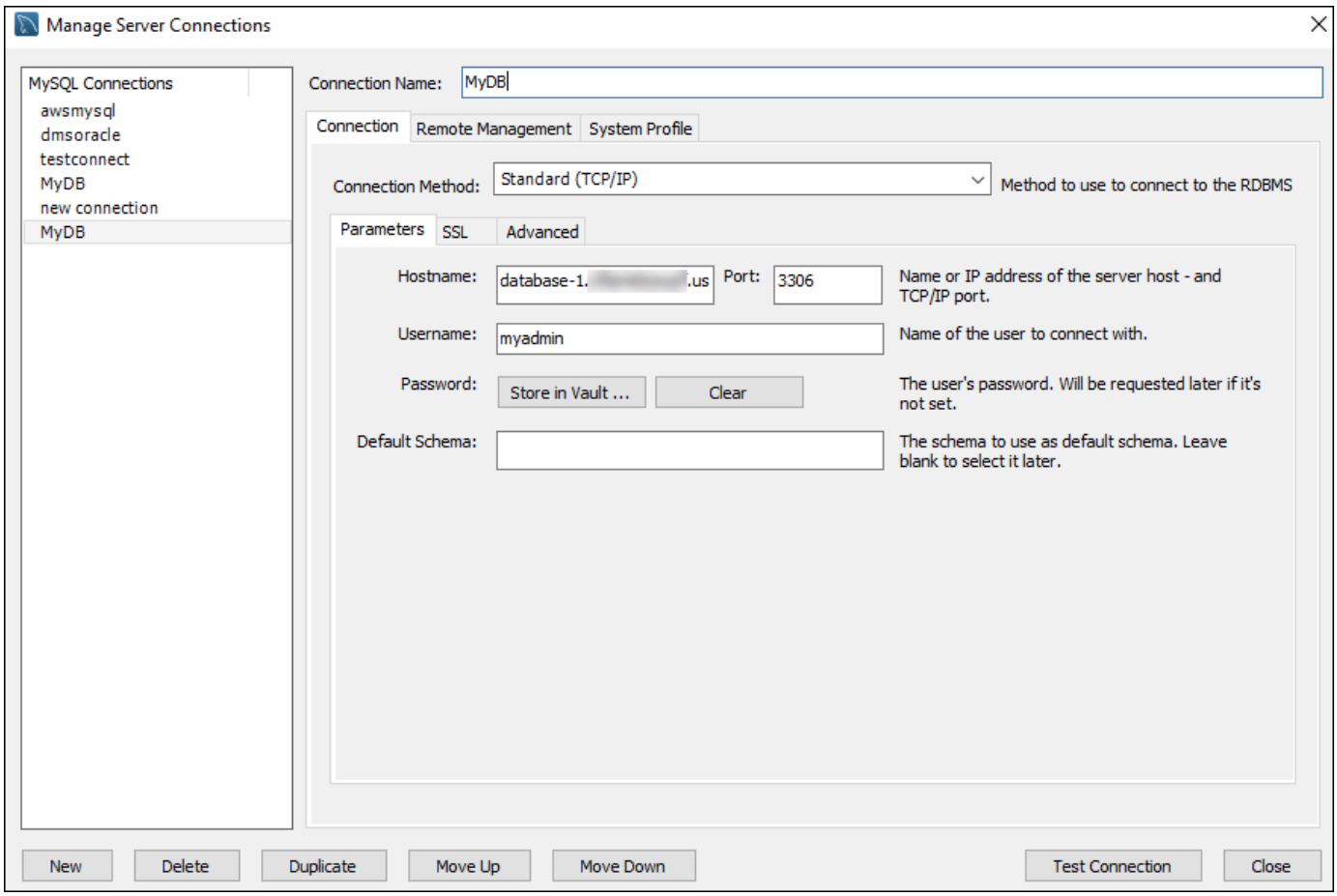
Per stabilire una connessione da MySQL Workbench

1. Scaricare e installare MySQL Workbench dalla pagina di download di MySQL Workbench.
2. Aprire MySQL Workbench.

3. From Database, scegliere Manage Connections (Gestisci connessioni).
4. Nella finestra Manage Server Connections (Gestisci connessioni al server), scegliere New (Nuova).
5. Nella finestra Connect to Database (Connetti a database), immettere le informazioni riportate di seguito:

   - Stored Connection (Connessione archiviata) – Immettere un nome per la connessione, ad esempio **MyDB**.
   - Hostname (Nome host) –Immettere l'endpoint dell'istanza database.
   - Port (Porta) – Immettere la porta usata dall'istanza database.
   - Nome utente – Immettere il nome utente di un utente del database valido, come l'utente master.
   - Password – Facoltativamente, scegliere Store in Vault (Archivia nel vault), quindi immettere e salvare la password per l'utente.

   La finestra è simile a quanto segue:



È possibile utilizzare le funzionalità di MySQL Workbench per personalizzare le connessioni. Ad esempio, puoi utilizzare la scheda SSL per configurare le connessioni SSL/TLS. Per informazioni sull'uso di MySQL Workbench, consulta la documentazione di MySQL Workbench. Crittografia delle connessioni client alle istanze database MySQL con SSL/TLS, consulta Crittografia delle connessioni client alle istanze database MySQL con SSL/TLS.

6. Facoltativamente, scegliere Test Connection (Verifica connessione) per confermare che la connessione all'istanza database è stata stabilita correttamente.
7. Scegli Chiudi.
8. Da Database, scegliere Connect to Database (Connetti al database).
9. Da Stored Connection (Connessione archiviata), scegliere la connessione.
10. Scegliere OK.

Connessione a RDS per MySQL con il driver JDBC Amazon Web Services ()AWS

Il driver JDBC di Amazon Web Services (AWS) è progettato come wrapper JDBC avanzato. Questo wrapper è complementare e amplia le funzionalità di un driver JDBC esistente. Il driver è compatibile direttamente con il driver MySQL Connector/J della community e il driver MariadB Connector/J della community.

Per installare il driver AWS JDBC, aggiungi il file.jar del driver AWS JDBC (che si trova nell'applicazione) e mantieni i riferimenti al rispettivo driver della community. `CLASSPATH` Aggiorna il rispettivo prefisso dell'URL di connessione come segue:

- `jdbc:mysql://` Da a `jdbc:aws-wrapper:mysql://`
- `jdbc:mariadb://` Da a `jdbc:aws-wrapper:mariadb://`

Per ulteriori informazioni sul driver AWS JDBC e istruzioni complete per il suo utilizzo, consulta l'archivio dei driver JDBC di Amazon Web Services (AWS). GitHub

Connessione a RDS per MySQL con il driver Python di Amazon Web Services ()AWS

Il driver Python di Amazon Web Services (AWS) è progettato come wrapper Python avanzato. Questo wrapper è complementare ed estende le funzionalità del driver open source Psycopg. Il

AWS Python Driver supporta le versioni Python 3.8 e successive. È possibile installare il `aws-advanced-python-wrapper` pacchetto utilizzando il `pip` comando, insieme ai pacchetti open source. `psycopg`

Per ulteriori informazioni sul driver AWS Python e istruzioni complete per il suo utilizzo, consulta il repository Amazon Web Services ()AWS Python Driver. GitHub

Risoluzione dei problemi relativi alle connessioni all'istanza database MySQL

Ecco due cause frequenti degli errori di connessione a una nuova istanza database:

- L'istanza database è stata creata tramite un gruppo di sicurezza che non autorizza le connessioni dal dispositivo o dall'istanza Amazon EC2 su cui è in esecuzione l'applicazione o l'utilità di MySQL. L'istanza database deve disporre di un gruppo di sicurezza VPC che autorizzi le connessioni. Per ulteriori informazioni, consulta VPC di Amazon VPC e Amazon RDS.

  Puoi aggiungere o modificare una regola in entrata nel gruppo di sicurezza: per Source (Origine), scegli My IP (Il mio IP). Questo consente l'accesso all'istanza database dall'indirizzo IP rilevato nel browser.
- L'istanza database è stata creata utilizzando la porta predefinita 3306 e nell'azienda vi sono regole del firewall che bloccano le connessioni a tale porta dai dispositivi nella rete aziendale. Per correggere l'errore, ricrea l'istanza con una porta diversa.

Per ulteriori informazioni sui problemi di connessione, consulta Impossibile connettersi all'istanza database di Amazon RDS.

Protezione delle connessioni di istanze database MySQL

Puoi gestire la sicurezza delle istanze database MySQL.

Argomenti

Sicurezza di MySQL in Amazon RDS

La sicurezza delle istanze database MySQL viene gestita su tre livelli:

- AWS Identity and Access Management controlla chi può eseguire azioni di gestione di Amazon RDS sulle istanze DB. Quando ti connetti AWS utilizzando le credenziali IAM, il tuo account IAM deve disporre di policy IAM che concedano le autorizzazioni necessarie per eseguire le operazioni di gestione di Amazon RDS. Per ulteriori informazioni, consulta Gestione accessi e identità per Amazon RDS.
- Quando crei un'istanza database, utilizzi un gruppo di sicurezza VPC per controllare i dispositivi e le istanze Amazon EC2 che possono aprire le connessioni all'endpoint e alla porta dell'istanza database. Queste connessioni possono essere stabilite tramite Secure Sockets Layer (SSL) e Transport Layer Security (TLS). Le regole del firewall aziendale possono inoltre determinare se i dispositivi in esecuzione nell'azienda possono aprire connessioni all'istanza database.
- Per autenticare l'accesso e le autorizzazioni per un'istanza database MySQL puoi seguire uno degli approcci riportati di seguito oppure utilizzare una loro combinazione.

  Puoi adottare lo stesso approccio utilizzato per un'istanza standalone di MySQL. I comandi come `CREATE USER`, `RENAME USER`, `GRANT`, `REVOKE` e `SET PASSWORD` funzionano esattamente come nei database in locale, modificando direttamente le tabelle dello schema del database. Tuttavia, la modifica diretta delle tabelle dello schema del database non è una best practice e, a partire dalla versione 8.0.36, non è supportata. Per ulteriori informazioni, consulta Access Control and Account Management nella documentazione MySQL.

Puoi anche utilizzare l'autenticazione database IAM. Questo metodo prevede l'autenticazione nell'istanza database tramite un utente IAM oppure con un ruolo IAM e un token di autenticazione. Il token di autenticazione è un valore univoco, generato tramite il processo di firma Signature Version 4. Utilizzando l'autenticazione del database IAM, puoi utilizzare le stesse credenziali per controllare l'accesso alle tue AWS risorse e ai tuoi database. Per ulteriori informazioni, consulta Autenticazione del database IAM per MariaDB, MySQL e PostgreSQL.

Un'altra opzione è l'autenticazione Kerberos per RDS per MySQL. L'istanza DB funziona con AWS Directory Service for Microsoft Active Directory (AWS Managed Microsoft AD) per abilitare l'autenticazione Kerberos. Quando gli utenti si autenticano con un'istanza database MySQL DB unita al dominio trusting, vengono inoltrate le richieste di autenticazione. Le richieste inoltrate vanno alla directory del dominio con cui crei. AWS Directory Service Per ulteriori informazioni, consulta Utilizzo dell'autenticazione Kerberos per MySQL.

Quando crei un'istanza database Amazon RDS, l'utente master ha i seguenti privilegi predefiniti:

Versione del motore	Privilegio del sistema	Ruolo di database
RDS per MySQL versione 8.0.36 e successive	`SELECT`, `INSERT`, `UPDATE`, `DELETE`, `CREATE`, `DROP`, `RELOAD`, `PROCESS`, `REFERENCES` , `INDEX`, `ALTER`, `SHOW DATABASES` , `CREATE TEMPORARY TABLES`, `LOCK TABLES`, `EXECUTE`, `REPLICATION SLAVE`, `REPLICATION CLIENT` , `CREATE VIEW`, `SHOW VIEW`, `CREATE ROUTINE`, `ALTER ROUTINE`, `CREATE USER`, `EVENT`, `TRIGGER`, `CREATE ROLE`, `DROP ROLE`, `APPLICATION_PASSWORD_ADMIN` , `ROLE_ADMIN` , `SET_USER_ID` , `XA_RECOVER_ADMIN`	`rds_superuser_role` Per ulteriori informazioni su `rds_superuser_role` , consulta Privilegio basato sui ruoli.
Versioni RDS per MySQL precedent	`SELECT`, `INSERT`, `UPDATE`, `DELETE`, `CREATE`, `DROP`, `RELOAD`, `PROCESS`, `REFERENCES` , `INDEX`, `ALTER`, `SHOW DATABASES` , `CREATE TEMPORARY TABLES`, `LOCK TABLES`, `EXECUTE`, `REPLICATION CLIENT` , `CREATE VIEW`, `SHOW VIEW`, `CREATE ROUTINE`,	—

Versione del motore	Privilegio del sistema	Ruolo di database
i alla 8.0.36	`ALTER ROUTINE`, `CREATE USER`, `EVENT`, `TRIGGER`, `REPLICATION SLAVE`	

> **Note**
>
> È possibile eliminare l'utente master nell'istanza database, ma non è consigliato farlo. Per ricreare l'utente master, utilizzate l'operazione API ModifyDBInstance RDS o il comando e specificate una nuova password per modify-db-instance AWS CLI l'utente principale con il parametro appropriato. Se l'utente master non è presente nell'istanza, viene creato con la password specificata.

Per fornire servizi di gestione per ogni istanza database, viene creato l'utente `rdsadmin` al momento della creazione dell'istanza database. I tentativi di rimuovere l'account `rdsadmin`, assegnargli un nuovo nome, modificarne la password o modificarne i privilegi genereranno un errore.

Per consentire la gestione dell'istanza database, i comandi standard `kill` e `kill_query` sono stati limitati. Vengono forniti i comandi Amazon RDS `rds_kill` e `rds_kill_query` per permettere di terminare le sessioni utente o le query nelle istanze database.

Utilizzo del plugin di convalida della password per RDS per MySQL

MySQL offre il plugin `validate_password` per l'ottimizzazione della sicurezza. Il plugin applica le policy delle password utilizzando i parametri nel gruppo di parametri DB per l'istanza database di MySQL. Il plugin è supportato per le istanze database che eseguono MySQL 5.7 e 8.0. Per ulteriori informazioni sul plugin `validate_password`, consulta The Password Validation Plugin nella documentazione di MySQL.

Per abilitare il plugin validate_password per un'istanza database di MySQL.

1. Connettiti all'istanza database di MySQL ed esegui questo comando.

```
INSTALL PLUGIN validate_password SONAME 'validate_password.so';
```

2. Puoi configurare i parametri per il plugin nel gruppo di parametri DB utilizzato dall'istanza database.

   Per ulteriori informazioni sui parametri, consulta Password Validation Plugin Options and Variables nella documentazione di MySQL.

   Per ulteriori informazioni sulla modifica dei parametri di un'istanza database, consulta Modifica di parametri in un gruppo di parametri del database.

Dopo aver installato e abilitato il plugin `password_validate`, reimposta le password esistenti affinché soddisfino le nuove policy di convalida.

Amazon RDS convalida le password. L'istanza database di MySQL esegue la convalida delle password. Se imposti una password utente con la AWS Management Console, il comando `modify-db-instance` AWS CLI o l'operazione API di RDS `ModifyDBInstance`, la modifica può essere apportata correttamente anche se la nuova password non soddisfa le policy delle password. Tuttavia, viene impostata una nuova password nell'istanza database di MySQL solo se questa soddisfa le policy delle password. In questo caso, Amazon RDS registra il seguente evento.

```
"RDS-EVENT-0067" - An attempt to reset the master password for the DB instance has
 failed.
```

Per ulteriori informazioni sugli eventi di Amazon RDS, consulta Utilizzo della notifica degli eventi di Amazon RDS.

Crittografia delle connessioni client alle istanze database MySQL con SSL/TLS

Secure Sockets Layer (SSL) è un protocollo standard del settore utilizzato per proteggere connessioni di rete tra client e server. Dopo SSL versione 3.0, il nome è stato modificato in Transport Layer Security (TLS). Amazon RDS supporta la crittografia SSL/TLS per le istanze database MySQL. Utilizzando SSL/TLS, puoi crittografare una connessione tra il client dell'applicazione e l'istanza database MariaDB. Il supporto SSL/TLS è disponibile in tutte le Regioni AWS per MySQL.

Argomenti

Utilizzo di SSL/TLS con un'istanza database MySQL

Amazon RDS crea un certificato SSL/TLS e installa il certificato nell'istanza database quando Amazon RDS effettua il provisioning dell'istanza. Questi certificati sono firmati da un'autorità di certificazione. Il certificato SSL/TLS include l'endpoint dell'istanza database come nome comune (CN) per il certificato SSL/TLS per la protezione contro attacchi di spoofing.

Un certificato SSL/TLS creato da Amazon RDS è l'entità root attendibile e funziona nella maggior parte dei casi, ma potrebbe non funzionare se l'applicazione non accetta catene di certificati. Se l'applicazione non accetta le catene di certificati, potrebbe essere necessario utilizzare un certificato intermedio per la connessione alla Regione AWS. Ad esempio, devi utilizzare un certificato intermedio per connetterti alle Regioni AWS GovCloud (US) tramite SSL/TLS.

Per ulteriori informazioni sul download dei certificati, consultare . Per ulteriori informazioni sull'uso di SSL/TLS con MySQL, consulta Aggiornamento delle applicazioni per la connessione a istanze database MySQL mediante nuovi certificati SSL/TLS.

MariaDB utilizza OpenSSL per connessioni sicure. Amazon RDS per MySQL supporta Transport Layer Security (TLS) versioni 1.0, 1.1, 1.2 e 1.3. Il supporto TLS dipende dalla versione MySQL. La tabella seguente mostra il supporto TLS per le versioni di MySQL.

MySQL versione	TLS 1.0	TLS 1.1	TLS 1.2	TLS 1.3
MySQL 8.0	Non supportato	Non supportato	Supportato	Supportato
MySQL 5.7	Supportato	Supportato	Supportato	Non supportato

Puoi richiedere le connessioni SSL/TLS per account utente specifici. Ad esempio, in base alla versione di MySQL, puoi utilizzare una delle seguenti istruzioni per richiedere connessioni SSL/TLS per l'account utente `encrypted_user`.

A tale scopo, utilizza la dichiarazione seguente.

```
ALTER USER 'encrypted_user'@'%' REQUIRE SSL;
```

Per ulteriori informazioni sulle connessioni SSL/TLS con MySQL, consulta Using encrypted connections nella documentazione di MySQL.

Richiesta di SSL/TLS per tutte le connessioni a un'istanza database MySQL

Puoi far sì che tutte le connessioni utente all'istanza database MySQL utilizzino SSL/TLS mediante il parametro `require_secure_transport`. Per impostazione predefinita, il parametro `require_secure_transport` è impostato su `OFF`. Puoi impostare il parametro `require_secure_transport` su `ON` per richiedere la crittografia SSL/TLS per le connessioni all'istanza database.

Puoi impostare il valore del parametro `require_secure_transport` aggiornando il gruppo di parametri database per l'istanza database. Non è necessario riavviare l'istanza database affinché la modifica abbia effetto.

Quando il parametro `require_secure_transport` è impostato su `ON` per un'istanza database, un client di database può connettersi a essa se è in grado di stabilire una connessione crittografata. In caso contrario, viene restituito al client un messaggio di errore simile al seguente:

```
MySQL Error 3159 (HY000): Connections using insecure transport are prohibited while --
require_secure_transport=ON.
```

Per informazioni sull'estensione dei parametri consulta Modifica di parametri in un gruppo di parametri del database.

Per ulteriori informazioni sul parametro `require_secure_transport`, consulta la documentazione di MySQL.

Connessione dal client a riga di comando MySQL con SSL/TLS (crittografato)

I parametri del programma client `mysql` sono leggermente diversi se si utilizza la versione MySQL 5.7, la versione MySQL 8.0 o la versione MariaDB.

Per scoprire quale versione è disponibile, esegui il comando `mysql` con l'opzione `--version`. Nell'esempio seguente, nell'output viene mostrato che il programma client proviene da MariaDB.

```
$ mysql --version
mysql  Ver 15.1 Distrib 10.5.15-MariaDB, for osx10.15 (x86_64) using readline 5.1
```

La maggior parte delle distribuzioni Linux, come Amazon Linux, CentOS, SUSE e Debian, hanno sostituito MySQL con MariaDB e la versione `mysql` presente proviene da MariaDB.

Per eseguire la connessione all'istanza database utilizzando SSL/TLS, segui questi passaggi:

Per eseguire la connessione a un'istanza database con SSL/TLS utilizzando il client a riga di comando MySQL

1. Scarica un certificato root che funziona per tutte le Regioni AWS.

   Per ulteriori informazioni sul download dei certificati, consultare .
2. Per stabilire la connessione a un'istanza database con la crittografia SSL/TLS, utilizza il client a riga di comando MySQL Per il parametro `-h`, sostituisci il nome DNS (endpoint) per l'istanza database. Per il parametro `--ssl-ca`, sostituisci il nome file del certificato SSL/TLS. Per il parametro `-P`, sostituisci la porta per l'istanza database. Per il parametro `-u`, sostituisci il nome utente di un utente di database valido, ad esempio l'utente master. Immetti la password dell'utente master quando richiesto.

   L'esempio seguente mostra come avviare il client utilizzando il parametro `--ssl-ca` per il client MySQL 5.7 o versioni successive.

   ```
   mysql -h mysql-instance1.123456789012.us-east-1.rds.amazonaws.com --ssl-ca=global-bundle.pem --ssl-mode=REQUIRED -P 3306 -u myadmin -p
   ```

   Per richiedere alla connessione SSL/TLS di verificare l'endpoint dell'istanza database confrontandolo con l'endpoint nel certificato SSL/TLS, immetti il seguente comando:

   ```
   mysql -h mysql-instance1.123456789012.us-east-1.rds.amazonaws.com --ssl-ca=global-bundle.pem --ssl-mode=VERIFY_IDENTITY -P 3306 -u myadmin -p
   ```

   L'esempio seguente mostra come avviare il client utilizzando il parametro `--ssl-ca` con il client MariaDB.

   ```
   mysql -h mysql-instance1.123456789012.us-east-1.rds.amazonaws.com --ssl-ca=global-bundle.pem --ssl -P 3306 -u myadmin -p
   ```
3. Immetti la password dell'utente master quando richiesto.

Verrà visualizzato un output simile al seguente.

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9738
Server version: 8.0.28 Source distribution

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql>
```

Aggiornamento delle applicazioni per la connessione a istanze database MySQL mediante nuovi certificati SSL/TLS

A partire dal 13 gennaio 2023, Amazon RDS ha pubblicato nuovi certificati dell'autorità di certificazione (CA) per la connessione alle istanze database RDS utilizzando Secure Socket Layer o Transport Layer Security (SSL/TLS). Di seguito sono disponibili le informazioni sull'aggiornamento delle applicazioni per utilizzare i nuovi certificati.

Questo argomento aiuta a determinare se le applicazioni client utilizzano SSL/TLS per connettersi alle istanze database. In caso affermativo, puoi determinare anche se le applicazioni richiedono la verifica del certificato per la connessione.

> **Note**
>
> Alcune applicazioni sono configurate per connettersi ai cluster DB MySQL solo se sono in grado di verificare il certificato del server. Per queste applicazioni, è necessario aggiornare gli archivi di trust delle applicazioni client per includere i nuovi certificati CA.
> Puoi specificare le seguenti modalità SSL: `disabled`, `preferred` e `required`. Quando si utilizza la modalità SSL `preferred` e il certificato CA non esiste o non è aggiornato, la connessione non utilizza SSL e continua a connettersi senza crittografia.
> Poiché queste versioni successive utilizzano il protocollo OpenSSL, un certificato server scaduto non impedisce che le connessioni vadano a buon fine, a meno che non venga specificata la modalità SSL `required`.
> Consigliamo di evitare la modalità `preferred`. In modalità `preferred`, se la connessione rileva un certificato non valido, interrompe l'utilizzo della crittografia e procede in modo non crittografato.

Dopo aver aggiornato i certificati CA negli archivi di trust delle applicazioni client, puoi ruotare i certificati nelle istanze database. Consigliamo vivamente di testare queste procedure in un ambiente di sviluppo o di gestione temporanea prima di implementarle negli ambienti di produzione.

Per ulteriori informazioni sulla rotazione dei certificati, consulta Rotazione del certificato SSL/TLS. Per ulteriori informazioni sul download, consulta . Per informazioni sull'utilizzo di SSL/TLS con le istanze database MySQL, consulta Utilizzo di SSL/TLS con un'istanza database MySQL.

Argomenti

- Determinare se un'applicazione si connette all'istanza database MySQL mediante SSL
- Determinare se un client richiede la verifica del certificato per la connessione
- Aggiornare l'archivio di trust delle applicazioni
- Codice Java di esempio per stabilire connessioni SSL

Determinare se un'applicazione si connette all'istanza database MySQL mediante SSL

Se utilizzi Amazon RDS per MySQL versione 5.7 o 8.0 e lo schema delle prestazioni è abilitato, esegui la query indicata di seguito per verificare se le connessioni utilizzano SSL/TLS. Per informazioni sull'abilitazione dello schema delle prestazioni, consulta l'argomento relativo alla guida rapida per lo schema delle prestazioni nella documentazione di MySQL.

```
mysql> SELECT id, user, host, connection_type
       FROM performance_schema.threads pst
       INNER JOIN information_schema.processlist isp
       ON pst.processlist_id = isp.id;
```

In questo output di esempio, puoi vedere che la tua sessione (`admin`) e un'applicazione collegata come `webapp1` stanno entrambe usando SSL.

```
+----+-----------------+-----------------+-----------------+
| id | user            | host            | connection_type |
+----+-----------------+-----------------+-----------------+
|  8 | admin           | 10.0.4.249:42590 | SSL/TLS         |
|  4 | event_scheduler | localhost       | NULL            |
| 10 | webapp1         | 159.28.1.1:42189 | SSL/TLS         |
+----+-----------------+-----------------+-----------------+
```

```
3 rows in set (0.00 sec)
```

Determinare se un client richiede la verifica del certificato per la connessione

Puoi verificare se i client JDBC e MySQL richiedono la verifica del certificato per la connessione.

JDBC

L'esempio seguente con MySQL Connector/J 8.0 mostra un modo per verificare le proprietà della connessione JDBC di un'applicazione per determinare se le connessioni riuscite richiedono un certificato valido. Per ulteriori informazioni su tutte le opzioni di connessione JDBC per MySQL, consulta l'argomento relativo alle proprietà di configurazione nella documentazione di MySQL.

Quando utilizzi MySQL Connector/J 8.0, la connessione SSL richiede la verifica del certificato CA del server se nelle proprietà di connessione `sslMode` è impostato su `VERIFY_CA` o `VERIFY_IDENTITY`, come nell'esempio seguente.

```
Properties properties = new Properties();
properties.setProperty("sslMode", "VERIFY_IDENTITY");
properties.put("user", DB_USER);
properties.put("password", DB_PASSWORD);
```

> **Note**
>
> Se si utilizza MySQL Java Connector v5.1.38 o versione successiva o MySQL Java Connector v8.0.9 o versione successiva per connettersi ai database, anche se non sono state configurate esplicitamente le applicazioni per l'utilizzo di SSL/TLS durante la connessione ai database, questi driver client utilizzano automaticamente SSL/TLS. Inoltre, quando utilizzano SSL/TLS, eseguono la verifica parziale del certificato e non riescono a connettersi se il certificato del server di database è scaduto.

MySQL

I seguenti esempi con il client MySQL mostrano due modi per verificare la connessione MySQL di uno script per determinare se le connessioni riuscite richiedono un certificato valido. Per ulteriori

informazioni su tutte le opzioni di connessione con il client MySQL, consulta l'argomento relativo alla configurazione lato client delle connessioni crittografate nella documentazione di MySQL.

Quando utilizzi il client MySQL 5.7 o MySQL 8.0, la connessione SSL richiede la verifica del certificato CA del server se per l'opzione `--ssl-mode` viene specificato `VERIFY_CA` o `VERIFY_IDENTITY`, come nell'esempio seguente.

```
mysql -h mysql-database.rds.amazonaws.com -uadmin -ppassword --ssl-ca=/tmp/ssl-cert.pem
 --ssl-mode=VERIFY_CA
```

Quando utilizzi il client MySQL 5.6, la connessione SSL richiede la verifica del certificato CA del server se viene specificata l'opzione `--ssl-verify-server-cert`, come nell'esempio seguente.

```
mysql -h mysql-database.rds.amazonaws.com -uadmin -ppassword --ssl-ca=/tmp/ssl-cert.pem
 --ssl-verify-server-cert
```

Aggiornare l'archivio di trust delle applicazioni

Per informazioni sull'aggiornamento dell'archivio attendibilità per le applicazioni MySQL, consulta l'argomento relativo all'installazione dei certificati SSL nella documentazione di MySQL.

Per ulteriori informazioni sul download del certificato root, consulta .

Per gli script di esempio che importano i certificati, consulta Script di esempio per l'importazione di certificati nel tuo archivio di trust.

> **Note**
>
> Quando aggiorni l'archivio di trust puoi conservare i certificati meno recenti oltre ad aggiungere i nuovi certificati.

Se utilizzi il driver mysql JDBC in un'applicazione, imposta le seguenti proprietà nell'applicazione.

```
System.setProperty("javax.net.ssl.trustStore", certs);
System.setProperty("javax.net.ssl.trustStorePassword", "password");
```

Quando avvii l'applicazione, imposta le seguenti proprietà.

```
java -Djavax.net.ssl.trustStore=/path_to_truststore/MyTruststore.jks -
Djavax.net.ssl.trustStorePassword=my_truststore_password com.companyName.MyApplication
```

> **Note**
>
> Specifica una password diversa dal prompt mostrato qui come best practice per la sicurezza.

Codice Java di esempio per stabilire connessioni SSL

L'esempio di codice seguente mostra come configurare la connessione SSL che convalida il certificato del server utilizzando JDBC.

```
public class MySQLSSLTest {

        private static final String DB_USER = "username";
        private static final String DB_PASSWORD = "password";
        // This key store has only the prod root ca.
        private static final String KEY_STORE_FILE_PATH = "file-path-to-keystore";
        private static final String KEY_STORE_PASS = "keystore-password";

        public static void test(String[] args) throws Exception {
            Class.forName("com.mysql.jdbc.Driver");

            System.setProperty("javax.net.ssl.trustStore", KEY_STORE_FILE_PATH);
            System.setProperty("javax.net.ssl.trustStorePassword", KEY_STORE_PASS);

            Properties properties = new Properties();
            properties.setProperty("sslMode", "VERIFY_IDENTITY");
            properties.put("user", DB_USER);
            properties.put("password", DB_PASSWORD);
```

```
            Connection connection = null;
            Statement stmt = null;
            ResultSet rs = null;
            try {
                connection =
 DriverManager.getConnection("jdbc:mysql://mydatabase.123456789012.us-
east-1.rds.amazonaws.com:3306",properties);
                stmt = connection.createStatement();
                rs=stmt.executeQuery("SELECT 1 from dual");
            } finally {
                if (rs != null) {
                    try {
                        rs.close();
                    } catch (SQLException e) {
                    }
                }
                if (stmt != null) {
                   try {
                        stmt.close();
                    } catch (SQLException e) {
                   }
                }
                if (connection != null) {
                    try {
                        connection.close();
                    } catch (SQLException e) {
                        e.printStackTrace();
                    }
                }
            }
            return;
        }
    }
```

Important

Dopo aver stabilito che le connessioni al database utilizzano SSL/TLS e aver aggiornato l'archivio attendibile dell'applicazione, è possibile aggiornare il database per utilizzare i certificati 2048-g1. rds-ca-rsa Per istruzioni, consulta la fase 3 in Aggiornamento del certificato CA modificando l'istanza o il cluster di database.

Specifica una password diversa dal prompt mostrato qui come best practice per la sicurezza.

Utilizzo dell'autenticazione Kerberos per MySQL

Puoi utilizzare Autenticazione Kerberos per autenticare gli utenti quando si connettono all'istanza database MySQL. L'istanza DB funziona con AWS Directory Service for Microsoft Active Directory (AWS Managed Microsoft AD) per abilitare l'autenticazione Kerberos. Quando gli utenti si autenticano con un'istanza database MySQL DB unita al dominio trusting, vengono inoltrate le richieste di autenticazione. Le richieste inoltrate vanno alla directory del dominio con cui crei. AWS Directory Service

Mantenere tutte le credenziali nella stessa directory consente di ridurre il tempo e l'impegno. Con questo approccio, è disponibile una posizione centralizzata per archiviare e gestire le credenziali per più istanze database. L'uso di una directory può inoltre migliorare il profilo di sicurezza complessivo.

Disponibilità di regioni e versioni

Il supporto varia a seconda delle versioni specifiche di ciascun motore di database e a seconda delle Regioni AWS. Per ulteriori informazioni sulla disponibilità di versioni e regioni per Amazon RDS con autenticazione Kerberos, consulta Regioni e motori DB supportati per l'autenticazione Kerberos in Amazon RDS.

Panoramica della configurazione dell'autenticazione Kerberos per istanze database MySQL

Per configurare l'autenticazione Kerberos per un'istanza database MySQL, completa le seguenti fasi generali, descritte in dettaglio più avanti:

1. Usa AWS Managed Microsoft AD per creare una AWS Managed Microsoft AD directory. È possibile utilizzare il AWS Management Console AWS CLI, il o il AWS Directory Service per creare la directory. Per ulteriori informazioni su questa operazione, consulta Creare la AWS Managed Microsoft AD directory nella Guida all'AWS Directory Service amministrazione.
2. Crea un ruolo AWS Identity and Access Management (IAM) che utilizzi la policy IAM gestita`AmazonRDSDirectoryServiceAccess`. Il ruolo consente ad Amazon RDS di effettuare chiamate alla tua directory.

   Affinché il ruolo consenta l'accesso, l'endpoint AWS Security Token Service (AWS STS) deve essere attivato nel campo Regione AWS per il tuo AWS account. AWS STS Gli endpoint sono tutti

Regioni AWS attivi per impostazione predefinita e puoi utilizzarli senza ulteriori azioni. Per ulteriori informazioni, consulta Attivazione e disattivazione AWS STSRegione AWS in un capitolo della IAM User Guide.

3. Crea e configura gli utenti nella AWS Managed Microsoft AD directory utilizzando gli strumenti di Microsoft Active Directory. Per ulteriori informazioni sulla creazione di utenti in Active Directory, vedere Gestire utenti e gruppi in Microsoft AD AWS gestito nella Guida all'AWS Directory Service amministrazione.
4. Creazione o modifica di un'istanza database MySQL. Se si utilizza l'interfaccia a riga di comando (CLI) o l'API RDS nella richiesta di creazione, specificare un identificatore di dominio con il parametro `Domain`. Utilizzare l'identificatore `d-*` generato al momento della creazione della directory e il nome del ruolo creato.

   Se si modifica un'istanza database MySQL esistente per utilizzare l'autenticazione Kerberos, impostare i parametri di dominio e ruolo IAM per l'istanza database. Individuare l'istanza database nello stesso VPC della directory di dominio.
5. Utilizza le credenziali dell'utente master Amazon RDS per connetterti all'istanza database MySQL. Crea l'utente in MySQL utilizzando la clausola `CREATE USER IDENTIFIED WITH 'auth_pam'`. Gli utenti creati in questo modo possono accedere all'istanza database MySQL utilizzando l'autenticazione Kerberos.

Configurazione dell'autenticazione Kerberos per istanze database MySQL

Si utilizza AWS Managed Microsoft AD per configurare l'autenticazione Kerberos per un'istanza DB MySQL. Per configurare l'autenticazione Kerberos, completa la procedura seguente.

Passaggio 1: creare una directory utilizzando AWS Managed Microsoft AD

AWS Directory Service crea una Active Directory completamente gestita nel AWS cloud. Quando crei una AWS Managed Microsoft AD directory, AWS Directory Service crea due controller di dominio e server DNS (Domain Name System) per tuo conto. I server di directory vengono creati in sottoreti diverse in un VPC. Questa ridondanza assicura che la directory rimanga accessibile anche se si verifica un errore.

Quando crei una AWS Managed Microsoft AD directory, AWS Directory Service esegue le seguenti attività per tuo conto:

- Configura una Active Directory all'interno del VPC.

- Crea un account amministratore della directory con nome utente Admin e la password specificata. Puoi utilizzare questo account per gestire le directory.

> **Note**
>
> Assicurati di salvare questa password. AWS Directory Service non la memorizza. È possibile reimpostarla ma non recuperarla.

- Crea un gruppo di sicurezza per i controller della directory.

Quando si avvia un AWS Managed Microsoft AD, AWS crea un'unità organizzativa (OU) che contiene tutti gli oggetti della directory. Questa unità organizzativa ha lo stesso nome NetBIOS che hai digitato al momento della creazione della directory e si trova nella radice del dominio. La radice del dominio è di proprietà e gestita da AWS.

L'account amministratore creato con la AWS Managed Microsoft AD directory dispone delle autorizzazioni per le attività amministrative più comuni dell'unità organizzativa:

- Creazione, aggiornamento o eliminazione di utenti
- Aggiungi risorse al dominio, come file server o server di stampa, e assegna le autorizzazioni per tali risorse a utenti dell'unità organizzativa
- Creazione di unità organizzative e container aggiuntivi
- Delega dell'autorità
- Ripristino degli oggetti eliminati dal cestino di Active Directory
- Esegui i PowerShell moduli Windows AD e DNS sul servizio Web Active Directory

L'account Admin dispone inoltre dei diritti per eseguire le seguenti attività estese a tutto il dominio:

- gestione delle configurazioni DNS (aggiunta, eliminazione o aggiornamento di record, zone e server d'inoltro);
- visualizzazione di log di eventi DNS;
- visualizzazione di log di eventi di sicurezza.

Per creare una directory con AWS Managed Microsoft AD

1. Accedere AWS Management Console e aprire la AWS Directory Service console all'indirizzo https://console.aws.amazon.com/directoryservicev2/.
2. Nel riquadro di navigazione, seleziona Directories (Directory) e quindi Set up directory (Configura la directory).
3. Scegliete AWS Managed Microsoft AD. AWS Managed Microsoft AD è l'unica opzione attualmente utilizzabile con Amazon RDS.
4. Immettere le seguenti informazioni:

   Nome DNS directory

   Il nome completo della directory, ad esempio **`corp.example.com`**.

   Nome NetBIOS della directory

   Nome breve per la directory, ad esempio **`CORP`**.

   Descrizione della directory

   (Opzionale) Una descrizione della directory.

   Password amministratore

   La password dell'amministratore della directory. Con il processo di creazione della directory viene generato un account amministratore con il nome utente Admin e questa password.

   La password dell'amministratore della directory e non può includere il termine "admin". La password distingue tra maiuscole e minuscole e la lunghezza deve essere compresa tra 8 e 64 caratteri. Deve anche contenere un carattere di almeno tre delle seguenti quattro categorie:

   - Lettere minuscole (a–z)
   - Lettere maiuscole (A–Z)
   - Numeri (0–9)
   - Caratteri non alfanumerici (~!@#$%^&*_-+=`|\(){}[]:;"'<>,.?/)

   Confirm password (Conferma password)

   La password dell'amministratore digitata nuovamente.

5. Seleziona Successivo.

6. Immettere le seguenti informazioni nella sezione Networking (Rete) e quindi scegliere Next (Avanti):

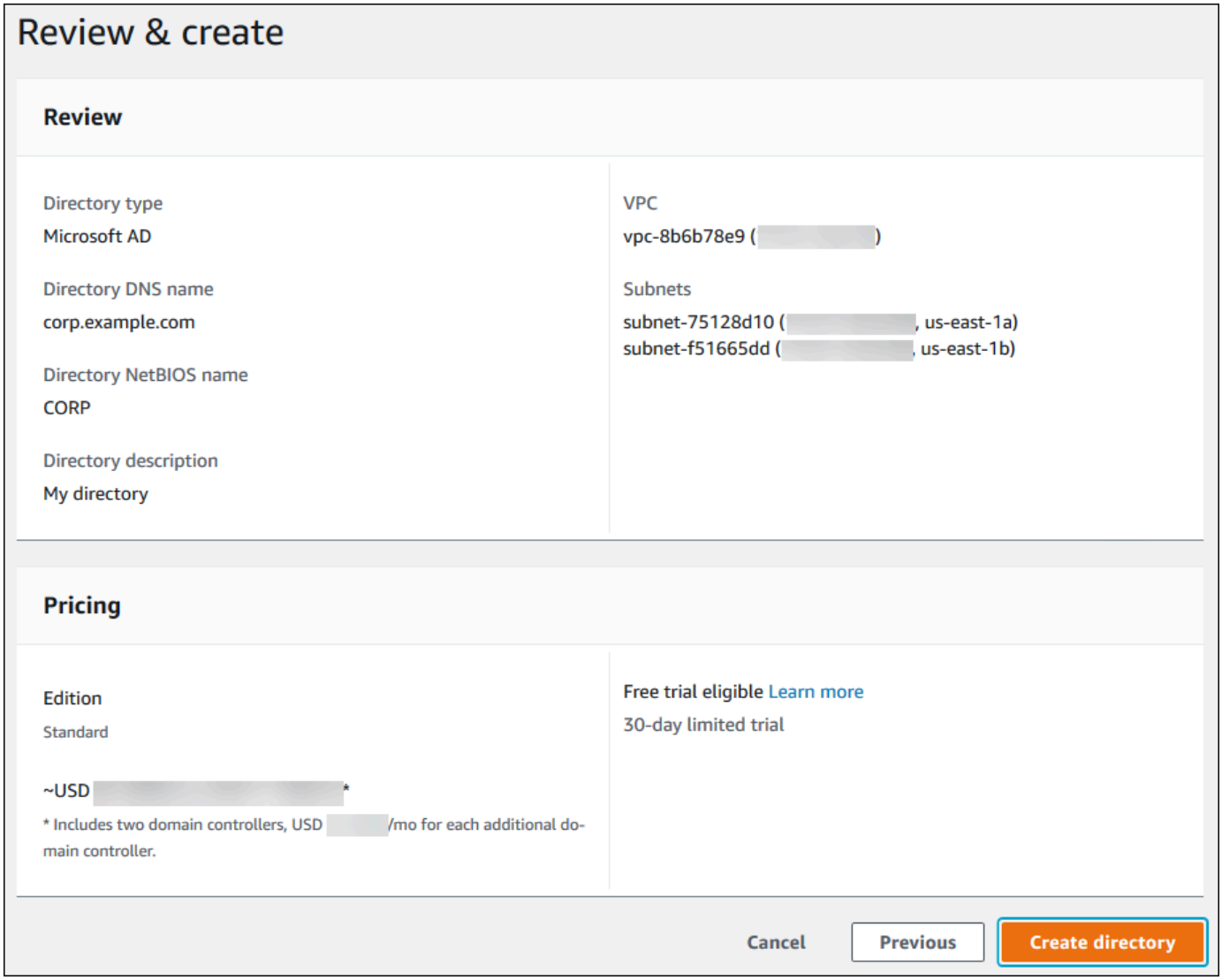
   VPC

   VPC per la directory. Crea l'istanza database MySQL in questo stesso VPC.

   Sottoreti

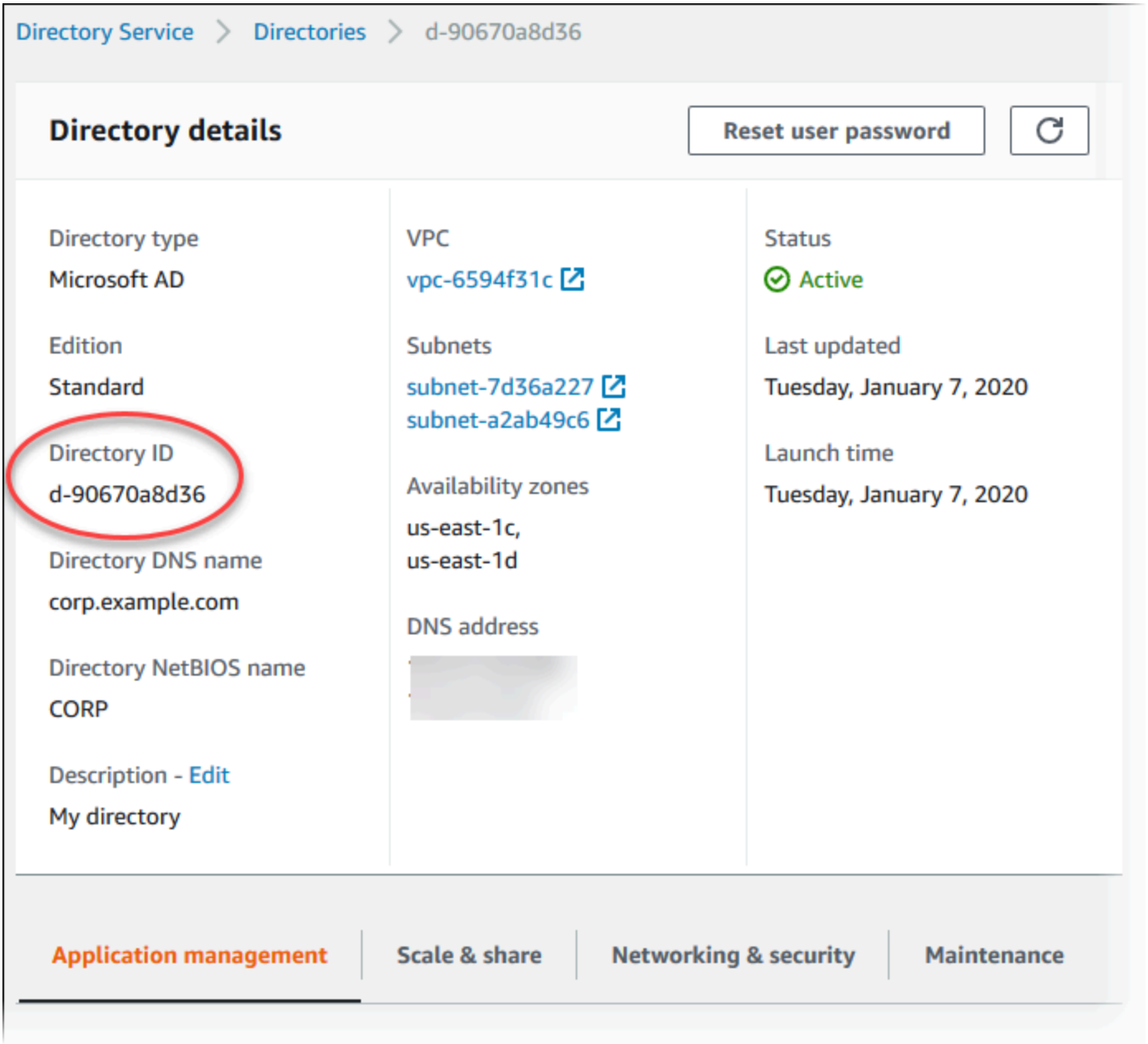
   Sottoreti per i server di directory. Le due sottoreti devono trovarsi in diverse zone di disponibilità.

7. Esaminare le informazioni relative alla directory e apportare eventuali modifiche. Quando le informazioni sono corrette, scegli Create Directory (Crea directory).



Per creare la directory sono necessari alcuni minuti. Una volta creata correttamente la directory, il valore Status (Stato) viene modificato in Active (Attivo).

Per consultare le informazioni sulla directory, selezionare il nome della directory nell'elenco di directory. Prendere nota del valore di Directory ID (ID directory) perché sarà necessario quando si crea o si modifica l'istanza database MySQL.



Fase 2: creazione del ruolo IAM per l'utilizzo da parte di Amazon RDS

Affinché Amazon RDS possa AWS Directory Service chiamarti, è necessario un ruolo IAM che utilizzi la policy `AmazonRDSDirectoryServiceAccess` IAM gestita. Questo ruolo permette ad Amazon RDS di effettuare chiamate alla AWS Directory Service.

Quando un'istanza DB viene creata utilizzando AWS Management Console e l'utente della console dispone dell'`iam:CreateRole`autorizzazione, la console crea questo ruolo automaticamente. In questo caso, il nome del ruolo è `rds-directoryservice-kerberos-access-role`. In caso contrario, è necessario creare manualmente il ruolo IAM. Quando crei questo ruolo IAM`Directory Service`, scegli e allega la policy AWS gestita `AmazonRDSDirectoryServiceAccess` ad esso.

Per ulteriori informazioni sulla creazione di ruoli IAM per un servizio, consulta Creating a role to delegate permissions to an AWS service nella IAM User Guide.

> **Note**
>
> Il ruolo IAM utilizzato per l'autenticazione Windows per RDS per SQL Server non può essere utilizzato per RDS per MySQL.

Facoltativamente, puoi creare policy con le autorizzazioni richieste anziché utilizzare la policy IAM gestita `AmazonRDSDirectoryServiceAccess`. In questo caso, il ruolo IAM deve avere la seguente policy di attendibilità IAM.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "directoryservice.rds.amazonaws.com",
          "rds.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
```

```
}
```

Il ruolo deve anche disporre della seguente policy del ruolo IAM.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "ds:DescribeDirectories",
        "ds:AuthorizeApplication",
        "ds:UnauthorizeApplication",
        "ds:GetAuthorizedApplicationDetails"
      ],
    "Effect": "Allow",
    "Resource": "*"
    }
  ]
}
```

Fase 3: creazione e configurazione di utenti

È possibile creare utenti con lo strumento Utenti Active Directory e computer. Questo strumento fa parte degli strumenti Active Directory Domain Services e Active Directory Lightweight Directory Services. Gli utenti possono essere individui singoli o entità che hanno accesso alla tua directory.

Per creare utenti in una AWS Directory Service directory, devi essere connesso a un'istanza Amazon EC2 basata su Microsoft Windows. Questa istanza deve essere un membro della AWS Directory Service directory ed essere connessa come utente con privilegi per creare utenti. Per ulteriori informazioni, consulta Gestione di utenti e gruppi in AWS Managed Microsoft AD nella AWS Directory Service - Guida di amministrazione.

Fase 4: creazione o modifica di un'istanza database MySQL

Creazione o modifica di un'istanza database MySQL per l'utilizzo con la directory. Puoi utilizzare la console, CLI o l'API RDS per associare un'istanza database a una directory. Questa operazione può essere eseguita in uno dei seguenti modi:

- Crea una nuova istanza DB MySQL utilizzando la console, il comando create-db-instanceCLI o l'operazione API CreateDBInstance RDS.

  Per istruzioni, consulta Creazione di un'istanza database Amazon RDS.

- Modifica un'istanza DB MySQL esistente utilizzando la console, il comando modify-db-instanceCLI o l'operazione API ModifyDBInstance RDS.

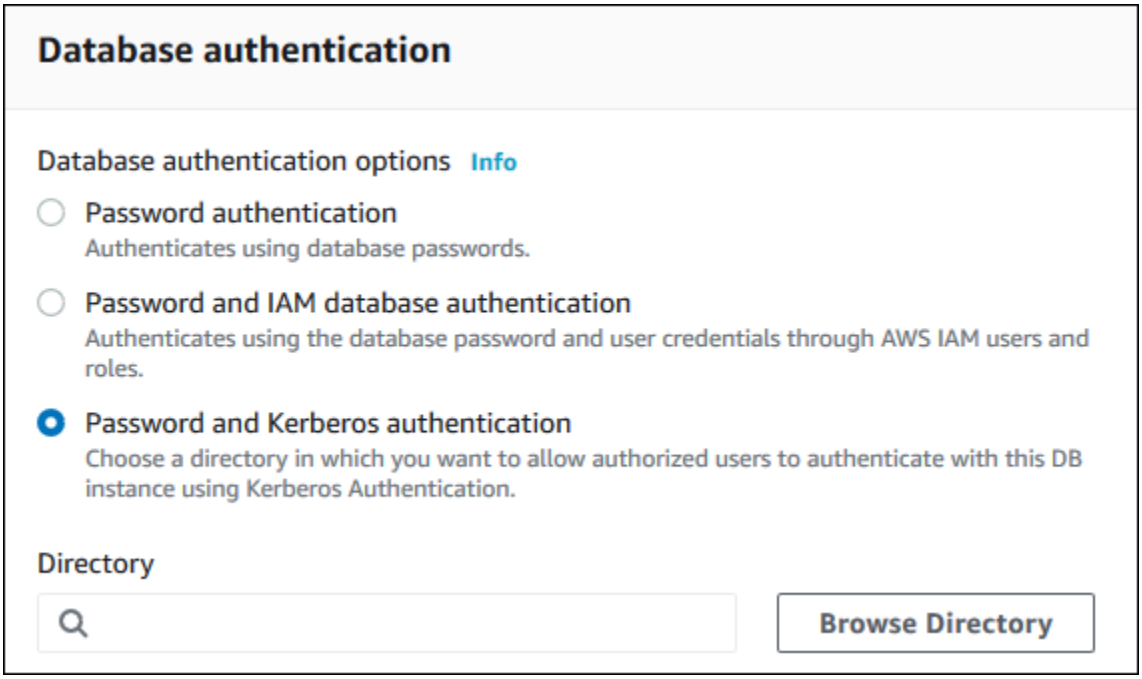
  Per istruzioni, consulta Modifica di un'istanza database Amazon RDS.
- Ripristina un'istanza DB MySQL da un'istantanea del DB utilizzando la console, il comando CLI _restore-db-instance-from-db-snapshot o l'operazione API RestoreDB DBSnapshot RDS. InstanceFrom

  Per istruzioni, consulta Ripristino da uno snapshot database.
- Ripristina un'istanza DB MySQL utilizzando la console, il comando _restore-db-instance-to- point-in-time CLI o l'operazione RestoreDB RDS API. point-in-time InstanceToPointInTime

  Per istruzioni, consulta Ripristino a un'ora specifica per un'istanza database.

L'autenticazione Kerberos è supportata solo per istanze database MySQL in un VPC. L'istanza database Oracle può trovarsi nello stesso VPC della directory o in un VPC diverso. L'istanza database deve utilizzare un gruppo di sicurezza che accetta traffico in uscita all'interno del VPC della directory, in modo che l'istanza database possa comunicare con la directory.

Quando utilizzi la console per creare, modificare o ripristinare un'istanza database, scegli Password e autenticazione Kerberos nella sezione Autenticazione database. Scegli Browse Directory (Sfoglia directory) quindi seleziona la directory oppure scegli Create a new directory (Crea una nuova directory).



Quando utilizzi l'API AWS CLI o RDS, associa un'istanza DB a una directory. Per consentire all'istanza database di utilizzare la directory del dominio che hai creato, sono necessari i seguenti parametri:

- Per il parametro `--domain`, utilizza l'identificatore di dominio (identificatore "d-*") generato durante la creazione della directory.
- Per il parametro `--domain-iam-role-name`, utilizza il ruolo creato che utilizza la policy IAM gestita `AmazonRDSDirectoryServiceAccess`.

Ad esempio, il comando CLI seguente modifica un'istanza database per utilizzare una directory.

Per LinuxmacOS, oUnix:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --domain d-ID \
    --domain-iam-role-name role-name
```

Per Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --domain d-ID ^
    --domain-iam-role-name role-name
```

> **Important**
>
> Se modifichi un'istanza database per abilitare l'autenticazione Kerberos, riavvia l'istanza database dopo aver apportato la modifica.

Fase 5: creazione di login MySQL di autenticazione Kerberos

Usa le credenziali dell'utente master Amazon RDS per eseguire la connessione all'istanza database MySQL come con qualunque altra istanza database. L'istanza DB viene aggiunta al AWS Managed Microsoft AD dominio. Pertanto, puoi effettuare il provisioning di login e utenti MySQL da utenti Active

Directory nel dominio. Le autorizzazioni del database vengono gestite tramite autorizzazioni MySQL standard concesse e revocate da questi accessi.

È possibile consentire a un utente di Active Directory di autenticarsi con MySQL. Per fare ciò, utilizzare innanzitutto le credenziali utente master Amazon RDS per connettersi all'istanza database MySQL come con qualsiasi altra istanza database. Dopo aver effettuato l'accesso, crea un utente autenticato esternamente con PAM (Pluggable Authentication Modules) in MySQL eseguendo il seguente comando. Sostituire *testuser* con il nome utente.

```
CREATE USER 'testuser'@'%' IDENTIFIED WITH 'auth_pam';
```

Gli utenti (persone e applicazioni) del dominio possono ora connettersi all'istanza database da un computer client associato al dominio utilizzando l'autenticazione Kerberos.

> **Important**
>
> È consigliabile che i client utilizzino connessioni SSL/TLS quando si utilizza l'autenticazione PAM. Se non utilizzano connessioni SSL/TLS, in alcuni casi la password potrebbe essere inviata come testo non crittografato. Per richiedere una connessione crittografata SSL/TLS per il tuo utente AD, esegui il comando seguente e sostituiscilo con il nome utente: *testuser*
>
> ```
> ALTER USER 'testuser'@'%' REQUIRE SSL;
> ```
>
> Per ulteriori informazioni, consulta Utilizzo di SSL/TLS con un'istanza database MySQL.

Gestione di un'istanza database in un dominio

Puoi utilizzare la CLI o l'API RDS per gestire l'istanza database e la sua relazione con Active Directory gestita. Ad esempio, è possibile associare un'autenticazione di Active Directory per Kerberos e annullare l'associazione di una Active Directory per disabilitare l'autenticazione Kerberos. Puoi anche spostare un'istanza database affinché venga autenticata esternamente da una Microsoft Active Directory a un'altra.

Ad esempio, puoi utilizzare l'API Amazon RDS per effettuare quanto segue:

- Per tentare di attivare nuovamente l'autenticazione Kerberos per un'appartenenza non riuscita, utilizzare l'operazione API `ModifyDBInstance` e specificare l'ID della directory dell'appartenenza corrente.
- Per aggiornare il nome del ruolo IAM dell'appartenenza, utilizza l'operazione API `ModifyDBInstance` e specifica l'ID della directory dell'appartenenza corrente e il nuovo ruolo IAM.
- Per disabilitare l'autenticazione Kerberos in un'istanza database, utilizzare l'operazione API `ModifyDBInstance` e specificare `none` come parametro di dominio.
- Per spostare un'istanza database da un dominio a un altro, utilizza l'operazione API `ModifyDBInstance` e specifica l'identificatore di dominio del nuovo dominio come parametro del dominio.
- Per elencare l'appartenenza per ogni istanza database, utilizzare l'operazione API `DescribeDBInstances`.

Appartenenza al dominio

Quando l'istanza database viene creata o modificata diventa membro del dominio. È possibile visualizzare lo stato dell'appartenenza al dominio per l'istanza DB eseguendo il comando describe-db-instancesCLI. Lo stato dell'istanza di database può essere uno dei seguenti:

- `kerberos-enabled`: l'autenticazione Kerberos è abilitata nell'istanza database.
- `enabling-kerberos`— AWS sta abilitando l'autenticazione Kerberos su questa istanza DB.
- `pending-enable-kerberos`: l'abilitazione dell'autenticazione Kerberos è in sospeso su questa istanza database.
- `pending-maintenance-enable-kerberos`— AWS tenterà di abilitare l'autenticazione Kerberos sull'istanza DB durante la successiva finestra di manutenzione pianificata.
- `pending-disable-kerberos`: la disabilitazione dell'autenticazione Kerberos è in sospeso su questa istanza database.
- `pending-maintenance-disable-kerberos`— AWS tenterà di disabilitare l'autenticazione Kerberos sull'istanza DB durante la successiva finestra di manutenzione programmata.
- `enable-kerberos-failed` - Un problema di configurazione ha impedito a AWS di abilitare l'autenticazione Kerberos sull'istanza database. Verifica e correggi la configurazione prima di eseguire nuovamente il comando di modifica dell'istanza database.
- `disabling-kerberos`— AWS sta disabilitando l'autenticazione Kerberos su questa istanza DB.

Una richiesta per abilitare l'autenticazione Kerberos potrebbe non andare a buon fine a causa di un problema di connettività di rete o un ruolo IAM non corretto. Ad esempio, si supponga di creare un'istanza database o di modificare un'istanza database esistente e il tentativo di attivare l'autenticazione Kerberos non riesce. In questo caso, eseguire nuovamente il comando di modifica o modificare l'istanza database appena creata per l'aggiunta al dominio.

Connessione a MySQL con l'autenticazione Kerberos

Per connetterti a MySQL con l'autenticazione Kerberos, è necessario accedere utilizzando il tipo di autenticazione Kerberos.

Per creare un utente di database a cui è possibile connettersi utilizzando l'autenticazione Kerberos, utilizzare una clausola `IDENTIFIED WITH` sull'istruzione `CREATE USER`. Per istruzioni, consulta Fase 5: creazione di login MySQL di autenticazione Kerberos.

Per evitare errori, utilizzare il client `mysql` MariaDB. È possibile scaricare il software MariaDB all'indirizzo https://downloads.mariadb.org/.

Al prompt dei comandi, connettersi a uno degli endpoint associati all'istanza database MySQL. Seguire le procedure generali descritte in Connessione a un'istanza database che esegue il motore di database di MySQL. Quando viene richiesta la password, immettere la password Kerberos associata al nome utente.

Ripristino di un'istanza database MySQL e aggiunta a un dominio

È possibile ripristinare uno snapshot DB o completare un point-in-time ripristino per un'istanza DB MySQL e quindi aggiungerla a un dominio. Dopo aver ripristinato l'istanza database, modificarla utilizzando il processo illustrato in Fase 4: creazione o modifica di un'istanza database MySQL per aggiungere l'istanza database a un dominio.

Limitazioni MySQL per l'autenticazione Kerberos

Le seguenti limitazioni si applicano all'autenticazione Kerberos per MySQL:

- È supportato solo un AWS Managed Microsoft AD . Tuttavia, puoi aggiungere le istanze database RDS per MySQL a domini Managed Microsoft AD condivisi di proprietà di account diversi nella stessa Regione AWS.
- È necessario riavviare l'istanza database dopo aver abilitato la caratteristica.
- La lunghezza del nome di dominio non può essere superiore a 61 caratteri.

- Non è possibile abilitare contemporaneamente l'autenticazione Kerberos e l'autenticazione IAM. Scegli un metodo di autenticazione o l'altro per l'istanza database MySQL.
- Non modificare la porta dell'istanza database dopo aver abilitato la caratteristica.
- Non utilizzare l'autenticazione Kerberos con le repliche di lettura.
- Se è attivato l'aggiornamento automatico della versione secondaria per un'istanza database MySQL che utilizza l'autenticazione Kerberos, è necessario disattivare l'autenticazione Kerberos e riattivarla dopo un aggiornamento automatico. Per maggiori informazioni sull'aggiornamento automatico di una versione minore, consulta Aggiornamenti a versioni secondarie automatiche per MySQL.
- Per eliminare un'istanza database con questa caratteristica abilitata, disabilitare prima la caratteristica. A tale scopo, utilizza il comando CLI `modify-db-instance` per l'istanza database e specifica `none` per il parametro `--domain`.

  Se si utilizza l'interfaccia a riga di comando (CLI) o l'API RDS per eliminare un'istanza database con questa caratteristica attivata, prevedere un ritardo.
- Non è possibile configurare una relazione di trust tra foreste tra Microsoft Active Directory on-premise o self-hosted e AWS Managed Microsoft AD.

Prestazioni delle query migliorate per RDS per MySQL con Amazon RDS Optimized Reads

Puoi ottenere un'elaborazione delle query più rapida per RDS per MySQL con Amazon RDS Optimized Reads. Un'istanza database RDS per MySQL o un cluster database multi-AZ che utilizza la funzionalità Letture ottimizzate per Amazon RDS può ottenere un'elaborazione delle query fino a due volte più veloce rispetto a un'istanza o un cluster database che non lo utilizza.

Argomenti

Panoramica di RDS Optimized Reads

Quando si utilizza un'istanza database RDS per MySQL o un cluster database multi-AZ con la funzionalità Letture ottimizzate per Amazon RDS attivata, si otterranno prestazioni di query più rapide tramite l'uso di un archivio dell'istanza. Un archivio istanze fornisce uno storage temporaneo di livello per l'istanza database o il cluster database multi-AZ. L'archiviazione è basata su unità di memoria a stato solido (SSD) NVMe (Non-Volatile Memory Express) fisicamente collegata al server host. Questa archiviazione è ottimizzata per bassa latenza, prestazioni I/O casuali elevate e velocità di trasmissione effettiva di lettura sequenziale elevata.

La funzionalità Letture ottimizzate per Amazon RDS è attivata per impostazione predefinita quando un'istanza database o un cluster database multi-AZ utilizza una classe di istanza database con un archivio dell'istanza, ad esempio db.m5d o db.m6gd. Con RDS Optimized Reads, alcuni oggetti temporanei vengono archiviati nell'archivio dell'istanza. Questi oggetti temporanei includono file temporanei interni, tabelle temporanee interne su disco, file di mappe in memoria e file di cache di log binario (binlog). Per ulteriori informazioni sull'archivio dell'istanza, consulta Instance store Amazon EC2 nella Guida per l'utente di Amazon Elastic Compute Cloud per istanze Linux.

I carichi di lavoro che generano gli oggetti temporanei in MySQL per l'elaborazione delle query possono sfruttare l'archivio dell'istanza per elaborare più rapidamente le query. Questo tipo di

carico di lavoro include query che coinvolgono ordinamenti, aggregazioni di hash, join a carico elevato, espressioni di tabella comuni (CTE) e query su colonne non indicizzate. Questi volumi dell'archivio dell'istanza forniscono operazioni IOPS e prestazioni più elevate, indipendentemente dalle configurazioni utilizzate per l'archivio persistente di Amazon EBS. Poiché RDS Optimized Reads trasferisce le operazioni sugli oggetti temporanei all'archivio dell'istanza, le operazioni di input/output al secondo (IOPS) o la velocità di trasmissione effettiva dell'archivio persistente (Amazon EBS) possono ora essere utilizzate per le operazioni su oggetti persistenti. Queste includono le normali operazioni di lettura e scrittura dei file di dati e le operazioni del motore in background, come lo svuotamento e l'unione di inserimenti di buffer.

> **Note**
>
> Gli snapshot RDS manuali e automatici contengono solo i file del motore per gli oggetti persistenti. Gli oggetti temporanei creati nell'archivio dell'istanza non sono inclusi negli snapshot RDS.

Casi d'uso per RDS Optimized Reads

Se hai carichi di lavoro che si basano pesantemente sugli oggetti temporanei, come tabelle o file interni, per l'esecuzione delle query, puoi trarre vantaggio dall'attivazione di RDS Optimized Reads. I seguenti casi d'uso sono candidati per RDS Optimized Reads:

- Applicazioni che eseguono query analitiche con espressioni di tabella comuni (CTE) complesse, tabelle derivate e operazioni di raggruppamento
- Repliche di lettura che generano un intenso traffico di lettura con query non ottimizzate
- Applicazioni che eseguono query di report on demand o dinamiche che includono operazioni complesse, ad esempio query con le clausole `GROUP BY` e `ORDER BY`
- Carichi di lavoro che utilizzano tabelle temporanee interne per l'elaborazione delle query

  È possibile monitorare la variabile di stato del motore `created_tmp_disk_tables` per determinare il numero di tabelle temporanee basate su disco create nell'istanza database.
- Applicazioni che creano tabelle temporanee di grandi dimensioni, direttamente o tramite procedure, per archiviare risultati intermedi
- Query di database che eseguono il raggruppamento o l'ordinamento di colonne non indicizzate

Best practice per RDS Optimized Reads

Usa le seguenti best practice per RDS Optimized Reads:

- Aggiungi la logica dei tentativi per le query di sola lettura, nel caso in cui non riescano perché l'archivio dell'istanza è completo durante l'esecuzione.
- Monitora lo spazio di archiviazione disponibile nell'archivio dell'istanza con la metrica CloudWatch `FreeLocalStorage`. Se l'archivio dell'istanza sta raggiungendo il limite a causa del carico di lavoro dell'istanza database, modifica l'istanza database in modo da utilizzare una classe di istanza database più grande.
- Se l'istanza database o il cluster database multi-AZ ha memoria sufficiente ma raggiunge comunque il limite di archiviazione dell'archivio dell'istanza, aumentare il valore `binlog_cache_size` per mantenere in memoria le voci binlog specifiche della sessione. Questa configurazione impedisce di scrivere le voci binlog in file di cache binlog temporanei memorizzati su disco.

  Il parametro `binlog_cache_size` è specifico della sessione. É possibile modificare il valore per ogni nuova sessione. L'impostazione di questo parametro può aumentare l'utilizzo della memoria dell'istanza database durante i picchi di carico di lavoro. Pertanto, è consigliabile aumentare il valore del parametro in base al modello di carico di lavoro dell'applicazione e alla memoria disponibile nell'istanza database.
- Usa il valore predefinito `MIXED` per `binlog_format`. A seconda della dimensione delle transazioni, l'impostazione `binlog_format` su `ROW` può comportare la creazione di file di cache binlog di grandi dimensioni nell'archivio dell'istanza.
- Imposta il parametro internal_tmp_mem_storage_engine su `TempTable` e il parametro temptable_max_mmap in modo che corrisponda alla dimensione dello spazio di archiviazione disponibile nell'archivio dell'istanza.
- Evita di apportare modifiche in blocco in una singola transazione. Questi tipi di transazioni possono generare file di cache binlog di grandi dimensioni nell'archivio dell'istanza e possono causare problemi quando l'archivio dell'istanza è pieno. Prendi in considerazione la suddivisione delle scritture in transazioni più piccole per ridurre al minimo l'uso dello spazio di archiviazione per i file di cache binlog.
- Usa il valore predefinito `ABORT_SERVER` per il parametro `binlog_error_action`. In questo modo si evitano problemi con i log binari sulle istanze database con i backup abilitati.

Utilizzo di RDS Optimized Reads

L'istanza database utilizza automaticamente la funzionalità Letture ottimizzate per Amazon RDS quando si effettua il provisioning di un'istanza database RDS per MySQL con una delle seguente classi di istanza database in un'implementazione di istanza database single-AZ o multi-AZ oppure in un'implementazione di cluster database multi-AZ:

Per attivare RDS Optimized Reads, procedi in uno dei seguenti modi:

- Creare un'istanza database o un cluster database multi-AZ RDS per MySQL utilizzando una di queste classi di istanza database. Per ulteriori informazioni, consulta Creazione di un'istanza database Amazon RDS.
- Modificare un'istanza database o un cluster database multi-AZ RDS per MySQL esistente per utilizzare una di queste classi di istanza database. Per ulteriori informazioni, consulta Modifica di un'istanza database Amazon RDS.

La funzionalità Letture ottimizzate per Amazon RDS è disponibile in tutte le Regioni AWS RDS in cui sono supportate una o più delle classi di istanza database con spazio di archiviazione SSD NVMe locale. Per informazioni sulle classi di istanza database, consulta the section called “Classi di istanze database”.

La disponibilità della classe di istanze DB è diversa nelle varie Regioni AWS. Per determinare se una classe di istanza DB è supportata in una Regione AWS specifica, consulta the section called “Determinazione del supporto delle classi di istanze DB in Regioni AWS”.

Se non si desidera utilizzare la funzionalità Letture ottimizzate per Amazon RDS, modificare l'istanza database o il cluster database multi-AZ in modo che non utilizzi una classe di istanza database che supporti la funzionalità.

Monitoraggio delle istanze database che utilizzano RDS Optimized Reads

Puoi monitorare le istanze database che utilizzano RDS Optimized Reads con le seguenti metriche di CloudWatch:

- `FreeLocalStorage`
- `ReadIOPSLocalStorage`
- `ReadLatencyLocalStorage`
- `ReadThroughputLocalStorage`

- `WriteIOPSLocalStorage`
- `WriteLatencyLocalStorage`
- `WriteThroughputLocalStorage`

Queste metriche forniscono dati sullo spazio di archiviazione dell'archivio dell'istanza, sulle operazioni IOPS e sulla velocità di trasmissione effettiva disponibili. Per ulteriori informazioni su questi parametri, consulta Parametri a CloudWatch livello di istanza Amazon per Amazon RDS.

Limitazioni per RDS Optimized Reads

Le seguenti limitazioni si applicano a RDS Optimized Reads:

- RDS Optimized Reads è supportato per RDS per MySQL versione 8.0.28 e successive. Per ulteriori informazioni sulle versioni di RDS per MySQL, consulta Versioni di MySQL in Amazon RDS.
- Non è possibile modificare la posizione degli oggetti temporanei nell'archivio persistente (Amazon EBS) nelle classi di istanza database che supportano RDS Optimized Reads.
- Quando i log binari sono abilitati su un'istanza database, la dimensione massima della transazione è limitata alla dimensione dell'archivio dell'istanza. In MySQL, qualsiasi sessione che richiede più spazio di archiviazione rispetto al valore `binlog_cache_size` scrive le modifiche della transazione nei file di cache binlog temporanei, che vengono creati nell'archivio dell'istanza.
- Le transazioni possono non riuscire quando l'archivio dell'istanza è pieno.

Prestazioni di scrittura migliorate con Scritture ottimizzate per Amazon RDS per MySQL

Puoi migliorare le prestazioni delle transazioni di scrittura con Scritture ottimizzate per RDS per MySQL. Quando il database RDS per MySQL utilizza RDS Optimized Writes, può raggiungere una velocità di trasmissione effettiva delle transazioni di scrittura fino a due volte superiore.

Argomenti

Panoramica di RDS Optimized Writes

Quando attivi Scritture ottimizzate per RDS, i database RDS per MySQL scrivono solo una volta, quando trasferiscono i dati nell'archiviazione durevole senza la necessità del buffer di doppia scrittura. I database continuano a fornire le protezioni delle proprietà ACID per le transazioni di database affidabili, insieme alle prestazioni migliorate.

I database relazionali, come MySQL, forniscono le proprietà ACID di atomicità, consistenza, isolamento e durabilità per le transazioni di database affidabili. Per fornire queste proprietà, MySQL utilizza un'area di archiviazione di dati chiamata buffer di doppia scrittura che impedisce gli errori di scrittura parziale della pagina. Questi errori si verificano nel caso di un guasto hardware mentre il database sta aggiornando una pagina, ad esempio in caso di interruzione dell'alimentazione. Un database MySQL può rilevare le scritture parziali della pagina e recuperarle con una copia della pagina nel buffer di doppia scrittura. Sebbene questa tecnica fornisca protezione, comporta anche operazioni di scrittura aggiuntive. Per ulteriori informazioni sul buffer di doppia scrittura MySQL, consulta Doublewrite Buffer (Buffer di doppia scrittura) nella documentazione di MySQL.

Quando attivi RDS Optimized Writes, i database RDS per MySQL scrivono una sola volta, quando trasferiscono i dati nell'archiviazione durevole senza usare il buffer di doppia scrittura. RDS Optimized Writes è utile se esegui carichi di lavoro intensivi in scrittura sui database RDS per MySQL. Esempi di database con carichi di lavoro intensivi in scrittura includono quelli che supportano pagamenti digitali, trading finanziario e applicazioni di gioco.

Questi database vengono eseguiti in classi di istanza database che utilizzano AWS Nitro System. Grazie alla configurazione hardware di questi sistemi, il database può scrivere pagine da 16 KiB direttamente su file di dati in modo affidabile e durevole in un solo passaggio. AWS Nitro System permette di usare RDS Optimized Writes.

Puoi impostare il nuovo parametro di database `rds.optimized_writes` per controllare la funzionalità RDS Optimized Writes per i database RDS per MySQL. Accedi a questo parametro nei gruppi di parametri database RDS per MySQL versione 8.0. Imposta il parametro su uno dei seguenti valori:

- `AUTO` - Attiva RDS Optimized Writes se la funzionalità è supportata dal database. In caso contrario, disattiva RDS Optimized Writes. Questa è l'impostazione di default.
- `OFF` - Disattiva RDS Optimized Writes anche la funzionalità è supportata dal database.

Se disponi di un database esistente con una versione del motore, una classe di istanza database e/o un formato del file system che non supporta Scritture ottimizzate per RDS, puoi abilitare la funzionalità creando un'implementazione blu/verde. Per ulteriori informazioni, consulta the section called “Abilitazione in un database esistente”.

Se si esegue la migrazione di un database RDS per MySQL configurato per utilizzare RDS Optimized Writes in una classe di istanza database che non supporta la funzionalità, RDS disattiva automaticamente RDS Optimized Writes per il database.

Quando la funzionalità RDS Optimized Writes è disattivata, il database utilizza il buffer di doppia scrittura MySQL.

Per determinare se un database RDS per MySQL utilizza RDS Optimized Writes, osserva il valore corrente del parametro `innodb_doublewrite` per il database. Se il database utilizza RDS Optimized Writes, questo parametro è impostato su `FALSE` (`0`).

Utilizzo di RDS Optimized Writes

È possibile attivare RDS Optimized Writes quando si crea un database RDS per MySQL con la console RDS, AWS CLI o l'API RDS. La funzionalità RDS Optimized Writes viene attivata automaticamente quando si verificano entrambe le seguenti condizioni durante la creazione del database:

- Si specificano una versione del motore di database e una classe di istanza database che supportano RDS Optimized Writes.

- RDS Optimized Writes è supportato per RDS per MySQL versione 8.0.30 e successive. Per ulteriori informazioni sulle versioni di RDS per MySQL, consulta Versioni di MySQL in Amazon RDS.
- La funzionalità RDS Optimized Writes è supportata per i database RDS per MySQL che utilizzano le seguenti classi di istanza database:
  - db.m7g
  - db.m6g
  - db.m6gd
  - db.m6i
  - db.m5
  - db.m5d
  - db.r7g
  - db.r6g
  - db.r6gd
  - db.r6i
  - db.r5
  - db.r5b
  - db.r5d
  - db.x2idn
  - db.x2iedn

  Per informazioni sulle classi di istanza database, consulta the section called “Classi di istanze database”.

  La disponibilità della classe di istanze DB è diversa nelle varie Regioni AWS. Per determinare se una classe di istanza DB è supportata in una Regione AWS specifica, consulta the section called “Determinazione del supporto delle classi di istanze DB in Regioni AWS”.

  Per aggiornare il database a una classe di istanza database che supporti Scritture ottimizzate per RDS, puoi creare un'implementazione blu/verde. Per ulteriori informazioni, consulta the section called “Abilitazione in un database esistente”.
- Nel gruppo di parametri associato al database, il parametro `rds.optimized_writes` è impostato su `AUTO`. Nei gruppi di parametri predefiniti, questo parametro è sempre impostato su `AUTO`.

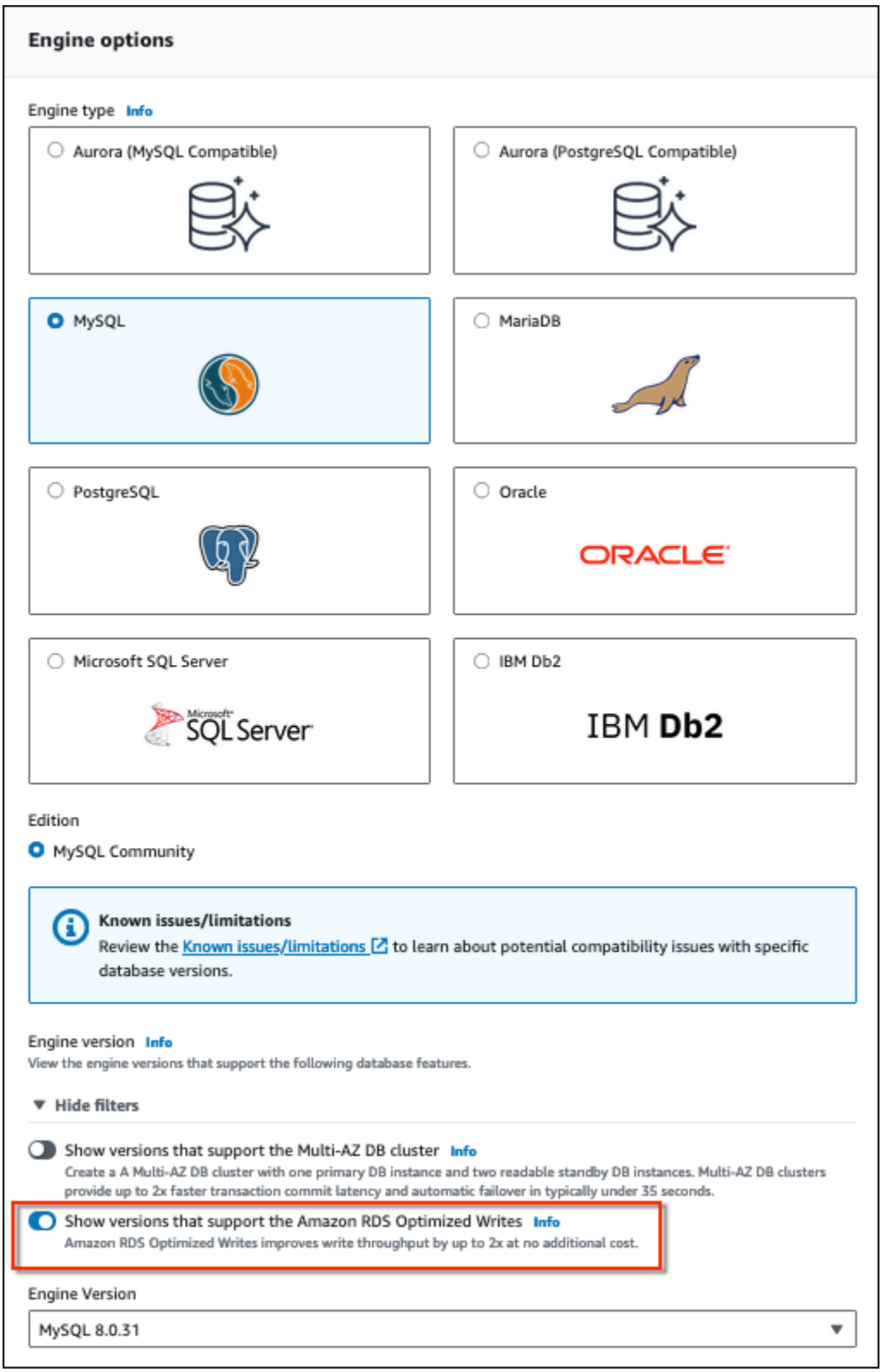
Se vuoi utilizzare una versione del motore di database e una classe di istanza database che supportino Scritture ottimizzate per Amazon RDS, senza usare questa funzionalità, specifica un gruppo di parametri personalizzato durante la creazione del database. In questo gruppo di parametri, imposta il parametro `rds.optimized_writes` su `OFF`. Se si desidera che il database utilizzi RDS Optimized Writes in un secondo momento, è possibile impostare il parametro su `AUTO` per attivarlo. Per informazioni sull'utilizzo dei gruppi di parametri personalizzati e sull'impostazione dei parametri, consulta Utilizzo di gruppi di parametri.

Per informazioni sulla creazione di un'istanza database, consulta Creazione di un'istanza database Amazon RDS.

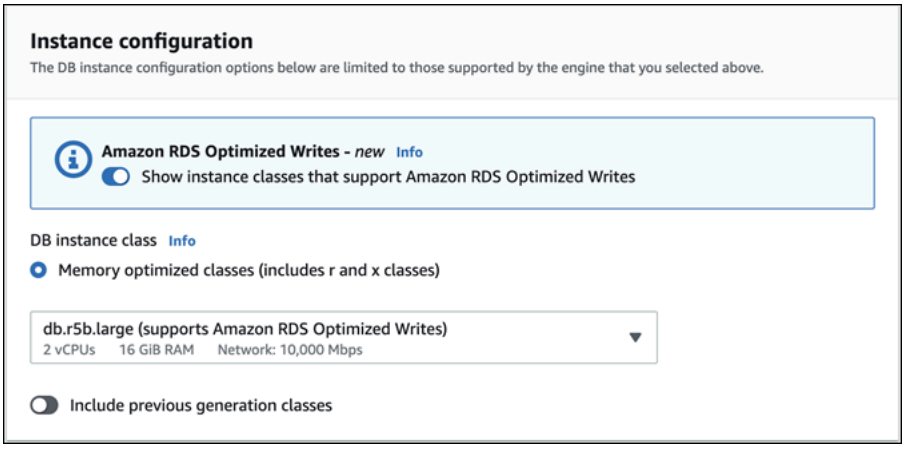
Console

Quando usi la console RDS per creare un database RDS per MySQL, puoi filtrare le versioni del motore di database e le classi di istanza database che supportano RDS Optimized Writes. Dopo aver attivato i filtri, puoi scegliere tra le versioni del motore di database e le classi di istanza database disponibili.

Per scegliere una versione del motore di database che supporti RDS Optimized Writes, filtra le versioni del motore di database RDS per MySQL che supportano la funzionalità in Engine version (Versione del motore), quindi scegli una versione.



Nella sezione Instance configuration (Configurazione dell'istanza), filtra le classi di istanza database che supportano RDS Optimized Writes, quindi scegli una classe di istanza database.



Dopo aver effettuato queste selezioni, puoi scegliere altre impostazioni che soddisfano i tuoi requisiti e completare la creazione del database RDS per MySQL con la console.

AWS CLI

Per creare un'istanza DB utilizzando ilAWS CLI, utilizzare il create-db-instancecomando. Assicurati che i valori `--engine-version` e `--db-instance-class` supportino RDS Optimized Writes. Inoltre, assicurati che il gruppo di parametri associato all'istanza database abbia il parametro `rds.optimized_writes` impostato su `AUTO`. Questo esempio associa il gruppo di parametri predefinito all'istanza database.

Example Creazione di un'istanza database che utilizza RDS Optimized Writes

Per LinuxmacOS, oUnix:

```
aws rds create-db-instance \
    --db-instance-identifier mydbinstance \
    --engine mysql \
    --engine-version 8.0.30 \
    --db-instance-class db.r5b.large \
    --manage-master-user-password \
    --master-username admin \
    --allocated-storage 200
```

Per Windows:

```
aws rds create-db-instance ^
    --db-instance-identifier mydbinstance ^
    --engine mysql ^
    --engine-version 8.0.30 ^
```

```
    --db-instance-class db.r5b.large ^
    --manage-master-user-password ^
    --master-username admin ^
    --allocated-storage 200
```

API RDS

È possibile creare un'istanza database utilizzando l'operazione CreateDBInstance. Quando utilizzi questa operazione, assicurati che i valori `EngineVersion` e `DBInstanceClass` supportino RDS Optimized Writes. Inoltre, assicurati che il gruppo di parametri associato all'istanza database abbia il parametro `rds.optimized_writes` impostato su `AUTO`.

Abilitazione delle scritture ottimizzate per RDS in un database esistente

Per modificare un database RDS per MySQL esistente per attivare Scritture ottimizzate per RDS, il database deve essere stato creato con una versione del motore di database e una classe di istanza database supportate. Inoltre, il database deve essere stato creato dopo il rilascio di Scritture ottimizzate per RDS del 27 novembre 2022, poiché la configurazione del file system sottostante richiesta è incompatibile con quella dei database creati prima del rilascio. Se queste condizioni sono soddisfatte, è possibile attivare Scritture ottimizzate per RDS impostando il parametro `rds.optimized_writes` su `AUTO`.

Se il database non è stato creato con una versione del motore, una classe di istanza o una configurazione del file system supportate, puoi utilizzare le implementazioni blu/verde di RDS per migrare a una configurazione supportata. Durante la creazione dell'implementazione blu/verde, esegui le seguenti operazioni:

- Seleziona Abilita le scritture ottimizzate sul database verde, quindi specifica una versione del motore e una classe di istanza database che supportano e scritture ottimizzate RDS. Per l'elenco delle versioni di motore e delle classi di istanza supportate, consulta Utilizzo di RDS Optimized Writes.
- In Archiviazione scegli Aggiorna la configurazione del file system di archiviazione. Questa opzione aggiorna il database a una configurazione del file system sottostante compatibile.

Se quando crei l'implementazione blu/verde il parametro `rds.optimized_writes` è impostato su `AUTO`, Scritture ottimizzate per RDS viene abilitato automaticamente nell'ambiente verde. Quindi puoi eseguire lo switchover all'implementazione blu/verde, che rende l'ambiente verde il nuovo ambiente di produzione.

Per ulteriori informazioni, consulta the section called “Creazione di un'implementazione blu/verde”.

Limitazioni per RDS Optimized Writes

Quando si ripristina un database RDS per MySQL da uno snapshot, è possibile attivare Scritture ottimizzate per RDS per il database solo se si verificano tutte le seguenti condizioni:

- Lo snapshot è stato creato da un database che supporta RDS Optimized Writes.
- Lo snapshot è stato creato da un database creato dopo il rilascio della funzionalità Scritture ottimizzate per Amazon RDS.
- Lo snapshot è stato ripristinato in un database che supporta RDS Optimized Writes.
- Il database ripristinato è associato a un gruppo di parametri con il parametro `rds.optimized_writes` impostato su `AUTO`.

Aggiornamento del motore di database MySQL

Quando Amazon RDS supporta una nuova versione di un motore del database, puoi effettuare l'aggiornamento delle istanze database alla nuova versione. Esistono due tipi di aggiornamenti per i database MySQL: aggiornamenti delle versioni principali e aggiornamenti delle versioni secondarie.

Aggiornamenti di una versione principale

Gli aggiornamenti di versione principali possono contenere modifiche al database non compatibili con le versioni precedenti delle applicazioni esistenti. Ne risulta che è necessario eseguire manualmente gli aggiornamenti della versione principale per le proprie istanze database. Puoi avviare manualmente un aggiornamento principale a una versione modificando l'istanza. Prima di eseguire un aggiornamento della versione principale, si consiglia di seguire le istruzioni riportate in. Aggiornamenti di versione principale per MySQL

Per gli upgrade delle versioni principali delle implementazioni di istanze DB Multi-AZ, Amazon RDS aggiorna contemporaneamente sia le repliche primarie che quelle in standby. L'istanza DB non sarà disponibile fino al completamento dell'aggiornamento. Attualmente, Amazon RDS non supporta gli aggiornamenti delle versioni principali per le implementazioni di cluster DB Multi-AZ.

> **Tip**
>
> Puoi ridurre al minimo i tempi di inattività necessari per un aggiornamento di una versione principale utilizzando una distribuzione blu/verde. Per ulteriori informazioni, consulta Utilizzo delle implementazioni blu/verde Amazon RDS per gli aggiornamenti del database.

Aggiornamenti della versione secondaria

Gli aggiornamenti di versione minori includono solo le modifiche retrocompatibili con le applicazioni esistenti. Puoi avviare un aggiornamento a una versione secondaria manualmente modificando la tua istanza database. In alternativa, è possibile abilitare l'opzione di aggiornamento automatico della versione secondaria durante la creazione o la modifica di un'istanza DB. Ciò significa che Amazon RDS aggiorna automaticamente l'istanza DB dopo aver testato e approvato la nuova versione. Per informazioni sull'esecuzione di un aggiornamento, consulta Aggiornamento della versione del motore di un'istanza database.

Quando esegui un aggiornamento di versione minore di un cluster DB Multi-AZ, Amazon RDS aggiorna le istanze DB del lettore una alla volta. Quindi, una delle istanze Reader DB diventa

la nuova istanza DB Writer. Amazon RDS aggiorna quindi la vecchia istanza writer (che ora è un'istanza reader).

> **Note**
>
> Il tempo di inattività per un aggiornamento di versione minore di un'implementazione di un'istanza DB Multi-AZ può durare diversi minuti. I cluster DB Multi-AZ in genere riducono i tempi di inattività degli aggiornamenti di versioni minori a circa 35 secondi. Se utilizzati con RDS Proxy, è possibile ridurre ulteriormente i tempi di inattività a un secondo o meno. Per ulteriori informazioni, consulta Utilizzo del Proxy RDS. In alternativa, è possibile utilizzare un proxy di database open source come ProxySQL o il driver PgBouncerAWSJDBC per MySQL.

Se l'istanza DB MySQL utilizza repliche di lettura, è necessario aggiornare tutte le repliche di lettura prima di aggiornare l'istanza di origine.

Argomenti

- Panoramica dell'aggiornamento
- Numeri di versione di MySQL
- Numero di versione RDS
- Aggiornamenti di versione principale per MySQL
- Verifica di un aggiornamento
- Aggiornamento di un'istanza database MySQL
- Aggiornamenti a versioni secondarie automatiche per MySQL
- Utilizzo di una replica di lettura per ridurre i tempi di inattività durante l'aggiornamento di un database MySQL

Panoramica dell'aggiornamento

Quando si utilizza AWS Management Console per aggiornare un'istanza DB, vengono visualizzati gli obiettivi di aggiornamento validi per l'istanza DB. È inoltre possibile utilizzare il AWS CLI comando seguente per identificare gli obiettivi di aggiornamento validi per un'istanza DB:

Per LinuxmacOS, oUnix:

```
aws rds describe-db-engine-versions \
  --engine mysql \
  --engine-version version-number \
  --query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" --
output text
```

Per Windows:

```
aws rds describe-db-engine-versions ^
  --engine mysql ^
  --engine-version version-number ^
  --query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" --
output text
```

Ad esempio, per identificare gli obiettivi di aggiornamento validi per un'istanza DB MySQL versione 8.0.28, esegui il comando seguente: AWS CLI

PerLinux, o: macOS Unix

```
aws rds describe-db-engine-versions \
  --engine mysql \
  --engine-version 8.0.28 \
  --query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" --
output text
```

Per Windows:

```
aws rds describe-db-engine-versions ^
  --engine mysql ^
  --engine-version 8.0.28 ^
  --query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" --
output text
```

Durante il processo di aggiornamento, Amazon RDS acquisisce due o più istantanee DB. Amazon RDS richiede fino a due istantanee dell'istanza database prima di apportare modifiche all'aggiornamento. Se l'aggiornamento non funziona per i database, puoi ripristinare una di queste istantanee per creare un'istanza database che esegue la versione precedente. Amazon RDS acquisisce un'altra istantanea dell'istanza database al termine dell'aggiornamento. Amazon RDS acquisisce queste istantanee indipendentemente dal fatto che AWS Backup gestisca o meno i backup per l'istanza DB.

> **Note**
>
> Amazon RDS acquisisce gli snapshot DB solo se hai impostato il periodo di retention dei backup per l'istanza database su un valore maggiore di 0. Per cambiare il periodo di retention dei backup, consulta Modifica di un'istanza database Amazon RDS.

Al termine dell'aggiornamento, non puoi ripristinare la versione precedente del motore di database. Se desideri tornare alla versione precedente, ripristina il primo snapshot DB acquisito per creare una nuova istanza database.

Puoi controllare quando eseguire l'aggiornamento dell'istanza database a una nuova versione supportata da Amazon RDS. Questo livello di controllo ti consente di mantenere la compatibilità con versioni di database specifiche e testare le nuove versioni con l'applicazione prima di distribuirle in produzione. Puoi aggiornare le versioni quando più appropriato in base alla tua pianificazione.

Se la tua istanza DB utilizza la replica di lettura, devi aggiornare tutte le repliche di lettura prima di aggiornare l'istanza di origine.

Numeri di versione di MySQL

La sequenza di numerazione delle versioni per il motore di database RDS for MySQL è nella forma di major.minor.patch.yyyymmdd o major.minor.patch, ad esempio 8.0.33.R2.20231201 o 5.7.44. Il formato utilizzato dipende dalla versione del motore MySQL. Per informazioni sulla numerazione delle versioni di RDS Extended Support, vedere. Denominazione delle versioni di Amazon RDS Extended Support

principale

Il numero di versione principale è sia il numero intero che la prima parte frazionaria del numero di versione, ad esempio 8.0. Un aggiornamento della versione principale incrementa la parte principale del numero di versione. Ad esempio, un aggiornamento da 5.7.4.4 a 8.0.33 è un aggiornamento della versione principale, dove 5.7 e 8.0 sono i numeri di versione principali.

minore

Il numero di versione secondario è la terza parte del numero di versione, ad esempio la 33 in 8.0.33.

patch

La patch è la quarta parte del numero di versione, ad esempio la R2 in 8.0.33.R2. Una versione della patch di RDS include importanti correzioni di bug aggiunte a una versione secondaria dopo il rilascio.

YYYYMMGD

La data è la quinta parte del numero di versione, ad esempio 20231201 in 8.0.33.R2.20231201. Una versione con data RDS è una patch di sicurezza che include importanti correzioni di sicurezza aggiunte a una versione secondaria dopo il suo rilascio. Non include correzioni che potrebbero modificare il comportamento di un motore.

Versione principale	Versione secondaria	Schema di denominazione
8.0	≥ 33	Le nuove istanze DB utilizzano Major.min or.patch.YYMMDD, ad esempio 8.0.33.R2 .20231201. Le istanze DB esistenti potrebbero utilizzar e major.minor.patch, ad esempio 8.0.33.R2, fino al prossimo aggiornamento della versione principale o secondaria.
	< 33	Le istanze DB esistenti utilizzano major.min or.patch, ad esempio 8.0.32.R2.
5.7	≥ 42	Le nuove istanze DB utilizzano Major.min or.patch.YYMMDD, ad esempio 5.7.42.R2 .20231201. Le istanze DB esistenti potrebbero utilizzar e major.minor.patch, ad esempio 5.7.42.R2, fino al prossimo aggiornamento della versione principale o secondaria.

Numero di versione RDS

I numeri di versione RDS utilizzano lo schema di denominazione o lo schema di denominazione. *major*.*minor*.*patch* *major*.*minor*.*patch*.*YYYYMMDD* Una versione della patch di RDS include importanti correzioni di bug aggiunte a una versione secondaria dopo il rilascio. Una versione con data RDS (*YYMMDD*) è una patch di sicurezza. Una patch di sicurezza non include correzioni che potrebbero modificare il comportamento del motore. Per informazioni sulla numerazione delle versioni di RDS Extended Support, vedere. Denominazione delle versioni di Amazon RDS Extended Support

Per identificare il numero di versione Amazon RDS del tuo database, è prima necessario creare l'estensione `rds_tools` utilizzando il seguente comando:

```
CREATE EXTENSION rds_tools;
```

Puoi scoprire il numero di versione RDS del tuo database RDS for MySQL con la seguente query SQL:

```
mysql> select mysql.rds_version();
```

Ad esempio, l'interrogazione di un database RDS per MySQL 8.0.34 restituisce il seguente output:

```
+---------------------+
| mysql.rds_version() |
+---------------------+
| 8.0.34.R2.20231201  |
+---------------------+
1 row in set (0.01 sec)
```

Aggiornamenti di versione principale per MySQL

Amazon RDS supporta i seguenti aggiornamenti in loco per le versioni principali del motore di database MySQL:

- Da MySQL 5.6 a MySQL 5.7
- Da MySQL 5.7 a MySQL 8.0

> **Note**
>
> Puoi creare istanze database di MySQL versione 5.7 e 8.0 con classi di istanze database della generazione corrente e dell'ultima generazione. oltre alla classe di istanze database della generazione precedente db.m3.
> In alcuni casi, potresti voler aggiornare un'istanza database di MySQL versione 5.6 in esecuzione su una classe di istanza database di generazione precedente (diversa da db.m3) in un'istanza database di MySQL versione 5.7. In questi casi, modifica innanzitutto l'istanza database per utilizzare una classe di istanza database di ultima generazione o generazione corrente. Dopo aver effettuato questa operazione, puoi modificare l'istanza database affinché utilizzi il motore di database di MySQL versione 5.7. Per informazioni sulle classi di istanza database Amazon RDS, consulta Classi di istanze database.

Argomenti

Panoramica degli aggiornamenti di una versione principale di MySQL

Gli aggiornamenti di versione principali possono contenere modifiche al database non compatibili con le versioni precedenti delle applicazioni esistenti. Di conseguenza, Amazon RDS non applica automaticamente gli aggiornamenti di versioni principale; devi modificare l'istanza database manualmente. Ti raccomandiamo di eseguire un test approfondito di qualsiasi aggiornamento prima di applicarlo alle istanze di produzione.

Per eseguire un aggiornamento di versione principale per un'istanza database di MySQL 5.6 su Amazon RDS a MySQL 5.7 o versione successiva, esegui prima tutti gli aggiornamenti del sistema operativo disponibili. Al termine degli aggiornamenti del sistema operativo, esegui l'aggiornamento a ogni versione principale: da 5.6 a 5.7 e quindi da 5.7 a 8.0. Le istanze database MySQL create prima del 24 aprile 2014 indicano la disponibilità di un aggiornamento del sistema operativo fino a quando l'aggiornamento non viene applicato. Per ulteriori informazioni sugli aggiornamenti del sistema operativo, consulta Applicazione di aggiornamenti a un'istanza database.

Durante un aggiornamento della versione principale di MySQL, Amazon RDS esegue il file binario `mysql_upgrade` di MySQL per aggiornare le tabelle, se necessario. Amazon RDS svuota inoltre le tabelle `slow_log` e `general_log` durante un aggiornamento della versione principale. Per conservare le informazioni di log, salva il contenuto dei log prima dell'aggiornamento di versione principale.

Gli aggiornamenti di versione principale di MySQL sono in genere completati nel giro di 10 minuti. Alcuni aggiornamenti possono richiedere più tempo a causa della dimensione della classe di istanza database o perché l'istanza non segue determinate linee guida operative descritte in Best practice per Amazon RDS. Se aggiorni un'istanza database dalla console di Amazon RDS, lo stato dell'istanza database indica quando l'aggiornamento è terminato. Se esegui l'aggiornamento utilizzando AWS Command Line Interface (AWS CLI), usa il comando e controlla il describe-db-instancesvalore. `Status`

Gli aggiornamenti a MySQL versione 5.7 potrebbero risultare lenti

MySQL versione 5.6.4 ha introdotto un nuovo formato di data e ora per le colonne `datetime`, `time` e `timestamp` che autorizza componenti frazionari nei valori di data e ora. Quando esegui l'aggiornamento di un'istanza database a MySQL versione 5.7, MySQL forza la conversione di tutti i tipi di colonne di data e ora al nuovo formato.

Poiché questa conversione ricrea le tabelle, il completamento dell'aggiornamento dell'istanza database può richiedere parecchio tempo. La conversione forzata si verifica per tutte le istanze database che eseguono una versione precedente alla versione 5.6.4 di MySQL. Si verifica anche per tutte le istanze database che sono state aggiornate da una versione precedente alla versione 5.6.4 di MySQL a una versione diversa da 5.7.

Se l'istanza database esegue una versione precedente alla versione 5.6.4 di MySQL o è stata aggiornata da una versione precedente alla 5.6.4, ti consigliamo un passaggio aggiuntivo. In questi casi, ti consigliamo di convertire le colonne `datetime`, `time` e `timestamp` nel tuo database prima di aggiornare l'istanza database a MySQL versione 5.7. Questa conversione può ridurre in modo significativo il tempo necessario per aggiornare l'istanza database a MySQL versione 5.7. Per aggiornare le colonne di data e ora al nuovo formato, esegui il comando `ALTER TABLE <table_name> FORCE;` per ogni tabella contenente tali colonne. Poiché la modifica di una tabella la rende di sola lettura, ti consigliamo di eseguire questo aggiornamento durante una finestra di manutenzione.

Puoi utilizzare la seguente query per trovare tutte le tabelle del database che hanno colonne di tipo `datetime`, `time` o `timestamp` e creare un comando `ALTER TABLE <table_name> FORCE;` per ogni tabella.

```
SET show_old_temporals = ON;
   SELECT table_schema, table_name,column_name, column_type
   FROM information_schema.columns
   WHERE column_type LIKE '%/* 5.5 binary format */';
   SET show_old_temporals = OFF;
```

Controlli preliminari per aggiornamenti da MySQL 5.7 a 8.0

MySQL 8.0 include un certo numero di incompatibilità con MySQL 5.7. Queste incompatibilità possono causare problemi durante l'aggiornamento da MySQL 5.7 a MySQL 8.0. Pertanto, potrebbe essere necessaria una specifica preparazione del database affinché l'aggiornamento possa concludersi correttamente. Di seguito è riportato un elenco generale di queste incompatibilità:

- Non devono essere presenti tabelle che utilizzano tipi di dati o funzioni obsolete.
- Non devono esistere file *.frm orfani.
- I trigger non devono avere un definer mancante o vuoto oppure un contesto di creazione non valido.
- Non devono essere presenti tabelle partizionate che utilizzano un motore di storage che non dispone di supporto di partizionamento nativo.
- Non devono essere presenti violazioni di parole chiave o parole riservate. Alcune parole chiavi, che non erano riservate in precedenza, possono essere riservate in MySQL 8.0.

  Per ulteriori informazioni, consulta Keywords and Reserved Words nella documentazione MySQL.
- Non devono essere presenti tabelle nel database di sistema MySQL 5.7 `mysql` che hanno lo stesso nome di una tabella utilizzata dal dizionario dati MySQL 8.0.
- Non devono esistere modalità SQL obsolete definite nell'impostazione della variabile di sistema `sql_mode`.
- Non devono essere presenti tabelle o stored procedure con singoli elementi di colonna `ENUM` o `SET` la cui lunghezza è superiore a 255 caratteri o 1020 byte.
- Prima dell'aggiornamento a MySQL 8.0.13 o versioni successive, non devono esistere partizioni di tabella che risiedono in spazi tabelle InnoDB condivisi.
- Non devono essere presenti definizioni di query e di programmi archiviati da MySQL 8.0.12 o versione inferiore che utilizzano qualificatori `ASC` o `DESC` per clausole `GROUP BY`.

- L'installazione MySQL 5.7 non deve utilizzare caratteristiche che non sono supportate in MySQL 8.0.

  Per ulteriori informazioni, consulta Features Removed in MySQL 8.0 nella documentazione MySQL.
- Non devono essere presenti nomi di vincoli della chiave più lunghi di 64 caratteri.
- Per supporto Unicode migliorato, valuta la conversione di oggetti che utilizzano il charset `utf8mb3` per utilizzare il charset `utf8mb4`. Il set di caratteri `utf8mb3` è obsoleto. Inoltre, valuta l'utilizzo di `utf8mb4` per i riferimenti al set di caratteri anziché `utf8`, perché attualmente `utf8` è un'alias per il charset `utf8mb3`.

  Per ulteriori informazioni, consulta The utf8mb3 Character Set (3-Byte UTF-8 Unicode Encoding) nella documentazione MySQL.

Quando avvii un aggiornamento da MySQL 5.7 a 8.0, Amazon RDS esegue automaticamente dei controlli preliminari per rilevare queste incompatibilità. Per informazioni sull'aggiornamento a MySQL 8.0, consulta Upgrading MySQL nella documentazione MySQL.

Questi controlli preliminari sono obbligatori. Non puoi scegliere di saltarli. I controlli preliminari offrono i seguenti vantaggi:

- Ti consentono di evitare tempi di inattività non pianificati durante l'aggiornamento.
- Se sono presenti incompatibilità, Amazon RDS impedisce l'aggiornamento e fornisce un log per ottenere informazioni sulle stesse. Puoi quindi utilizzare il log per preparare il database per l'aggiornamento a MySQL 8.0 riducendo le incompatibilità. Per informazioni dettagliate sulla rimozione di incompatibilità, consulta l'argomento relativo alla preparazione dell'installazione per l'aggiornamento nella documentazione di MySQL e il post relativo alle informazioni sull'aggiornamento di MySQL 8.0 nel blog di MySQL Server.

I controlli preliminari comprendono alcuni controlli inclusi in MySQL e alcuni che sono stati creati specificamente dal team Amazon RDS. Per informazioni sui controlli preliminari forniti da MySQL, consultare Utility di controllo aggiornamenti.

I controlli preliminari vengono eseguiti prima dell'arresto dell'istanza database per l'aggiornamento, il che significa che non generano alcun tempo di inattività durante l'esecuzione. Se i controlli preliminari rilevano un'incompatibilità, Amazon RDS annulla automaticamente l'aggiornamento prima che l'istanza database venga arrestata. Amazon RDS genera anche un evento per l'incompatibilità.

Per ulteriori informazioni sugli eventi di Amazon RDS, consulta Utilizzo della notifica degli eventi di Amazon RDS.

Amazon RDS memorizza le informazioni dettagliate su ciascuna incompatibilità nel file di log `PrePatchCompatibility.log`. Nella maggior parte dei casi, la voce di log include un collegamento alla documentazione MySQL utile per correggere l'incompatibilità. Per ulteriori informazioni sulla visualizzazione dei file di log, consultare Visualizzazione ed elenco dei file di log del database.

A causa della natura dei controlli preliminari, questi analizzano gli oggetti nel database. Questa analisi comporta il consumo di risorse e incrementa il tempo di completamento dell'aggiornamento.

> **Note**
>
> Amazon RDS esegue tutti questi controlli preliminari solo in caso di aggiornamento da MySQL 5.7 a MySQL 8.0. Per un aggiornamento da MySQL 5.6 a MySQL 5.7, i precontrolli sono limitati a confermare che non ci sono tabelle orfane e che c'è abbastanza spazio di archiviazione per ricostruire le tabelle. I precontrolli on vengono eseguiti per aggiornamenti a versioni precedenti a MySQL 5.7.

Rollback dopo l'errore di aggiornamento da MySQL 5.7 a 8.0

Quando si aggiorna un'istanza database da MySQL versione 5.7 a MySQL versione 8.0, l'aggiornamento può non riuscire. In particolare, può fallire se il dizionario dati contiene incompatibilità che non sono state acquisite dai precontrolli. In questo caso, il database non viene avviato correttamente nella nuova versione di MySQL 8.0. A questo punto, Amazon RDS esegue il rollback delle modifiche eseguite per l'aggiornamento. Dopo il rollback, l'istanza database MySQL esegue MySQL versione 5.7. Quando un aggiornamento non riesce e viene eseguito il rollback, Amazon RDS genera un evento con l'ID evento RDS-EVENT-0188.

In genere, un aggiornamento non riesce perché ci sono incompatibilità nei metadati tra i database nell'istanza database e la versione di MySQL di destinazione. Quando un aggiornamento non riesce, è possibile visualizzare i dettagli su queste incompatibilità nel file `upgradeFailure.log`. Risolvere le incompatibilità prima di provare a eseguire nuovamente l'aggiornamento.

Durante un tentativo di aggiornamento e rollback non riusciti, l'istanza database viene riavviata. Eventuali modifiche dei parametri in sospeso vengono applicate durante il riavvio e persistono dopo il rollback.

Per ulteriori informazioni sull'aggiornamento a MySQL 8.0, consulta i seguenti argomenti nella documentazione di MySQL:

- Preparazione dell'installazione per l'aggiornamento
- Aggiornamento a MySQL 8.0? Ecco cosa devi sapere…

> **Note**
>
> Al momento, il rollback automatico dopo l'errore di aggiornamento è supportato solo per gli aggiornamenti delle versioni principali di MySQL 5.7 a 8.0.

Verifica di un aggiornamento

Prima di eseguire l'aggiornamento di una versione principale nell'istanza database, testa a fondo il database per verificarne la compatibilità con la nuova versione. Testa inoltre tutte le applicazioni che accedono al database per verificarne la compatibilità con la nuova versione. È consigliabile utilizzare la procedura seguente.

Per testare un aggiornamento di una versione principale

1. Esaminare la documentazione dell'aggiornamento per la nuova versione del motore di database per verificare se vi sono problemi di compatibilità che potrebbero interessare il database o le applicazioni:
   - Changes in MySQL 5.6 (Modifiche in MySQL 5.6)
   - Changes in MySQL 5.7 (Modifiche in MySQL 5.7)
   - Changes in MySQL 8.0 (Modifiche in MySQL 8.0)
2. Se l'istanza database è un membro di un gruppo di parametri database personalizzato, crea un nuovo gruppo di parametri database con le impostazioni esistenti che sia compatibile con la nuova versione principale. Specifica il nuovo gruppo di parametri database quando aggiorni l'istanza di prova, di modo che il test dell'aggiornamento ne garantisca il corretto funzionamento. Per ulteriori informazioni sulla creazione di un gruppo di parametri database, consulta Utilizzo di gruppi di parametri.
3. Creare uno snapshot DB dell'istanza database da aggiornare. Per ulteriori informazioni, consulta Creazione di uno snapshot DB per un'istanza DB Single-AZ.

4. Ripristinare lo snapshot DB per creare una nuova istanza database di test. Per ulteriori informazioni, consulta Ripristino da uno snapshot database.
5. Modificare la nuova istanza database di prova per aggiornarla alla nuova versione, utilizzando uno dei metodi descritti di seguito. Se è stato creato un nuovo gruppo di parametri nel passaggio 2, specificare quel gruppo di parametri.
6. Valutare lo storage utilizzato dall'istanza aggiornata per determinare se l'aggiornamento richiede storage aggiuntivo.
7. Eseguire quanti più test di controllo qualità possibili per l'istanza database aggiornata come necessario per assicurare che il database e l'applicazione funzionino correttamente con la nuova versione. Implementare qualsiasi nuovo test necessario per valutare l'impatto di problemi di compatibilità identificati nella fase 1. Testare tutte le stored procedure e le funzioni. Indirizzare le versioni di test delle applicazioni all'istanza database aggiornata.
8. Se tutti i test vengono superati, eseguire l'aggiornamento nell'istanza database di produzione. Consigliamo di non consentire le operazioni di scrittura sull'istanza database fino alla conferma che tutto funzioni correttamente.

Aggiornamento di un'istanza database MySQL

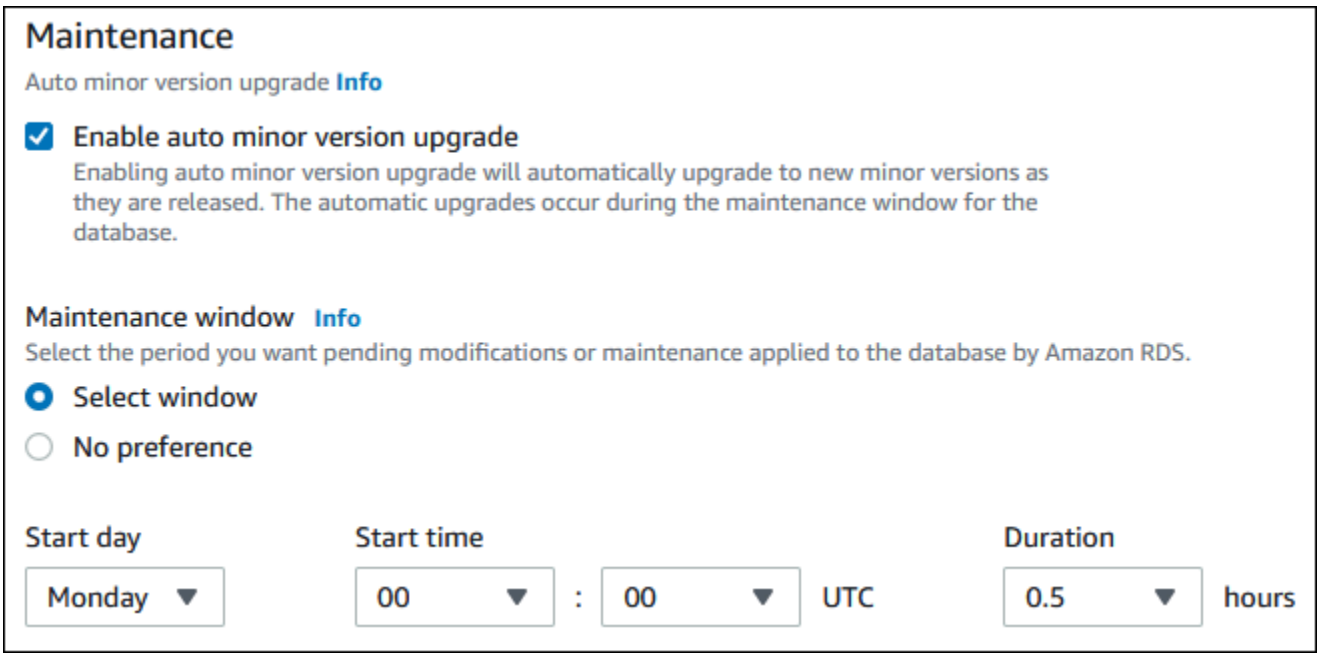
Per informazioni sull'aggiornamento manuale o automatico di un'istanza database MySQL, consulta Aggiornamento della versione del motore di un'istanza database.

Aggiornamenti a versioni secondarie automatiche per MySQL

Se specifichi le seguenti impostazioni durante la creazione o la modifica di un'istanza database, puoi decidere aggiornare automaticamente l'istanza database.

- L'impostazione di aggiornamento automatico della versione secondaria deve essere attivata.
- L'impostazione del periodo di conservazione del backup deve essere maggiore di 0.

In AWS Management Console, queste impostazioni si trovano in Configurazione aggiuntiva. L'immagine che segue mostra l’impostazione Auto Minor Version Upgrade (Aggiornamento automatico versione secondaria).



Per ulteriori informazioni su queste impostazioni, consultare Impostazioni per istanze database.

Per alcune versioni principali di RDS for MySQL, in Regioni AWS altre, una versione secondaria viene designata da RDS come versione di aggiornamento automatico. Una volta che una versione secondaria è stata testata e approvata da Amazon RDS, l'aggiornamento della versione secondaria avviene automaticamente nel corso della finestra di manutenzione. RDS non imposta mai automaticamente le nuove release secondarie come versione di aggiornamento automatico. Prima che RDS indichi una versione di aggiornamento automatico più recente, vengono considerati diversi livelli di valutazione, quali:

- Problemi di sicurezza noti
- Bug nella versione della community di MySQL
- Stabilità generale del parco istanze da quando la versione secondaria è stata rilasciata

È possibile utilizzare il seguente AWS CLI comando per determinare la versione di destinazione dell'aggiornamento secondario automatico corrente per una versione secondaria di MySQL specificata in uno specifico. Regione AWS

PerLinux, omacOS: Unix

```
aws rds describe-db-engine-versions \
--engine mysql \
--engine-version minor-version \
```

```
--region region \
--query "DBEngineVersions[*].ValidUpgradeTarget[*].
{AutoUpgrade:AutoUpgrade,EngineVersion:EngineVersion}" \
--output text
```

Per Windows:

```
aws rds describe-db-engine-versions ^
--engine mysql ^
--engine-version minor-version ^
--region region ^
--query "DBEngineVersions[*].ValidUpgradeTarget[*].
{AutoUpgrade:AutoUpgrade,EngineVersion:EngineVersion}" ^
--output text
```

Ad esempio, il AWS CLI comando seguente determina l'obiettivo di aggiornamento secondario automatico per la versione secondaria di MySQL 8.0.11 negli Stati Uniti orientali (Ohio) (us-east-2). Regione AWS

PermacOS, oUnix: Linux

```
aws rds describe-db-engine-versions \
--engine mysql \
--engine-version 8.0.11 \
--region us-east-2 \
--query "DBEngineVersions[*].ValidUpgradeTarget[*].
{AutoUpgrade:AutoUpgrade,EngineVersion:EngineVersion}" \
--output table
```

Per Windows:

```
aws rds describe-db-engine-versions ^
--engine mysql ^
--engine-version 8.0.11 ^
--region us-east-2 ^
--query "DBEngineVersions[*].ValidUpgradeTarget[*].
{AutoUpgrade:AutoUpgrade,EngineVersion:EngineVersion}" ^
--output table
```

L'output è simile a quello riportato di seguito.

```
---------------------------------
|    DescribeDBEngineVersions    |
+--------------+-----------------+
|  AutoUpgrade |  EngineVersion  |
+--------------+-----------------+
|  False       |  8.0.15         |
|  False       |  8.0.16         |
|  False       |  8.0.17         |
|  False       |  8.0.19         |
|  False       |  8.0.20         |
|  False       |  8.0.21         |
|  True        |  8.0.23         |
|  False       |  8.0.25         |
+--------------+-----------------+
```

In questo esempio, il valore `AutoUpgrade` è `True` per MySQL versione 8.0.23. Quindi, il target di aggiornamento secondario automatico è MySQL versione 8.0.23, che è evidenziato nell'output.

Un'istanza database MySQL viene aggiornata automaticamente durante la finestra di manutenzione se vengono soddisfatti i seguenti criteri:

- L'impostazione di aggiornamento automatico della versione secondaria deve essere attivata.
- L'impostazione del periodo di conservazione del backup deve essere maggiore di 0.
- L'istanza database esegue una versione motore database minore rispetto a una versione minore automatica dell'aggiornamento corrente.

Per ulteriori informazioni, consulta Aggiornamento automatico della versione secondaria del motore.

Utilizzo di una replica di lettura per ridurre i tempi di inattività durante l'aggiornamento di un database MySQL

Nella maggior parte dei casi, un'implementazione blu/verde è l'opzione migliore per ridurre i tempi di inattività durante l'aggiornamento di un'istanza database MySQL. Per ulteriori informazioni, consulta Utilizzo delle implementazioni blu/verde Amazon RDS per gli aggiornamenti del database.

Se non è possibile utilizzare un'implementazione blu/verde e l'istanza database MySQL è attualmente in uso con un'applicazione di produzione, puoi seguire la seguente procedura per aggiornare la versione del database per l'istanza database. Questa procedura permette di ridurre i tempi di indisponibilità dell'applicazione.

Utilizzando una replica di lettura, è possibile eseguire la maggior parte dei passaggi di manutenzione in anticipo e ridurre al minimo le modifiche necessarie durante l'interruzione effettiva. Con questa tecnica, è possibile testare e preparare la nuova istanza database senza apportare alcuna modifica all'istanza database esistente.

La seguente procedura mostra un esempio di aggiornamento da MySQL versione 5.7 a MySQL versione 8.0. Puoi utilizzare la stessa procedura generale per gli aggiornamenti ad altre versioni principali.

> **Note**
>
> Se esegui l'aggiornamento da MySQL versione 5.7 a MySQL versione 8.0, completa i controlli preliminari prima di eseguire l'aggiornamento. Per ulteriori informazioni, consulta Controlli preliminari per aggiornamenti da MySQL 5.7 a 8.0.

Per aggiornare un database MySQL con un'istanza database in uso

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Crea una replica di lettura dell'istanza database MySQL 5.7. Questo processo crea una copia aggiornabile del database. Potrebbero esistere già presenti altre repliche di lettura dell'istanza database.
   a. Nella console, seleziona Database e quindi l'istanza database da aggiornare.
   b. Per Actions (Operazioni), scegliere Create read replica (Crea replica di lettura).
   c. Specifica un valore per Identificativo istanza DB per la replica di lettura e assicurati che Classe di istanza database e altre impostazioni corrispondano all'istanza database MySQL 5.7.
   d. Scegliere Create read replica (Crea replica di lettura).
3. (Facoltativo) Una volta creata la replica di lettura e il campo Stato riporta Disponibile, converti la replica di lettura in una implementazione Multi-AZ e abilita i backup.

   Per impostazione predefinita, una replica di lettura viene creata come implementazione single-AZ con backup disabilitati. Poiché la replica di lettura diventerà in definitiva l'istanza database di produzione, è opportuno configurare un'implementazione multi-AZ e abilitare i backup in questo momento.

a. Nella console, seleziona Database, quindi seleziona la replica di lettura appena creata.

b. Scegliere Modify (Modifica).

c. Per Implementazione Multi-AZ, seleziona Crea istanza di standby.

d. In Backup Retention Period (Periodo di conservazione dei backup), seleziona un valore positivo diverso da zero, ad esempio 3 giorni, quindi scegli Continue (Continua).

e. In Scheduling of Modifications (Pianificazione delle modifiche), scegliere Apply immediately (Applica immediatamente).

f. Scegliere Modify DB Instance (Modifica istanza database).

4. Quando il campo Stato della replica di lettura riporta Disponibile, aggiorna la replica di lettura a MySQL 8.0:

a. Nella console, seleziona Database, quindi seleziona la replica di lettura appena creata.

b. Scegliere Modify (Modifica).

c. In Versione motore database seleziona la versione MySQL 8.0 da aggiornare, quindi scegli Continua.

d. In Scheduling of Modifications (Pianificazione delle modifiche), scegliere Apply immediately (Applica immediatamente).

e. Scegliere Modify DB Instance (Modifica istanza database) per avviare l'aggiornamento.

5. Quando l'aggiornamento è completo e lo stato mostra Disponibile, verifica che la replica di lettura aggiornata sia up-to-date con l'istanza database MySQL 5.7 di origine. Per verificare, connettiti alla replica di lettura ed esegui il comando `SHOW REPLICA STATUS`. Se il `Seconds_Behind_Master` campo è, allora la replica è`0`. up-to-date

> **Note**
>
> Versioni precedenti di MySQL utilizzate `SHOW SLAVE STATUS` al posto di `SHOW REPLICA STATUS`. Se si utilizza una versione MySQL prima della 8.0.23, utilizzare `SHOW SLAVE STATUS`.

6. (Facoltativo) Crea una replica di lettura della replica di lettura.

Se desideri che l'istanza database disponga di una replica di lettura dopo che è stata promossa a un'istanza database autonoma, puoi crearla in questo momento.

a. Nella console, seleziona Database, quindi scegli la replica di lettura appena aggiornata.

b. Per Actions (Operazioni), scegliere Create read replica (Crea replica di lettura).

c. Specifica un valore per Identificativo istanza DB per la replica di lettura e assicurati che Classe di istanza database e altre impostazioni corrispondano all'istanza database MySQL 5.7.

d. Scegliere Create read replica (Crea replica di lettura).

7. (Facoltativo) Configura un gruppo di parametri database personalizzato per la replica di lettura.

   Se desideri che l'istanza database utilizzi un gruppo di parametri personalizzato dopo che è stato promossa a un'istanza database autonoma, puoi creare il gruppo e associarlo alla replica di lettura.

   a. Crea un gruppo di parametri database personalizzato. Per istruzioni, consulta Creazione di un gruppo di parametri del database.

   b. Modifica i parametri che desideri modificare nel gruppo di parametri database appena creato. Per istruzioni, consulta Modifica di parametri in un gruppo di parametri del database.

   c. Nella console seleziona Database, quindi scegli la replica di lettura.

   d. Scegliere Modify (Modifica).

   e. Per il Gruppo di parametri database, sceglie il gruppo di parametri database MySQL 8.0 appena creato, quindi scegli Continua.

   f. In Scheduling of Modifications (Pianificazione delle modifiche), scegliere Apply immediately (Applica immediatamente).

   g. Scegliere Modify DB Instance (Modifica istanza database) per avviare l'aggiornamento.

8. Converti la replica di lettura MySQL 8.0 in un'istanza DB autonoma.

   > **Important**
   >
   > Quando promuovi la replica di lettura MySQL 8.0 a un'istanza database autonoma, la replica non sarà più dell'istanza database MySQL 5.7. Ti consigliamo di promuovere la replica di lettura MySQL 8.0 durante una finestra di manutenzione quando l'istanza database MySQL 5.7 di origine è in modalità di sola lettura e tutte le operazioni di scrittura sono sospese. Al termine dell'operazione, è possibile indirizzare le operazioni di scrittura all'istanza database MySQL 8.0 aggiornata per evitare la perdita di qualsiasi operazione di scrittura.
   > Inoltre, prima di promuovere la replica di lettura MySQL 8.0, ti consigliamo di eseguire tutte le operazioni DDL (Data Definition Language) necessarie sulla replica di lettura

MySQL 8.0. Un esempio di tale operazione è la creazione degli indici. Questo approccio consente di evitare qualsiasi effetto negativo sulle prestazioni della replica di lettura MySQL 8.0 dopo la promozione. Per promuovere una replica di lettura, utilizzare la procedura seguente.

a. Nella console, seleziona Database, quindi scegli la replica di lettura appena aggiornata.
b. In Actions (Operazioni), selezionare Promote (Promuovi).
c. Scegliere Yes (Sì) per abilitare backup automatizzati per l'istanza della replica di lettura. Per ulteriori informazioni, consulta Introduzione ai backup.
d. Scegli Continue (Continua).
e. Selezionare Promote read replica (Promuovi replica di lettura).

9. Ora si dispone di una versione aggiornata del database MySQL. A questo punto, puoi indirizzare le applicazioni alla nuova istanza database MySQL 8.0.

Aggiornamento di una versione del motore di snapshot MySQL DB

Con Amazon RDS puoi creare uno snapshot DB del volume di storage dell'istanza database MySQL. Quando crei uno snapshot DB, lo snapshot si basa sulla versione del motore utilizzata dall'istanza DB. Oltre ad aggiornare la versione del motore DB dell'istanza database, puoi anche aggiornare la versione del motore per gli snapshot DB. Per RDS for MySQL, è possibile aggiornare uno snapshot della versione 5.7 alla versione 8.0. È possibile aggiornare istantanee DB crittografate o non crittografate.

Le seguenti versioni supportano l'aggiornamento degli snapshot di MySQL DB:

- È possibile eseguire l'aggiornamento da RDS for MySQL snapshot versione 5.7.16 e versioni successive 5.7.
- È possibile eseguire l'aggiornamento a RDS for MySQL snapshot versione 8.0.28 e successive, ad eccezione delle versioni 8.0.29, 8.0.30 e 8.0.31.

  Non è possibile aggiornare le versioni 5.7.40, 5.7.41 e 5.7.42 alla versione 8.0.28, ma è possibile aggiornare queste versioni alla versione 8.0.32 e successive.

Dopo aver ripristinato uno snapshot DB aggiornato a una nuova versione del motore, verificare che l'aggiornamento abbia avuto esito positivo. Per maggiori informazioni sull'aggiornamento di una versione principale, consultare the section called “Aggiornamento del motore di database MySQL”. Per informazioni su come ripristinare uno snapshot DB, consulta the section called “Ripristino da uno snapshot database”.

> **Note**
>
> Non è possibile aggiornare le istantanee DB automatizzate create durante il processo di backup automatico.

È possibile aggiornare uno snapshot DB utilizzando AWS Management Console AWS CLI, o l'API RDS.

Console

Per aggiornare uno snapshot DB

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, selezionare Snapshots (Snapshot).
3. Scegli la snapshot da usare per l'aggiornamento.
4. Per Actions (Operazioni), scegliere Upgrade snapshot (Aggiorna snapshot). Viene visualizzata la pagina Upgrade snapshot (Aggiorna snapshot).
5. Scegli New engine version (Nuova versione del motore) per eseguire l'aggiornamento.
6. Scegliere Save changes (Salva modifiche) per aggiornare lo snapshot.

   Durante il processo di aggiornamento, tutte le operazioni dello snapshot sono disabilitate per lo snapshot database. Inoltre, lo stato dello snapshot DB passa da Disponibile a Aggiornamento e quindi diventa Attivo al termine. Se lo snapshot del DB non può essere aggiornato a causa di problemi di danneggiamento dello snapshot, lo stato cambia in Non disponibile. Non è possibile recuperare lo snapshot quando è in questo stato.

   > **Note**
   >
   > Se l'aggiornamento dello snapshot fallisce, lo snapshot viene riportato allo stato originario con la versione iniziale.

AWS CLI

Per aggiornare uno snapshot DB a una nuova versione del motore di database, usa il comando. AWS CLI modify-db-snapshot

Opzioni

- `--db-snapshot-identifier` – Identificatore dello snapshot DB da aggiornare. L'identificatore deve essere un Amazon Resource Name (ARN) univoco. Per ulteriori informazioni, consulta Utilizzo di Amazon Resource Name (ARN) in Amazon RDS.
- `--engine-version` – Versione del motore a cui aggiornare lo snapshot DB.

Example

Per LinuxmacOS, oUnix:

```
aws rds modify-db-snapshot \
    --db-snapshot-identifier my_db_snapshot \
    --engine-version new_version
```

Per Windows:

```
aws rds modify-db-snapshot ^
    --db-snapshot-identifier my_db_snapshot ^
    --engine-version new_version
```

API RDS

Per aggiornare uno snapshot DB a una nuova versione del motore di database, chiama l'operazione ModifyDBSnapshot dell'API RDS.

Parametri

- `DBSnapshotIdentifier` – Identificatore dello snapshot DB da aggiornare. L'identificatore deve essere un Amazon Resource Name (ARN) univoco. Per ulteriori informazioni, consulta Utilizzo di Amazon Resource Name (ARN) in Amazon RDS.
- `EngineVersion` – Versione del motore a cui aggiornare lo snapshot DB.

Importazione di dati in un'istanza database MySQL

Puoi utilizzare diverse tecniche per importare i dati in un'istanza database RDS for MySQL. L'approccio migliore dipende dall'origine e dalla quantità dei dati e dal fatto che l'importazione venga eseguita in modo occasionale o continuo. Se stai migrando un'applicazione insieme a tutti i suoi dati, dovrai valutare per quanto tempo il sistema può rimanere inattivo.

Panoramica

La tabella di seguito riporta le varie tecniche per importare i dati in un'istanza database RDS for MySQL.

Origine	Quantità di dati	Una volta o continua	Tempo di inattività delle applicazioni	Tecnica	Ulteriori informazioni
Database MySQL esistente in locale o su Amazon EC2	Qualsiasi	Una volta	Medio	Crea un backup del database locale, archivialo in Amazon S3 e quindi ripristina il file di backup in una nuova istanza database Amazon RDS che esegue MySQL.	Ripristino di un backup in un'istanza database MySQL
Qualsiasi database esistente	Qualsiasi	Una volta o continua	Minima	AWS Database Migration Service Utilizzatelo per migrare il database con tempi di inattività minimi e, per molti motori di database di database, continuare la replica continua.	Cos'è AWS Database Migration Service e Utilizzo di un database

Origine	Quantità di dati	Una volta o continua	Tempo di inattività delle applicazioni	Tecnica	Ulteriori informazioni
					compatibile con MySQL come destinazione per AWS DMS nella Guida per l'utente di AWS Database Migration Service
Istanza database MySQL esistente	Qualsiasi	Una volta o continua	Minima	Creare una replica di lettura per la replica continua. Promuovere la replica di lettura per la creazione una tantum di una nuova istanza database.	Uso delle repliche di lettura dell'istanza database

Origine	Quantità di dati	Una volta o continua	Tempo di inattività delle applicazioni	Tecnica	Ulteriori informazioni
Database MariaDB o MySQL esistente	Small	Una volta	Medio	Copiare i dati direttamente nell'istanza database MySQL utilizzando un'utilità a riga di comando.	Importazione di dati da un database MariaDB o MySQL esterno in un'istanza RDS per MariaDB o RDS per MySQL DB

Origine	Quantità di dati	Una volta o continua	Tempo di inattività delle applicazioni	Tecnica	Ulteriori informazioni
Dati non salvati in un database esistente	Medium	Una volta	Medio	Crea file flat e importali utilizzando istruzioni `LOAD DATA LOCAL INFILE` MySQL.	Importazione dei dati da qualsiasi origine a un'istanza database MariaDB o MySQL

Origine	Quantità di dati	Una volta o continua	Tempo di inattività delle applicazioni	Tecnica	Ulteriori informazioni
Database MariaDB o MySQL esistente in locale o su Amazon EC2	Qualsiasi	Continua	Minima	Configurare la replica utilizzando un database MariaDB o MySQL esistente come origine della replica.	Configurazione della replica della posizione del file di log binario con un'istanza di origine esterna. Importazione dei dati in un database Amazon RDS MariaDB o MySQL con tempi di inattività ridotti

> **Note**
>
> Il database di sistema `'mysql'` contiene le informazioni di autenticazione e autorizzazione necessarie per accedere all'istanza database e ai dati. L'eliminazione, la modifica, la ridenominazione o il troncamento di tabelle, dati o altro contenuto del database `'mysql'` nell'istanza database può causare un errore e rendere inaccessibili dati e istanza database. In tal caso, è possibile ripristinare l'istanza DB da un'istantanea utilizzando il comando. AWS CLI `restore-db-instance-from-db-snapshot` È possibile ripristinare l'istanza DB utilizzando il AWS CLI `restore-db-instance-to-point-in-time` comando.

Importazione delle considerazioni sui dati

Di seguito sono riportate informazioni tecniche aggiuntive relative al caricamento dei dati in MySQL. Tali informazioni sono indirizzate a utenti esperti, con una buona conoscenza dell'architettura server MySQL.

Log binario

Quando il log binario è attivo, il caricamento dei dati ha un impatto negativo sulle prestazioni e richiede spazio aggiuntivo su disco (fino a quattro volte maggiore) rispetto al caricamento degli stessi dati con il log binario disattivato. L'impatto sulle prestazioni e la quantità di spazio su disco richiesta è direttamente proporzionale alle dimensioni delle transazioni usate per il caricamento dei dati.

Dimensioni delle transazioni

Le dimensioni delle transazioni ricoprono un ruolo importante nel caricamento dei dati MySQL e incidono in modo significativo sull'utilizzo delle risorse e dello spazio su disco, sui tempi di ripristino dei processi e di ritorno alle attività e sul formato dell'input (file flat o SQL). Questa sezione descrive in che modo le dimensioni della transazione incidono sul log binario e spiega perché sia conveniente disattivare il logo binario durante il caricamento di grandi quantità di dati. Come detto in precedenza, i log binari vengono attivati e disattivati impostando il periodo di retention dei backup automatico di Amazon RDS. Un valore pari a zero disattiva il log binario, mentre qualsiasi altro valore lo attiva. Descriveremo anche l'impatto delle transazioni di grandi dimensioni su InnoDB e analizzeremo le motivazioni per cui è importante contenere le dimensioni delle transazioni.

Transazioni di piccole dimensioni

Nel caso delle transazioni di piccole dimensioni, i log binari raddoppiano il numero di scritture su disco richieste per il caricamento dei dati. Tale effetto può incidere molto negativamente sulle prestazioni di altre sessioni di database e allungare i tempi richiesti per il caricamento dei dati. Il calo prestazionale dipende in parte dalla velocità di caricamento, da altre attività del database in esecuzione durante il caricemento e dalla capacità dell'istanza database Amazon RDS.

I log binari consumano una quantità di spazio su disco approssimativamente pari alla quantità di dati caricati, fino a quando non viene effettuato il backup e i dati non vengono rimossi. Fortunatamente, Amazon RDS riduce questo problema al minimo mediante backup frequenti e la conseguente rimozione dei log binari.

Transazioni di grandi dimensioni

Le transazioni di grandi dimensioni triplicano la penalità IOPS e il consumo di spazio su disco se il log binario è attivo. Tale effetto è dovuto al riversamento della cache del log binario nel disco, con un conseguente consumo di spazio e un aumento dell'utilizzo dell'IO ad ogni scrittura. La cache non può essere scritta nel binlog fino a quando la transazione non è aggiornata o non viene eseguito il rollback, pertanto consuma una quantità di disco proporzionale alla quantità di dati caricati. Quando la transazione viene aggiornata, la cache deve essere copiata nel binlog, creando una terza copia dei dati all'interno del disco.

Per tale ragione, per il caricamento dei dati è necessario disporre di una quantità di spazio su disco tripla rispetto alla stessa attività eseguita con il log binario disattivato. Ad esempio, il caricamento di 10 GiB di dati con un'unica transazione richiede almeno 30 GiB di spazio su disco durante l'operazione: 10 GiB per la tabella + 10 GiB per la cache del log binario + 10 GiB per il log binario vero e proprio. Il file della cache rimane nel disco fino alla terminazione della sessione per cui è stato creato oppure fino a quando la sessione non riempie nuovamente la cache del log binario durante un'altra transazione. Il log binario deve restare nel disco fino al backup, per cui potrebbe passare diverso tempo prima che i 20 GiB aggiuntivi vengano resi di nuovo disponibili.

Se i dati sono caricati utilizzando LOAD DATA LOCAL INFILE, e il database deve essere recuperato da un backup eseguito prima del caricamento, verrà creata una copia ulteriore dei dati. Durante il recupero, MySQL estrae i dati dal log binario in un file flat. A quel punto, MySQL esegue LOAD DATA LOCAL INFILE, come nella transazione originale. Tuttavia, questa volta il file di input è locale rispetto al server del database. Continuando con l'esempio precedente, il recupero non potrà essere eseguito correttamente se non vi sono almeno 40 GiB di spazio disponibile su disco.

Disattivazione del log binario

Quando possibile, eseguire il caricamento di grandi dimensioni di dati con il log binario disattivato, per evitare di sovraccaricare le risorse e di occupare una quantità eccessiva di spazio su disco In Amazon RDS i log binari vengono disattivati semplicemente impostando il periodo di retention dei backup su zero. In questo caso, ti consigliamo di fare una snapshot DB dell'istanza database immediatamente prima di caricare i dati. In questo modo, se fosse necessario, potrai annullare rapidamente e con facilità tutte le modifiche apportate durante il caricamento.

Dopo il caricamento, imposta il periodo di retention dei backup su un valore appropriato, diverso da zero.

Non puoi impostare il periodo di retention dei backup su zero se l'istanza database è un'origine per le Repliche di lettura.

InnoDB

Le informazioni contenute in questa sezione spiegano perché è conveniente ridurre le dimensioni delle transazioni quando si utilizza InnoDB.

Annulla operazione

InnoDB genera annullamenti per supportare caratteristiche quali rollback delle transazioni e MVCC. L'annullamento viene salvato nello spazio tabella del sistema InnoDB (in genere ibdata1) e viene conservato fino a quando il thread di eliminazione non lo rimuove. Il thread di eliminazione non può procedere oltre l'annullamento della transazione attiva più vecchia, e viene quindi bloccato fino a quando la transazione non viene confermata o non completa un rollback. Se il database elabora altre transazioni durante il caricamento, tutti gli annullamenti si accumulano nello spazio tabella del sistema e non possono essere rimossi neanche in caso di conferma e se nessun altra transazione richiede l'annullamento per MVCC. In questa situazione, tutte le transazioni (incluse quelle di sola lettura) che accedono a righe modificate da qualsiasi transazione (non solo quella caricata) subiranno un rallentamento, perché saranno tutte sottoposte alla scansione da parte dell'annullamento che sarebbe stato eliminato se non fosse stato per la transazione il cui caricamento richiede un tempo lungo.

L'annullamento viene salvato nello spazio tabella del sistema, le cui dimensioni non si riducono mai. Per tale ragione, le transazioni di grandi quantità di dati possono causare l'aumento delle dimensioni dello spazio tabella del sistema, consumando spazio su disco che non può essere recuperato senza ricreare il database da zero.

Rollback

InnoDB è ottimizzato per le conferme. Il rollback di una transazione di grandi dimensioni può richiedere un tempo molto lungo. In alcuni casi, potrebbe essere più veloce eseguire un point-in-time ripristino o ripristinare un'istantanea del DB.

Formato dei dati di input

MySQL può accettare i dati in due forme: file flat e SQL. Questa sezione descrive i vantaggi e gli svantaggi di ciascun formato.

File flat

Caricare i file flat con LOAD DATA LOCAL INFILE può risultare il metodo più conveniente e rapido, se le dimensioni delle transazioni rimangono relativamente piccole. Rispetto al caricamento degli stessi dati con SQL, i file flat generano di solito un minore traffico di rete, con una riduzione dei costi di trasmissione, dei tempi di caricamento e del sovraccarico del database.

Transazione unica di grandi dimensioni

LOAD DATA LOCAL INFILE carica l'intero file flat come in un'unica transazione. Questa non è necessariamente una cosa negativa, al contrario, presenta una serie di vantaggi purché le dimensioni dei singoli file rimangano limitate:

- Capacità di ripristino – si può tenere facilmente traccia dei file caricati. In caso di problemi durante il caricamento, puoi riprendere l'operazione dal punto in cui era stata interrotta. Potrebbe essere necessario trasmettere nuovamente alcuni file a Amazon RDS, ma se le loro dimensioni sono piccole il tempo per la ritrasmissione sarà minimo.
- Caricamento dati in parallelo – se disponi di IOPS e larghezza di banda sufficienti per eseguire il caricamento con file singolo, lavorare in parallelo potrebbe aiutarti a risparmiare tempo.
- Ridurre la velocità di caricamento – se il caricamento produce effetti negativi sugli altri processi, puoi ridurne la velocità, aumentando l'intervallo fra i file.

Attenzione

I vantaggi offerti da LOAD DATA LOCAL INFILE diminuiscono rapidamente man mano che le dimensioni della transazione aumentano. Se non fosse possibile suddividere un set di dati voluminoso in parti più piccole, SQL potrebbe costituire una soluzione migliore.

SQL

SQL presenta un grande vantaggio rispetto ai file flat: consente di mantenere piccole le dimensioni delle transazioni. Tuttavia, SQL ha tempi di caricamento significativamente più lunghi rispetto ai file flat e, in caso di errore, può essere difficile determinare il punto da cui riprendere. Ad esempio, i file mysqldump non sono riavviabili. In caso di errore durante il caricamento di un file mysqldump, questo dovrà essere modificato o sostituito prima che sia possibile riprendere il caricamento. L'alternativa consiste nel ritornare al punto temporale precedente al caricamento e ripetere l'operazione dopo avere corretto la causa dell'errore.

Rilevamento dei checkpoint con snapshot Amazon RDS

Se devi eseguire un caricamento che richiede molte ore o addirittura giorni, non utilizzare i log binari potrebbe non essere un'idea particolarmente allettante, a meno che non ci sia la possibilità di rilevare periodicamente dei checkpoint. È proprio in queste situazioni che la caratteristica snapshot DB di Amazon RDS risulta particolarmente utile. Uno snapshot DB crea una copia point-in-time coerente dell'istanza del database che può essere utilizzata per ripristinare il database a quel momento dopo un arresto anomalo o un altro incidente.

Per creare un checkpoint è sufficiente eseguire unaa snapshot DB. Tutte le snapshot DB eseguite in precedenza possono essere rimosse senza ripercussioni sulla durata o sul tempo di ripristino.

Le snapshot sono rapide e l'aggiunta frequente di checkpoint non incide in modo significativo sui tempi di caricamento.

Riduzione dei tempi di caricamento

Di seguito sono riportati alcuni suggerimenti per ridurre i tempi di caricamento:

- Crea tutti gli indici secondari prima del caricamento. Se sei abituato a utilizzare altri database, questa operazione potrebbe apparire illogica. Quando aggiungi o modifichi un indice secondario, MySQL crea una nuova tabella con le modifiche, copia i dati dalla tabella esistente alla nuova ed elimina la tabella originale.
- Carica i dati nell'ordine della chiave primaria. Questa operazione risulta particolarmente utile con le tabelle InnoDB, perché consente di abbreviare i tempi di caricamento del 75–80 percento e di dimezzare le dimensioni dei file di dati.
- Disattiva le limitazioni relative alla chiave esterna (foreign_key_checks=0). Spesso, quando i file flat sono caricati con LOAD DATA LOCAL INFILE, questa operazione è obbligatoria. La disattivazione

dei controlli della chiave esterna in tutti i carichi ti permette di migliorare sensibilmente le prestazioni. Tuttavia, ricorda di attivare le limitazioni e verificare i dati dopo il caricamento.

- Esegui il caricamento in parallelo, a meno di non essere già in vicinanza di un limite di risorse. Se possibile, usa tabelle partitionate.
- Durante il caricamento con SQL utilizza inserimento con valori multipli per ridurre il carico dell'esecuzione delle istruzioni. Se utilizzi mysqldump, questa operazione viene eseguita in modo automatico.
- Riduci l'I/O del log InnoDB (innodb_flush_log_at_trx_commit=0)
- Se carichi i dati in un'istanza database che non include repliche di lettura, imposta il parametro sync_binlog su 0 durante il caricamento dei dati. Al termine del caricamento, reimposta il parametro sync_binlog su 1.
- Carica i dati prima di convertire l'istanza database in un'implementazione Multi-AZ. Tuttavia, se l'istanza database utilizza già un'implementazione Multi-AZ, non è consigliabile passare a un'implementazione Single-AZ per il caricamento dei dati, perché i vantaggi sarebbero minimi.

> **Note**
>
> Se utilizzi innodb_flush_log_at_trx_commit=0, InnoDB cancellerà is log ogni secondo, senza attendere la conferma. Tale impostazione velocizza sensibilmente il processo, ma in caso di errori potrebbe portare alla perdita di dati. Utilizza questa soluzione con cautela.

Argomenti

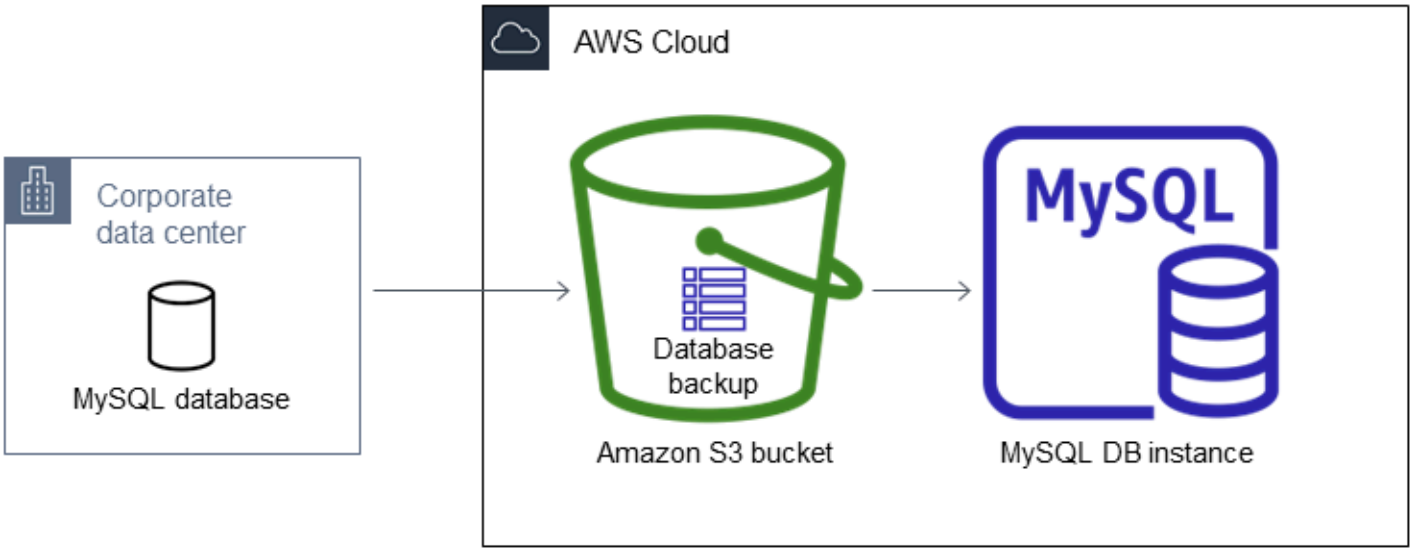
- Ripristino di un backup in un'istanza database MySQL
- Importazione di dati da un database MariaDB o MySQL esterno in un'istanza RDS per MariaDB o RDS per MySQL DB
- Importazione dei dati in un database Amazon RDS MariaDB o MySQL con tempi di inattività ridotti
- Importazione dei dati da qualsiasi origine a un'istanza database MariaDB o MySQL

Ripristino di un backup in un'istanza database MySQL

Amazon RDS supporta l'importazione di database MySQL utilizzando file di backup. Puoi creare un backup del database locale, archiviarlo in Amazon S3 e quindi ripristinare il file di backup in una nuova istanza database Amazon RDS che esegue MySQL.

Lo scenario descritto in questa sezione ripristina un backup di un database locale. Puoi utilizzare questa tecnica per database in altre sedi, come Amazon EC2 o servizi non AWS cloud, purché il database sia accessibile.

Lo scenario supportato è riportato nel seguente diagramma.



L'importazione di file di backup da Amazon S3 è supportata per MySQL in tutte le Regioni AWS.

Ti consigliamo di importare il database in Amazon RDS utilizzando i file di backup se è possibile che il database sia offline quando il file di backup viene creato, copiato e ripristinato. Se il database locale non può essere offline, puoi usare la replica del binlog per aggiornare il database dopo averne eseguito la migrazione in Amazon RDS tramite Amazon S3, come spiegato in questo argomento. Per ulteriori informazioni, consulta Configurazione della replica della posizione del file di log binario con un'istanza di origine esterna.. Per eseguire la migrazione del database ad Amazon RDS. puoi usare anche la AWS Database Migration Service . Per ulteriori informazioni, consulta Cos'è AWS Database Migration Service?

Limitazioni e consigli per l'importazione di file di backup da Amazon S3 ad Amazon RDS

Di seguito vengono riportate alcune limitazioni e consigli per l'importazione di file di backup da Amazon S3:

- Puoi importare i dati solo in una nuova istanza database, non in una istanza database esistente.
- È necessario utilizzare Percona XtraBackup per creare il backup del database locale.
- Non puoi importare dati da un'esportazione di snapshot DB in Amazon S3.
- Non puoi eseguire la migrazione da un database di origine con tabelle definite all'esterno della directory dei dati MySQL predefinita.
- Percona Server for MySQL non è supportato come database di origine perché può `compression_dictionary*` contenere tabelle nello schema. `mysql`
- È necessario importare i dati nella versione secondaria predefinita della versione principale di MySQL nella Regione AWS. Ad esempio, se la versione principale è MySQL 8.0 e la versione secondaria predefinita per la Regione AWS è 8.0.28, è necessario importare i dati in un'istanza database MySQL versione 8.0.28. È possibile aggiornare l'istanza DB dopo l'importazione. Per informazioni sulla determinazione della versione secondaria predefinita, vedere Versioni di MySQL in Amazon RDS.
- La migrazione alle versioni precedenti non è supportata per le versioni maggiori e minori. Ad esempio, non puoi eseguire la migrazione dalla versione 8.0 alla versione 5.7 e non puoi eseguire la migrazione dalla versione 8.0.32 alla versione 8.0.31.
- Non puoi importare un database MySQL 5.5 o 5.6.
- Non è possibile importare un database MySQL locale da una versione principale a un'altra. Ad esempio, non è possibile importare un database MySQL 5.7 in un database MySQL 8.0. Puoi aggiornare l'istanza database al termine dell'importazione.
- Non puoi eseguire il ripristino da un database di origine crittografato, ma puoi eseguire il ripristino in un'istanza database di Amazon RDS crittografata.
- Non è possibile ripristinare da un backup crittografato nel bucket Amazon S3.
- Non puoi eseguire il ripristino da un bucket Amazon S3 in una Regione AWS diversa da quella dell'istanza database di Amazon RDS.
- L'importazione da Amazon S3 non è supportata sulla classe istanza database db.t2.micro. Tuttavia, puoi eseguire il ripristino in una classe istanza database diversa e modificare la classe di istanza in seguito. Per ulteriori informazioni sulle classi di istanza, consulta Specifiche hardware per le classi di istanza database .

- Amazon S3 limita la dimensione del file caricato in un bucket Amazon S3 a 5 TB. Se un file di backup supera i 5 TB, devi dividerlo in file più piccoli.
- Quando ripristini il database, il backup viene copiato e quindi estratto sull'istanza database. Di conseguenza, il provisioning dello spazio di storage per l'istanza database è uguale o superiore alla somma delle dimensioni del backup, più la dimensione del database originale su disco.
- Amazon RDS limita il numero di file caricati in un bucket Amazon S3 a 1 milione. Se i dati di backup del database, inclusi tutti i backup completi e incrementali, superano 1 milione di file, utilizza un file Gzip (.gz), tar (.tar.gz) o Percona xbstream (.xbstream) per memorizzare i file dei backup completi e incrementali nel bucket Amazon S3. Percona XtraBackup 8.0 supporta solo Percona xbstream per la compressione.
- Gli account utente non vengono importati automaticamente. Salva gli account utente dal database di origine e aggiungili nella nuova istanza database in n seguito.
- Le funzioni non vengono importate automaticamente. Salva le funzioni dal database di origine e aggiungile nella nuova istanza database in seguito.
- Le stored procedure non vengono importate automaticamente. Salva le stored procedure dal database di origine e aggiungile nella nuova istanza database in seguito.
- Le informazioni sul fuso orario non vengono importate automaticamente. Registra le informazioni sul fuso orario per il database di origine e imposta il fuso orario della nuova istanza database in seguito. Per ulteriori informazioni, consulta Fuso orario locale per le istanze database MySQL.
- Il parametro `innodb_data_file_path` deve essere configurato con un solo file di dati che utilizza il nome di file di dati predefinito `"ibdata1:12M:autoextend"`. I database con due file di dati o con un file di dati con un nome diverso non possono essere migrati utilizzando questo metodo.

  Di seguito sono riportati esempi di nomi di file che non sono permessi: `"innodb_data_file_path=ibdata1:50M; ibdata2:50M:autoextend"` e `"innodb_data_file_path=ibdata01:50M:autoextend"`.
- La dimensione massima del database ripristinato è la dimensione massima del database supportata meno la dimensione del backup. Pertanto, se la dimensione massima del database supportata è 64 TiB e la dimensione del backup è 30 TiB, la dimensione massima del database ripristinato è 34 TiB, come nell'esempio seguente:

  ```
  64 TiB - 30 TiB = 34 TiB
  ```

  Per informazioni sulle dimensioni massime del database supportate da Amazon RDS for MySQL, vedere Storage SSD per scopi generici e Storage SSD Provisioned IOPS.

Panoramica della configurazione per l'importazione di file di backup da Amazon S3 ad Amazon RDS

Di seguito sono elencati i componenti che devi configurare per importare file di backup da Amazon S3 ad Amazon RDS:

- Un bucket Amazon S3 per archiviare i file di backup.
- Un backup del database locale creato da Percona. XtraBackup
- Un ruolo AWS Identity and Access Management (IAM) per consentire ad Amazon RDS di accedere al bucket.

Se disponi già di un bucket Amazon S3, puoi utilizzarlo. Se non hai già un bucket Amazon S3, puoi creane uno nuovo. Se intendi creare un nuovo bucket, consulta Creazione di un bucket.

Usa lo XtraBackup strumento Percona per creare il tuo backup. Per ulteriori informazioni, consulta Creazione del backup di database.

Se disponi già di un ruolo IAM, puoi utilizzarlo. Se non hai già un ruolo IAM, puoi creane uno nuovo manualmente. In alternativa, puoi scegliere la creazione automatica di nuovo ruolo IAM nel tuo account tramite la procedura guidata quando ripristini il database usando la AWS Management Console. Se vuoi creare un nuovo ruolo IAM manualmente o collegare policy di trust e di autorizzazioni a un ruolo IAM esistente, consulta Creazione di un ruolo IAM manualmente. Se vuoi che venga creato automaticamente un nuovo ruolo IAM, segui la procedura in Console.

Creazione del backup di database

Usa il XtraBackup software Percona per creare il tuo backup. Ti consigliamo di utilizzare l'ultima versione di XtraBackup Percona. Puoi installare Percona XtraBackup da Download Percona. XtraBackup

> **Warning**
>
> Durante la creazione di un backup del database, XtraBackup potrebbe salvare le credenziali nel file xtrabackup_info. Assicurati di esaminare quel file in modo che l'impostazione `tool_command` non contenga informazioni sensibili.

> **Note**
>
> Per la migrazione a MySQL 8.0, è necessario utilizzare Percona 8.0. XtraBackup Percona XtraBackup 8.0.12 e versioni successive supportano la migrazione di tutte le versioni di MySQL. Se stai migrando a RDS for MySQL 8.0.20 o versioni successive, devi usare Percona 8.0.12 o versioni successive. XtraBackup
> Per le migrazioni MySQL 5.7, puoi anche usare Percona 2.4. XtraBackup Per le migrazioni di versioni precedenti di MySQL, puoi anche usare XtraBackup Percona 2.3 o 2.4.

È possibile creare un backup completo dei file del database MySQL utilizzando Percona. XtraBackup In alternativa, se utilizzi già Percona XtraBackup per eseguire il backup dei file del database MySQL, puoi caricare le directory e i file di backup completi e incrementali esistenti.

Per ulteriori informazioni sul backup del database con Percona XtraBackup, consulta Percona XtraBackup - documentazione e The xtrabackup binary sul sito Web di Percona.

Creazione di un backup completo con Percona XtraBackup

Per creare un backup completo dei file del database MySQL che possono essere ripristinati da Amazon S3, usa l'utilità XtraBackup Percona `xtrabackup` () per eseguire il backup del database.

Ad esempio, il comando seguente consente di creare un backup di un database MySQL e memorizzare i file nella cartella `/on-premises/s3-restore/backup`.

```
xtrabackup --backup --user=<myuser> --password=<password> --target-dir=</on-premises/
s3-restore/backup>
```

Se desideri comprimere il backup in un singolo file (che può essere diviso in seguito, se necessario), puoi salvare il backup in uno dei seguenti formati:

- Gzip (.gz)
- tar (.tar)
- Percona xbstream (.xbstream)

> **Note**
>
> Percona XtraBackup 8.0 supporta solo Percona xbstream per la compressione.

Il comando seguente consente di creare un backup del database MySQL diviso in più file Gzip.

```
xtrabackup --backup --user=<myuser> --password=<password> --stream=tar \
   --target-dir=</on-premises/s3-restore/backup> | gzip - | split -d --bytes=500MB \
   - </on-premises/s3-restore/backup/backup>.tar.gz
```

Il comando seguente consente di creare un backup del database MySQL diviso in più file tar.

```
xtrabackup --backup --user=<myuser> --password=<password> --stream=tar \
   --target-dir=</on-premises/s3-restore/backup> | split -d --bytes=500MB \
   - </on-premises/s3-restore/backup/backup>.tar
```

Il comando seguente consente di creare un backup del database MySQL diviso in più file xbstream.

```
xtrabackup --backup --user=<myuser> --password=<password> --stream=xbstream \
   --target-dir=</on-premises/s3-restore/backup> | split -d --bytes=500MB \
   - </on-premises/s3-restore/backup/backup>.xbstream
```

> **Note**
>
> Se viene visualizzato il seguente errore, potrebbe essere causato dalla combinazione di formati di file nel comando:
>
> ```
> ERROR:/bin/tar: This does not look like a tar archive
> ```

Utilizzo di backup incrementali con Percona XtraBackup

Se utilizzi già Percona XtraBackup per eseguire backup completi e incrementali dei file del tuo database MySQL, non è necessario creare un backup completo e caricare i file di backup su Amazon S3. Puoi, invece, risparmiare tempo copiando le directory e i file di backup esistenti nel bucket Amazon S3. Per ulteriori informazioni sulla creazione di backup incrementali con Percona, consulta Backup incrementale. XtraBackup

Durante la copia dei file del backup completo e incrementale in un bucket Amazon S3, devi copiare in modo ricorsivo i contenuti della directory di base. Questi contenuti includono il backup completo e anche tutte le directory e i file del backup incrementale. Questa copia deve mantenere la struttura di directory nel bucket Amazon S3. Amazon RDS esegue l'iterazione di tutti i file e le directory. Amazon

RDS usa il file `xtrabackup-checkpoints` incluso con ogni backup incrementale per identificare la directory di base e ordinare i backup incrementali in base all'intervallo dei numeri di sequenza log (LSN).

Considerazioni sul backup per Percona XtraBackup

Amazon RDS consuma i file di backup in base al nome del file. Assegnare un nome ai file di backup con l'estensione file appropriata in base al formato, —ad esempio, `.xbstream` per i file archiviati tramite il formato xbstream di Percona.

Amazon RDS consuma i file di backup in ordine alfabetico e anche in ordine numerico naturale. Utilizza l'opzione `split` quando invii il comando `xtrabackup` per assicurarti che i file di backup vengano scritti e denominati nell'ordine corretto.

Amazon RDS non supporta backup parziali creati con Percona. XtraBackup Non puoi usare le opzioni seguenti per creare un backup parziale quando esegui il backup dei file di origine per il database: `--tables`, `--tables-exclude`, `--tables-file`, `--databases`, `--databases-exclude` o `--databases-file`.

Amazon RDS supporta backup incrementali creati con Percona. XtraBackup Per ulteriori informazioni sulla creazione di backup incrementali utilizzando Percona, consulta Backup incrementale. XtraBackup

Creazione di un ruolo IAM manualmente

Se non hai già un ruolo IAM, puoi creane uno nuovo manualmente. In alternativa, puoi scegliere la creazione automatica di un nuovo ruolo IAM tramite la procedura guidata quando ripristini il database usando la AWS Management Console. Se vuoi che venga creato automaticamente un nuovo ruolo IAM, segui la procedura in Console.

Per creare manualmente un nuovo ruolo IAM per importare il database da Amazon S3, crea un ruolo per delegare le autorizzazioni da Amazon RDS al bucket Amazon S3. Quando crei un ruolo IAM, vengono collegate le policy di attendibilità e autorizzazione. Per importare i file di backup da Amazon S3, utilizza policy di attendibilità e autorizzazione simili agli esempi seguenti. Per ulteriori informazioni sulla creazione del ruolo, vedere Creazione di un ruolo per delegare le autorizzazioni a un servizio. AWS

In alternativa, puoi scegliere la creazione automatica di un nuovo ruolo IAM tramite la procedura guidata quando ripristini il database usando la AWS Management Console. Se vuoi che venga creato automaticamente un nuovo ruolo IAM, segui la procedura in Console

Le policy di attendibilità e autorizzazione richiedono che venga fornito un Amazon Resource Name (ARN). Per ulteriori informazioni sulla formattazione ARN, consulta Amazon Resource Names (ARNs) e service namespace. AWS

Example Policy di attendibilità per l'importazione da Amazon S3

```
{
    "Version": "2012-10-17",
    "Statement":
    [{
        "Effect": "Allow",
        "Principal": {"Service":  "rds.amazonaws.com"},
        "Action": "sts:AssumeRole"
    }]
}
```

Example Policy di autorizzazioni per l'importazione da Amazon S3 — Autorizzazioni utente IAM

```
{
    "Version":"2012-10-17",
    "Statement":
    [
        {
            "Sid":"AllowS3AccessRole",
            "Effect":"Allow",
            "Action":"iam:PassRole",
            "Resource":"arn:aws:iam::IAM User ID:role/S3Access"
        }
    ]
}
```

Example Policy di autorizzazioni per l'importazione da Amazon S3 — Autorizzazioni dei ruoli

```
{
    "Version": "2012-10-17",
    "Statement":
    [
        {
        "Effect": "Allow",
        "Action":
            [
                "s3:ListBucket",
```

```
                "s3:GetBucketLocation"
            ],
        "Resource": "arn:aws:s3:::bucket_name"
        },
        {
        "Effect": "Allow",
        "Action":
            [
                "s3:GetObject"
            ],
        "Resource": "arn:aws:s3:::bucket_name/prefix*"
        }
    ]
}
```

> **Note**
>
> Se includi un prefisso del nome file, aggiungi l'asterisco (*) dopo il prefisso. Se non intendi specificare un prefisso, specifica solo un asterisco.

Importazione di dati da Amazon S3 in una nuova istanza database MySQL

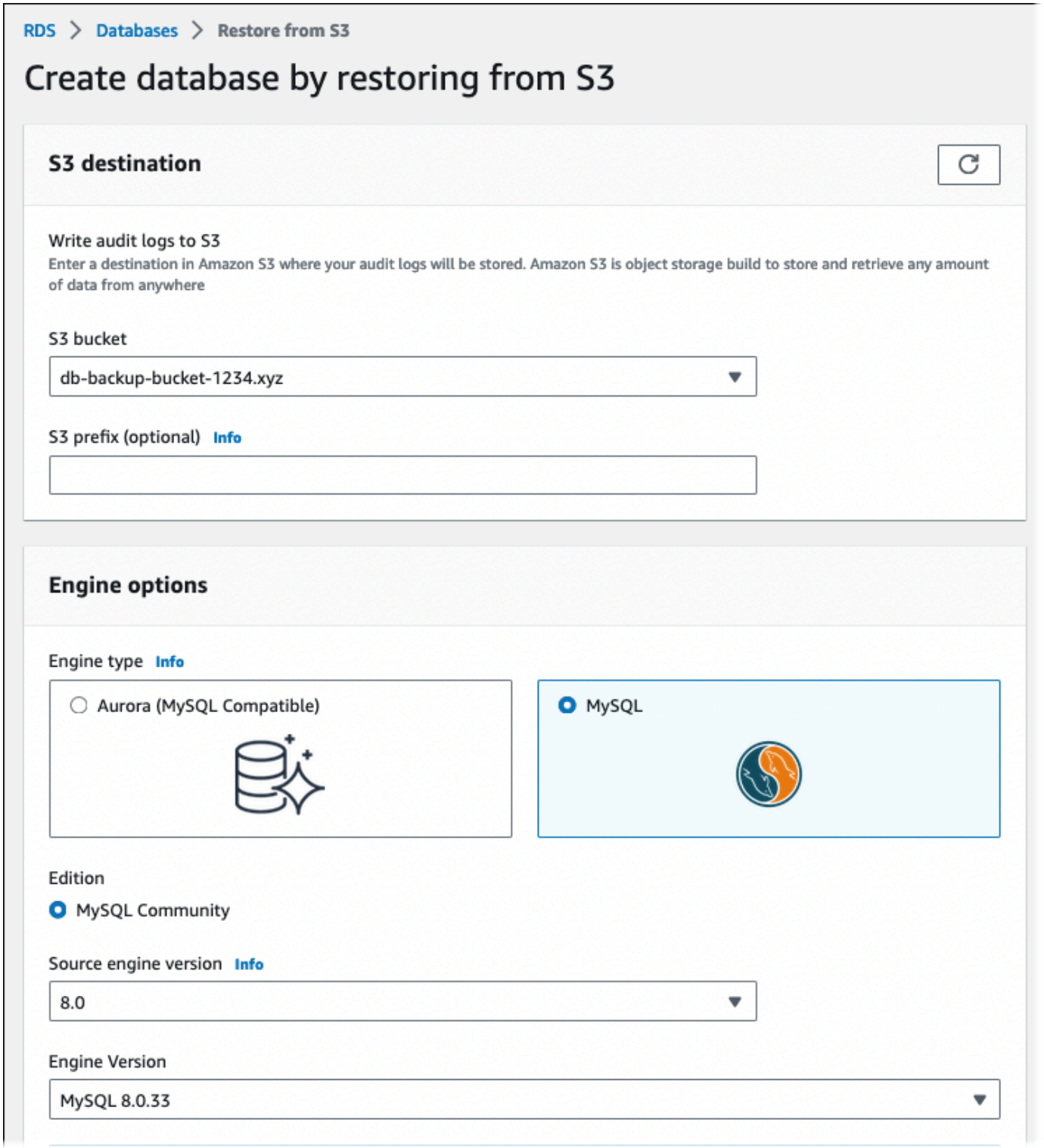
Puoi importare dati da Amazon S3 in una nuova istanza DB MySQL utilizzando l' AWS Management Console API, o RDS. AWS CLI

Console

Per importare dati da Amazon S3 in una nuova istanza database MySQL

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nell'angolo in alto a destra della console Amazon RDS, scegli l'istanza database Regione AWS in cui creare la tua istanza DB. Scegli lo Regione AWS stesso bucket Amazon S3 che contiene il backup del database.
3. Nel riquadro di navigazione, scegliere Databases (Database).
4. Seleziona Ripristina da S3.

   Sarà visualizzata la pagina Crea database ripristinando da S3 .



5. In Destinazione S3:

   a. Seleziona il bucket S3 che contiene il backup.

b. (Facoltativo) Per S3 folder path prefix (Prefisso percorso cartella S3) inserire un prefisso del percorso per i file archiviati nel bucket Amazon S3.

   Se non si specifica un prefisso, RDS crea l'istanza database utilizzando tutti i file e le cartelle nella cartella root del bucket S3. Se si specifica un prefisso, RDS crea l'istanza database utilizzando tutti i file e le cartelle nel bucket S3 in cui il percorso del file inizia con il prefisso specificato.

   Ad esempio, si supponga di archiviare i file di backup su S3 in una sottocartella denominata backups e di disporre di più set di file di backup, ciascuno nella sua directory (gzip_backup1, gzip_backup2 e così via). In questo caso, specificare un prefisso di backups/gzip_backup1 per eseguire il ripristino dai file nella cartella gzip_backup1.

6. In Opzioni motore:

   a. Per Tipo di motore, seleziona MySQL.

   b. In Versione motore di origine, seleziona la versione MySQL del database di origine.

   c. Per Version (Versione), scegli la versione secondaria predefinita della versione principale di MySQL nella Regione AWS.

      In AWS Management Console, è disponibile solo la versione secondaria predefinita. È possibile aggiornare l'istanza DB dopo l'importazione.

7. Per Ruolo IAM, puoi scegliere un ruolo IAM esistente.

8. (Facoltativo) È inoltre possibile creare un nuovo ruolo IAM scegliendo Crea un nuovo ruolo e immettendo il nome del ruolo IAM.

9. Specifica le informazioni sull'istanza database. Per informazioni su ciascuna impostazione, consulta Impostazioni per istanze database.

   > **Note**
   >
   > Assicurati di allocare memoria sufficiente per la nuova istanza database in modo che il ripristino vada a buon fine.
   > Puoi inoltre scegliere l'opzione Abilita dimensionamento automatico dello storage per consentire automaticamente la crescita futura.

10. Scegliere impostazioni aggiuntive in base alle esigenze.

11. Scegliere Create database (Crea database).

AWS CLI

Per importare dati da Amazon S3 in una nuova istanza DB MySQL utilizzando il AWS CLI, chiama il comando restore-db-instance-from-s3 con i seguenti parametri. Per informazioni su ciascuna impostazione, consulta Impostazioni per istanze database.

> **Note**
>
> Assicurati di allocare memoria sufficiente per la nuova istanza database in modo che il ripristino vada a buon fine.
> Puoi inoltre utilizzare il parametro `--max-allocated-storage` per abilitare il dimensionamento automatico dello storage e consentire automaticamente una crescita futura.

- `--allocated-storage`
- `--db-instance-identifier`
- `--db-instance-class`
- `--engine`
- `--master-username`
- `--manage-master-user-password`
- `--s3-bucket-name`
- `--s3-ingestion-role-arn`
- `--s3-prefix`
- `--source-engine`
- `--source-engine-version`

Example

Per, o: Linux macOS Unix

```
aws rds restore-db-instance-from-s3 \
    --allocated-storage 250 \
    --db-instance-identifier myidentifier \
    --db-instance-class db.m5.large \
    --engine mysql \
    --master-username admin \
    --manage-master-user-password \
```

```
    --s3-bucket-name mybucket \
    --s3-ingestion-role-arn arn:aws:iam::account-number:role/rolename \
    --s3-prefix bucketprefix \
    --source-engine mysql \
    --source-engine-version 8.0.32 \
    --max-allocated-storage 1000
```

Per Windows:

```
aws rds restore-db-instance-from-s3 ^
    --allocated-storage 250 ^
    --db-instance-identifier myidentifier ^
    --db-instance-class db.m5.large ^
    --engine mysql ^
    --master-username admin ^
    --manage-master-user-password ^
    --s3-bucket-name mybucket ^
    --s3-ingestion-role-arn arn:aws:iam::account-number:role/rolename ^
    --s3-prefix bucketprefix ^
    --source-engine mysql ^
    --source-engine-version 8.0.32 ^
    --max-allocated-storage 1000
```

API RDS

Per importare dati da Amazon S3 in una nuova istanza DB MySQL utilizzando l'API Amazon RDS, chiama l'operazione RestoreDB S3. InstanceFrom

Importazione di dati da un database MariaDB o MySQL esterno in un'istanza RDS per MariaDB o RDS per MySQL DB

In alternativa, puoi importare i dati da un database MariaDB o MySQL esistente a un'istanza database MySQL o MariaDB. A questo scopo, copia il database con mysqldump e reindirizzalo direttamente nell'istanza database MariaDB o MySQL. L'utility a riga di comando `mysqldump` viene spesso usata per creare backup e trasferire dati da un server MariaDB o MySQL a un altro. ed è inclusa nel software del client MySQL e MariaDB.

> **Note**
>
> Se stai importando o esportando grandi quantità di dati con un'istanza DB MySQL, è più affidabile e veloce spostare i dati da e verso Amazon RDS utilizzando file di backup e

Amazon S3. `xtrabackup` Per ulteriori informazioni, consulta Ripristino di un backup in un'istanza database MySQL.

Un tipico comando `mysqldump` per spostare dati da un database esterno a un'istanza database Amazon RDS è simile al seguente.

```
mysqldump -u local_user \
    --databases database_name \
    --single-transaction \
    --compress \
    --order-by-primary  \
    -plocal_password | mysql -u RDS_user \
        --port=port_number \
        --host=host_name \
        -pRDS_password
```

Important

Assicurati di non lasciare spazi tra l'opzione `-p` e la password immessa.
Specifica credenziali diverse dai prompt mostrati qui come best practice per la sicurezza.

Assicurati di essere a conoscenza dei seguenti suggerimenti e considerazioni:

- Escludi gli schemi seguenti dal file dump: `sys`, `performance_schema` e `information_schema`. Per impostazione predefinita, l'utility `mysqldump` esclude questi schemi.
- Se devi migrare utenti e privilegi, prendi in considerazione l'utilizzo di uno strumento che genera il linguaggio di controllo dei dati (DCL) per ricrearli, come l'utilità. pt-show-grants
- L'utente che esegue l'importazione deve avere accesso all'istanza database. Per ulteriori informazioni, consulta Controllo dell'accesso con i gruppi di sicurezza.

I parametri utilizzati sono i seguenti:

- `-u local_user` – Specifica un nome utente. La prima volta che usi questo parametro, devi specificare il nome di un account utente nel database MariaDB o MySQL locale, identificato dal parmetro `--databases`.

- `--databases` *`database_name`* – Specifica il nome del database nell'istanza database MariaDB o MySQL locale che vuoi importare in Amazon RDS.
- `--single-transaction` – Verifica che tutti i dati caricati dal database locale siano coerenti a un singolo punto temporale. Nel caso in cui vi siano altri processi che modificano i dati mentre `mysqldump` li legge, l'uso di questo parametro aiuta a preservare l'integrità dei dati.
- `--compress` – Riduce il consumo della larghezza di banda di rete comprimendo i dati dal database locale prima di inviarli ad Amazon RDS.
- `--order-by-primary` – Riduce il tempo di caricamento ordinando i dati di ogni tabella in base alla chiave primaria.
- `-p`*`local_password`* – Specifica una password. La prima volta che usi questo parametro, devi specificare la password per l'account utente identificato dal primo parametro `-u`.
- `-u` *`RDS_user`* – Specifica un nome utente. La seconda volta che usi questo parametro, devi specificare il nome di un account utente nel database predefinito per l'istanza database MariaDB o MySQL identificata dal parametro `--host`.
- `--port` *`port_number`* – Specifica la porta per l'istanza database MariaDB o MySQL. Il valore predefinito è 3306, ma può essere modificato al momento della creazione dell'istanza.
- `--host` *`host_name`* – Specifica il nome del sistema dei nomi di dominio (DNS) dall'endpoint dell'istanza database Amazon RDS, ad esempio `myinstance.123456789012.us-east-1.rds.amazonaws.com`. Puoi trovare il valore dell'endpoint è disponibile nei dettagli dell'istanza, nella console di gestione Amazon RDS.
- `-p`*`RDS_password`* – Specifica una password. La seconda volta che usi questo parametro, devi specificare la password per l'account utente identificato dal secondo parametro `-u`.

Eventuali procedure, trigger, funzioni o eventi devono essere creati manualmente nel database Amazon RDS. Se il database da copiare dovesse contenere questi tipi di oggetti, dovrai escluderli al momento di eseguire `mysqldump`. Per farlo, includi i seguenti parametri obbligatori con il tuo comando `mysqldump`: `--routines=0 --triggers=0 --events=0`.

Nell'esempio seguente viene copiato il database di esempio `world` sull'host locale in un'istanza database MySQL.

PerLinux, o: macOS Unix

```
sudo mysqldump -u localuser \
    --databases world \
```

```
    --single-transaction \
    --compress \
    --order-by-primary  \
    --routines=0 \
    --triggers=0 \
    --events=0 \
    -plocalpassword | mysql -u rdsuser \
        --port=3306 \
        --host=myinstance.123456789012.us-east-1.rds.amazonaws.com \
        -prdspassword
```

Per Windows, esegui il comando seguente in un prompt dei comandi che viene aperto facendo clic con il pulsante destro del mouse su Command Prompt (Prompt dei comandi) del menu dei programmi di Windows e selezionando Run as administrator (Esegui come amministratore):

```
mysqldump -u localuser ^
    --databases world ^
    --single-transaction ^
    --compress ^
    --order-by-primary  ^
    --routines=0 ^
    --triggers=0 ^
    --events=0 ^
    -plocalpassword | mysql -u rdsuser ^
        --port=3306 ^
        --host=myinstance.123456789012.us-east-1.rds.amazonaws.com ^
        -prdspassword
```

> **Note**
>
> Specifica credenziali diverse dai prompt mostrati qui come best practice per la sicurezza.

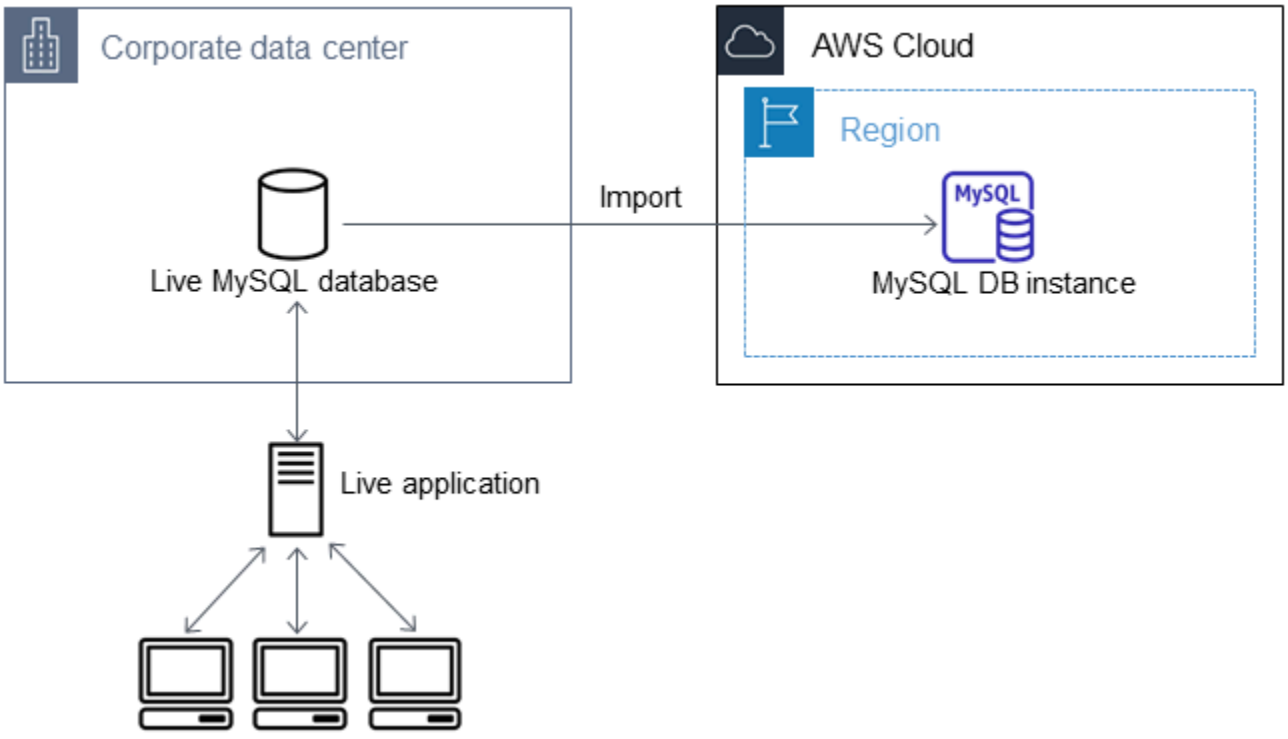
Importazione dei dati in un database Amazon RDS MariaDB o MySQL con tempi di inattività ridotti

In alcuni casi, potrebbe essere necessaria l'importazione dei dati da un database MariaDB o MySQL esterno che supporti un'applicazione live per un'istanza database MariaDB o MySQL oppure per un cluster database multi-AZ MySQL. Usa la seguente procedura per ridurre l'impatto sulla disponibilità delle applicazioni. Questa procedura può risultare utile anche quando utilizzi un database

di dimensioni particolarmente elevate. Utilizzando questa procedura, è possibile ridurre il costo dell'importazione riducendo la quantità di dati trasmessi attraverso la rete AWS.

Con questa procedura trasferisci una copia dei dati del database in un'istanza Amazon EC2 e li importi in un nuovo database Amazon RDS. Utilizza quindi la replica per portare il database Amazon RDS up-to-date con la tua istanza esterna attiva, prima di reindirizzare l'applicazione al database Amazon RDS. La replica MariaDB viene configurata in base agli identificatori globali di transazione (GTID) se l'istanza esterna è MariaDB 10.0.24 o versioni successive e l'istanza di destinazione è RDS per MariaDB. In alternativa, è possibile configurare la replica in base alle coordinate del log binario. Si consiglia la replica basata su GTID se supportata dal database esterno perché è un metodo più affidabile. Per ulteriori informazioni, consulta ID globali di transazione nella documentazione di MariaDB.

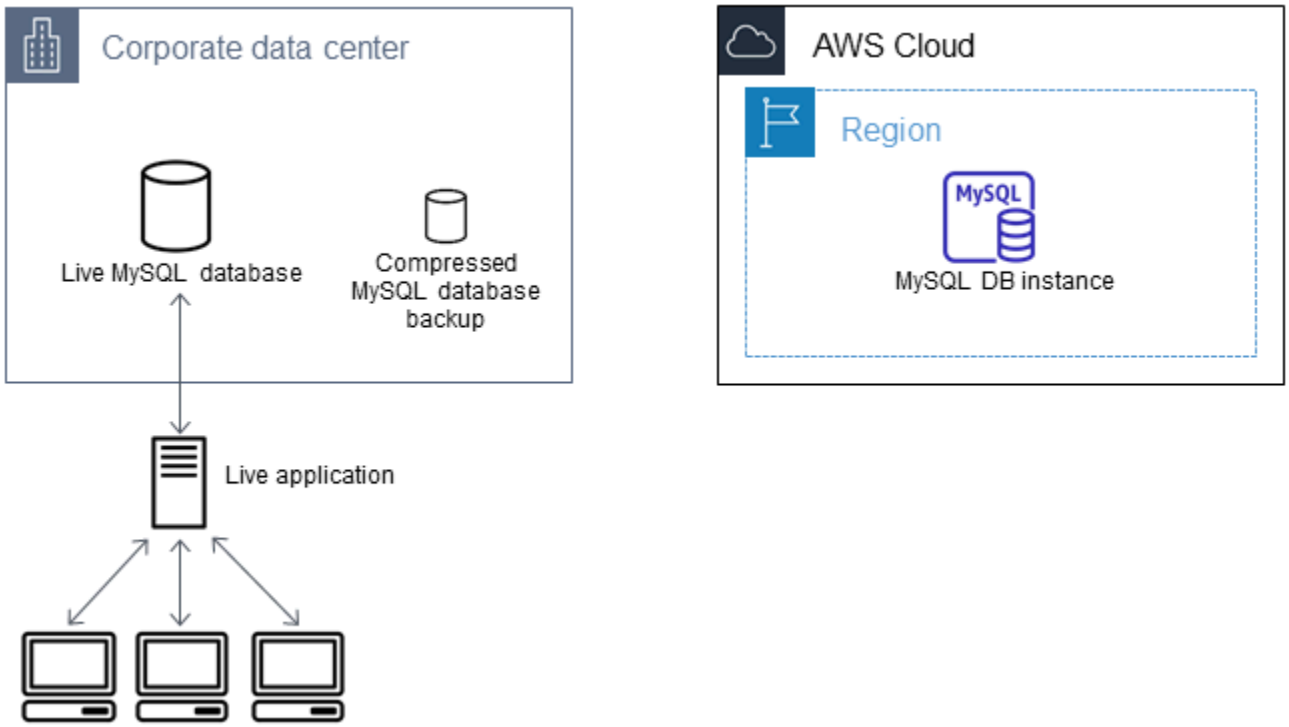
> **Note**
>
> Per importare i dati in un'istanza database MySQL e lo scenario supporta questo approccio, si consiglia di spostare dati da e verso Amazon RDS usando i file di backup e Amazon S3. Per ulteriori informazioni, consulta Ripristino di un backup in un'istanza database MySQL.



> **Note**
>
> Non è consigliabile utilizzare questa procedura per i database MySQL di origine con versioni di MySQL precedenti alla 5.5 a causa dei potenziali problemi di replica. Per ulteriori informazioni, consulta Compatibilità delle repliche fra le versioni di MySQL nella documentazione di MySQL.

Creazione di una copia del database esistente

La prima fase per eseguire la migrazione di grandi quantità di dati in un database RDS per MariaDB o RDS per MySQL riducendo al minimo i tempi di inattività consiste nella creazione di una copia dei dati di origine.



Puoi utilizzare l'utility mysqldump per creare un backup del database in formato SQL o come testo delimitato. Consigliamo di eseguire un test con ciascun formato, fuori dall'ambiente di produzione, per capire quale metodo consente di ridurre maggiormente il tempo di esecuzione di mysqldump.

Consigliamo anche di soppesare le prestazioni di mysqldump e i vantaggi offerti dal caricamento con il formato a testo delimitato. Un backup eseguito con testo delimitato crea un file di testo separato da tabulazioni per ciascuna tabella eliminata. Per ridurre il tempo di importazione del database, puoi caricare questi file in parallelo con il comando `LOAD DATA LOCAL INFILE`. Per ulteriori informazioni

sul formato di mysqldump più adatto per il caricamento dei dati, consulta Utilizzo di mysqldump per i backup nella documentazione di MySQL.

Prima di iniziare l'operazione di backup, devi impostare le opzioni di replica nel database MariaDB o MySQL da copiare in Amazon RDS. Le opzioni di replic includono l'attivazione del log binario e l'impostazione di un ID server univoco. L'impostazione di tali opzioni porta il server ad avviare la registrazione delle transazioni del database e lo prepara per diventare l'istanza di replica di origine in una fase successiva del processo.

> **Note**
>
> Utilizzare l'opzione `--single-transaction` con mysqldump perché esegue il dump di uno stato coerente del database. Per garantire un file di dump valido, non eseguire istruzioni DDL (Data Definition Language) durante l'esecuzione di mysqldump. È possibile pianificare una finestra di manutenzione per queste operazioni.
> Escludi gli schemi seguenti dal file dump: `sys`, `performance_schema` e `information_schema`. Per impostazione predefinita, l'utility mysqldump esclude questi schemi.
> Per migrare utenti e privilegi, prendi in considerazione l'utilizzo di uno strumento che genera il linguaggio di controllo dei dati (DCL) per ricrearli, come l'utilità. pt-show-grants

Per impostare le opzioni di autenticazione

1. Modificare il file `my.cnf` (posto in genere sotto `/etc`).

   ```
   sudo vi /etc/my.cnf
   ```

   Aggiungere le opzioni `log_bin` e `server_id` alla sezione `[mysqld]`. L'opzione `log_bin` fornisce un identificatore di nome file per i file di log binari. L'opzione `server_id` fornisce un identificatore univoco per il server in relazioni master-replica.

   L'esempio seguente mostra la sezione `[mysqld]` aggiornata di un file `my.cnf`.

   ```
   [mysqld]
   log-bin=mysql-bin
   server-id=1
   ```

   Per ulteriori informazioni, consulta la documentazione di MySQL.

2. Per la replica con un cluster database multi-AZ, imposta `ENFORCE_GTID_CONSISTENCY` e il parametro `GTID_MODE` su `ON`.

```
mysql> SET @@GLOBAL.ENFORCE_GTID_CONSISTENCY = ON;
```

```
mysql> SET @@GLOBAL.GTID_MODE = ON;
```

   Queste impostazioni non sono necessarie per la replica con un'istanza database.

3. Riavvia il servizio `mysql`.

```
sudo service mysqld restart
```

Per creare una copia di backup del database esistente

1. Creare un backup dei dati con l'utility mysqldump, specificando SQL o testo delimitato.

   Specificare `--master-data=2` per creare un file di backup che possa essere utilizzato per avviare la replica fra i server. Per ulteriori informazioni, consultare la documentazione di mysqldump.

   Per migliorare le prestazioni e garantire l'integrità dei dati, utilizzare le opzioni `--order-by-primary` e `--single-transaction` di mysqldump.

   Per non includere il database del sistema MySQL nel backup, non utilizzare l'opzione `--all-databases` con mysqldump. Per ulteriori informazioni, consultare Creating a Data Snapshot Using mysqldump nella documentazione di MySQL.

   Se necessario, utilizzare `chmod` per avere la certezza che la directory in cui viene creato il file di backup sia scrivibile.

   > **Important**
   >
   > In Windows, eseguire la finestra di comando come amministratore.

   - Per produrre un output SQL, utilizzare il comando seguente.

     PerLinux, o: macOS Unix

```
sudo mysqldump \
    --databases database_name \
    --master-data=2  \
    --single-transaction \
    --order-by-primary \
    -r backup.sql \
    -u local_user \
    -p password
```

> **Note**
>
> Specifica credenziali diverse dai prompt mostrati qui come best practice per la sicurezza.

Per Windows:

```
mysqldump ^
    --databases database_name ^
    --master-data=2  ^
    --single-transaction ^
    --order-by-primary ^
    -r backup.sql ^
    -u local_user ^
    -p password
```

> **Note**
>
> Specifica credenziali diverse dai prompt mostrati qui come best practice per la sicurezza.

- Per produrre un output in testo delimitato, utilizzare il comando seguente.

  Per LinuxmacOS, oUnix:

```
sudo mysqldump \
    --tab=target_directory \
    --fields-terminated-by ',' \
    --fields-enclosed-by '"' \
    --lines-terminated-by 0x0d0a \
    database_name \
```

```
    --master-data=2 \
    --single-transaction \
    --order-by-primary \
    -p password
```

Per Windows:

```
mysqldump ^
    --tab=target_directory ^
    --fields-terminated-by "," ^
    --fields-enclosed-by """ ^
    --lines-terminated-by 0x0d0a ^
    database_name ^
    --master-data=2 ^
    --single-transaction ^
    --order-by-primary ^
    -p password
```

> **Note**
>
> Specifica credenziali diverse dai prompt mostrati qui come best practice per la sicurezza. Eventuali procedure, trigger, funzioni o eventi devono essere creati manualmente nel database Amazon RDS. Se il database da copiare dovesse contenere questi tipi di oggetti, dovrai escluderli al momento di eseguire mysqldump. A tale scopo, includi i seguenti argomenti con il comando mysqldump: `--routines=0 --triggers=0 --events=0`.

Quando si utilizza il formato con testo delimitato, il commento `CHANGE MASTER TO` viene restituito all'esecuzione di mysqldump. Tale commento contiene il nome e la posizione del file log principale. Se l'istanza esterna è diversa da MariaDB versione 10.0.24 o successiva, annotare i valori per `MASTER_LOG_FILE` e `MASTER_LOG_POS`. Questi valori sono necessari durante l'impostazione della replica.

```
-- Position to start replication or point-in-time recovery from
--
-- CHANGE MASTER TO MASTER_LOG_FILE='mysql-bin-changelog.000031',
 MASTER_LOG_POS=107;
```

Se si utilizza il formato SQL, è possibile ottenere il nome e la posizione del file log principale nel commento `CHANGE MASTER TO` nel file di backup. Se l'istanza esterna è MariaDB, versione 10.0.24 o successiva, si può ottenere il GTID nella fase successiva.

2. Se l'istanza esterna è MariaDB, versione 10.0.24 o successiva, si utilizza la replica basata su GTID. Eseguire `SHOW MASTER STATUS` nell'istanza MariaDB esterna per ottenere il nome e la posizione del file di log binario e convertirlo in un GTID utilizzando `BINLOG_GTID_POS` nell'istanza MariaDB esterna.

```
SELECT BINLOG_GTID_POS('binary log file name', binary log file position);
```

Annotare il GTID restituito, perché sarà necessario per configurare la replica.

3. Comprimi i dati copiati per ridurre la quantità di risorse di rete necessarie per copiare i dati nell'istanza database Amazon RDS. Annota la dimensione del file di backup. Questa informazione è necessaria per determinare le dimensioni dell'istanza Amazon EC2 da creare. Al termine, comprimere il file di backup con GZIP o un'altra utility simile.

   - Per comprimere l'output SQL, utilizzare il comando seguente.

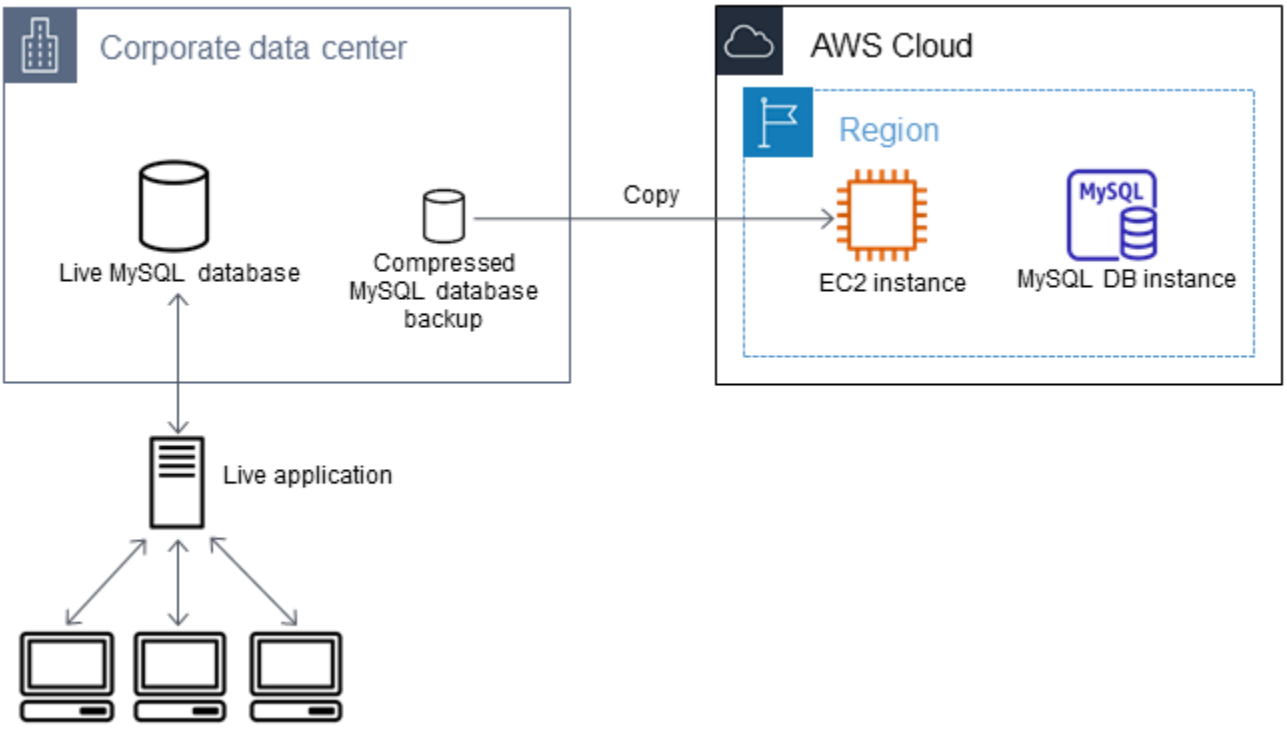
```
gzip backup.sql
```

   - Per comprimere un output in testo delimitato, utilizzare il comando seguente.

```
tar -zcvf backup.tar.gz target_directory
```

Creazione di un'istanza Amazon EC2 e copia del database compresso

La copia del file di backup del database compresso in un'istanza Amazon EC2 richiede una quantità di risorse di rete inferiore rispetto alla copia diretta di dati non compressi da un'istanza database a un'altra. Una volta che i dati sono presenti in Amazon EC2, puoi copiarli direttamente nell'istanza database MariaDB o MySQL. Per risparmiare sul costo delle risorse di rete, l'istanza Amazon EC2 deve trovarsi nella stessa AWS regione dell'istanza Amazon RDS DB. La presenza dell'istanza Amazon EC2 nella stessa AWS regione del database Amazon RDS riduce anche la latenza di rete durante l'importazione.



Per creare un'istanza Amazon EC2 e copiare i dati

1. Nel luogo in Regione AWS cui prevedi di creare il database RDS, crea un cloud privato virtuale (VPC), un gruppo di sicurezza VPC e una sottorete VPC. Verificare che le regole in entrata del gruppo di sicurezza VPC consentano agli indirizzi IP necessari per l'applicazione di connettersi ad AWS. Puoi specificare un intervallo di indirizzi IP (ad esempio `203.0.113.0/24`) oppure un altro gruppo di sicurezza VPC. È possibile utilizzare la Console di gestione Amazon VPC per creare e gestire VPC, sottoreti e gruppi di sicurezza. Per ulteriori informazioni, consultare le nozioni di base su Amazon VPC nella Guida alle operazioni di base di Amazon Virtual Private Cloud.
2. Apri la console di gestione Amazon EC2 e scegli la AWS regione in cui contenere sia l'istanza Amazon EC2 che il database Amazon RDS. Avviare un'istanza di Amazon EC2 utilizzando il VPC, la sottorete e il gruppo di sicurezza creati nella fase 1. Assicurarsi di selezionare un tipo di istanza con spazio di storage sufficiente per il file di backup del database decompresso. Per informazioni sulle istanze Amazon EC2, consulta l'argomento Nozioni di base sulle istanze Amazon EC2 Linux nella Guida per l'utente di Amazon Elastic Compute Cloud per Linux.
3. Per connetterti al database Amazon RDS dall'istanza Amazon EC2, modifica il gruppo di sicurezza VPC. Aggiungere una regola in entrata specificando l'indirizzo IP privato e l'istanza EC2. L'indirizzo IP privato è indicato nella scheda Details (Dettagli) del riquadro Instance (Istanza) della finestra della console EC2. Per modificare il gruppo di sicurezza VPC e aggiungere una regola in entrata, selezionare Security Groups (Gruppi di sicurezza) nel riquadro di navigazione della console di

EC2, selezionare il gruppo di sicurezza e aggiungere una regola in entrata per MySQL/Aurora specificando l'indirizzo IP privato dell'istanza EC2. Per ulteriori informazioni sull'aggiunta di una regola in entrata a un gruppo di sicurezza VPC, consulta Aggiunta ed eliminazione delle regole nella Guida per l'utente di Amazon VPC.

4. Copiare il file compresso con il backup del database dal sistema locale all'istanza Amazon EC2. Se necessario, utilizzare `chmod` per ottenere l'autorizzazione di scrittura per la directory di destinazione dell'istanza Amazon EC2. Il file può essere copiato con `scp` oppure con un client Secure Shell (SSH). Di seguito è riportato un esempio.

```
scp -r -i key pair.pem backup.sql.gz ec2-user@EC2 DNS:/target_directory/backup.sql.gz
```

> **Important**
>
> Assicurarsi di copiare i dati sensibili utilizzando un protocollo di trasferimento di rete sicuro.

5. Eseguire la connessione all'istanza Amazon EC2 e installare gli ultimi aggiornamenti e gli strumenti del clien MySQL mediante i seguenti comandi.

```
sudo yum update -y
sudo yum install mysql -y
```

Per ulteriori informazioni, consultare la pagina relativa alla connessione all'istanza nella Guida per l'utente di Amazon Elastic Compute Cloud per Linux.

> **Important**
>
> Questo esempio installa il client MySQL su una Amazon Machine Image (AMI) per una distribuzione Amazon Linux. Non si applica all'installazione del client MySQL su una distribuzione diversa, come Ubuntu o Red Hat Enterprise Linux. Per informazioni sull'installazione di MySQL, visita la pagina Installazione e aggiornamento di MySQL nella documentazione MySQL.

6. Durante la connessione all'istanza Amazon EC2 decomprimere il file di backup del database. Di seguito vengono mostrati gli esempi.
   - Per decomprimere l'output SQL, utilizzare il comando seguente.

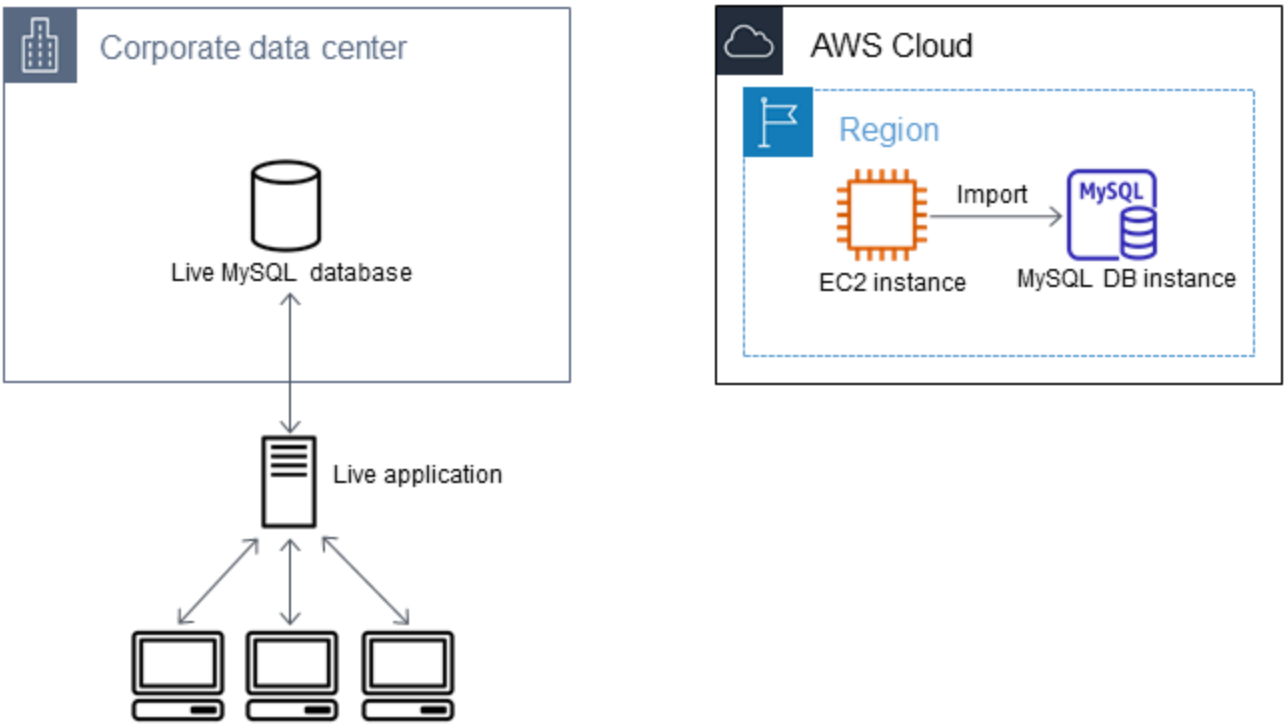
```
gzip backup.sql.gz -d
```

- Per decomprimere un output in testo delimitato, utilizzare il comando seguente.

```
tar xzvf backup.tar.gz
```

Creazione di un database MySQL o MariaDB e importazione dei dati dall'istanza Amazon EC2

Creando un'istanza DB MariaDB, un'istanza DB MySQL o un cluster DB MySQL Multi-AZ nella AWS stessa regione dell'istanza Amazon EC2, puoi importare il file di backup del database da EC2 più velocemente che su Internet.



Per creare un database MariaDB o MySQL e importare i dati

1. Determina la classe di istanza database e la quantità di spazio di archiviazione necessaria per supportare il carico di lavoro previsto per il database Amazon RDS. Come parte di questo processo, è necessario valutare la quantità di spazio richiesta e la capacità di elaborazione per le procedure di caricamento dati. Valuta anche l'occorrente per gestire il carico di lavoro della produzione. È possibile produrre una stima sulla base delle dimensioni e delle risorse del database MariaDB o MySQL di origine. Per ulteriori informazioni, consulta Classi di istanze database.

2. Crea un'istanza DB o un cluster DB Multi-AZ nella AWS regione che contiene la tua istanza Amazon EC2.

   Per creare un cluster database multi-AZ MySQL, segui le istruzioni riportate in Creazione di un cluster di database Multi-AZ.

   Per creare un'istanza database MariaDB o MySQL, segui le istruzioni riportate in Creazione di un'istanza database Amazon RDS e attieniti alle seguenti linee guida:

   - Specifica una versione del motore di database compatibile con l'istanza database di origine, come indicato di seguito:
     - Se l'istanza di origine è MySQL 5.5.x, l'istanza database Amazon RDS deve essere MySQL.
     - Se l'istanza di origine è MySQL 5.6.x o 5.7.x, l'istanza database Amazon RDS deve essere MySQL o MariaDB.
     - Se l'istanza di origine è MySQL 8.0.x, l'istanza database di Amazon RDS deve essere MySQL 8.0.x.
     - Se l'istanza di origine è MariaDB 5.5 o versione successiva, l'istanza database Amazon RDS deve essere MariaDB.
   - Specifica lo stesso cloud privato virtuale (VPC) e lo stesso gruppo di sicurezza VPC dell'istanza Amazon EC2. Questo approccio garantisce che l'istanza Amazon EC2 e l'istanza Amazon RDS siano visibili una all'altra in rete. Assicurati che l'istanza database sia accessibile pubblicamente. L'istanza database deve essere pubblicamente accessibile per impostare la replica con il database di origine descritto successivamente in questo argomento.
   - Non configurare più zone di disponibilità, retention dei backup o repliche di lettura fino a quando non è stato importato il backup del database. Al termine dell'importazione, puoi configurare le varie zone di disponibilità e la conservazione dei backup per l'istanza di produzione.
3. Esamina le opzioni di configurazione predefinite per il database Amazon RDS. Se il gruppo di parametri predefinito per il database non include le opzioni di configurazione desiderate, cercane uno che le contenga oppure crea un gruppo di parametri nuovo. Per ulteriori informazioni sulla creazione di gruppi di parametri, consulta Utilizzo di gruppi di parametri.
4. Connettiti al nuovo database Amazon RDS come utente master. Creare gli utenti necessari per supportare gli amministratori, le applicazioni e i servizi che devono accedere all'istanza. Il nome host per il database Amazon RDS corrisponde al valore dell'endpoint per l'istanza, senza includere il numero di porta. Un esempio è `mysampledb.123456789012.us-`

`west-2.rds.amazonaws.com`. Il valore dell'endpoint è disponibile nei dettagli del database nella Console di gestione Amazon RDS.

5. Eseguire la connessione all'istanza di Amazon EC2. Per ulteriori informazioni, consultare la pagina relativa alla connessione all'istanza nella Guida per l'utente di Amazon Elastic Compute Cloud per Linux.
6. Connettiti al database Amazon RDS come host remoto dall'istanza Amazon EC2 usando il comando `mysql`. Di seguito è riportato un esempio.

```
mysql -h host_name -P 3306 -u db_master_user -p
```

   Il nome host corrisponde all'endpoint del database Amazon RDS.
7. Al prompt `mysql` eseguire il comando `source` e passare al comando il nome del file dump del database per caricare i dati nell'istanza database Amazon RDS.

   - Per il formato SQL, utilizzare il comando seguente.

```
mysql> source backup.sql;
```

   - Per il formato con testo delimitato, crea innanzitutto il database, se non usi il database predefinito creato al momento dell'impostazione del database Amazon RDS.

```
mysql> create database database_name;
mysql> use database_name;
```

     Quindi creare le tabelle.

```
mysql> source table1.sql
mysql> source table2.sql
etc...
```

     Infine, importare i dati.

```
mysql> LOAD DATA LOCAL INFILE 'table1.txt' INTO TABLE table1 FIELDS TERMINATED BY
 ',' ENCLOSED BY '"' LINES TERMINATED BY '0x0d0a';
mysql> LOAD DATA LOCAL INFILE 'table2.txt' INTO TABLE table2 FIELDS TERMINATED BY
 ',' ENCLOSED BY '"' LINES TERMINATED BY '0x0d0a';
etc...
```

Per migliorare le prestazioni, puoi eseguire queste operazioni in parallelo da più connessioni, in modo che tutte le tabelle vengano create e caricate contemporaneamente.

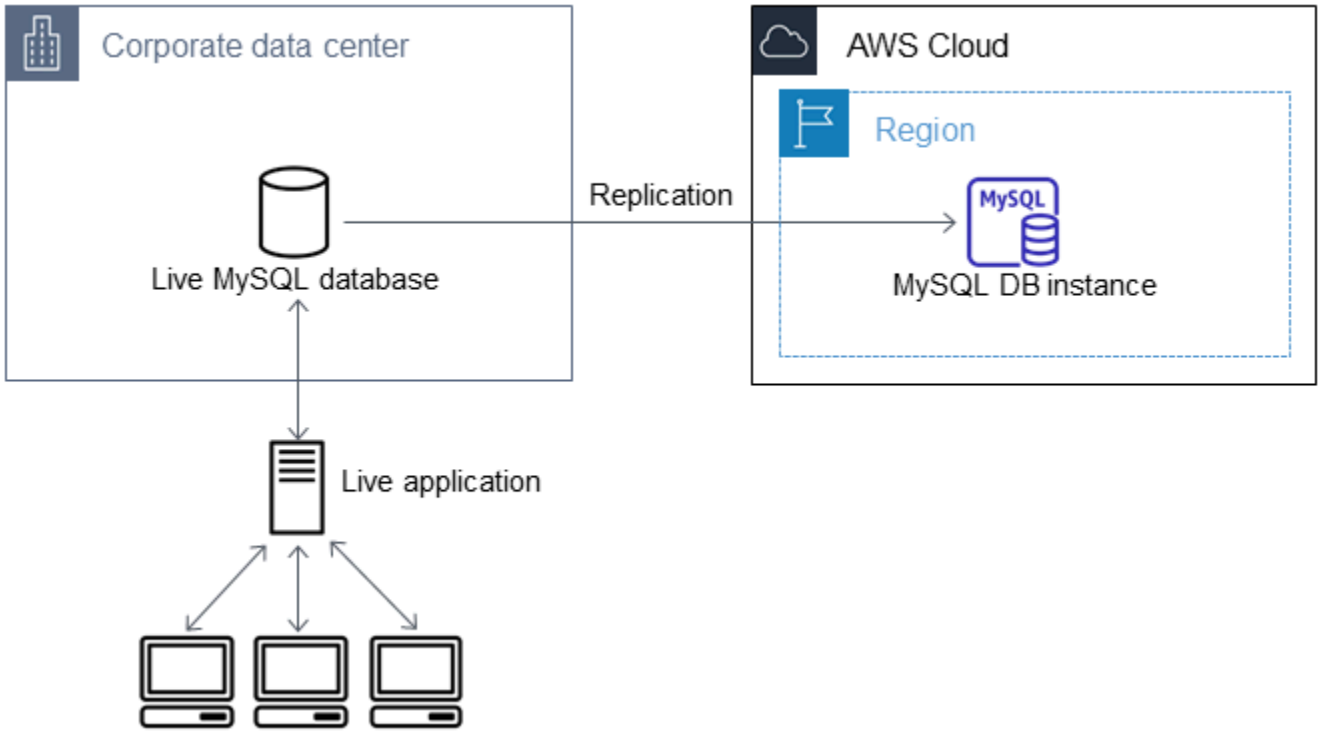
> **Note**
>
> Se hai utilizzato opzioni di formattazione dei dati con mysqldump quando hai inizialmente scaricato la tabella, assicurati di utilizzare le stesse opzioni per garantire una corretta interpretazione del contenuto del file di `LOAD DATA LOCAL INFILE` dati.

8. Eseguite una semplice `SELECT` query su una o due tabelle del database importato per verificare che l'importazione sia avvenuta correttamente.

Se non hai più bisogno dell'istanza Amazon EC2 utilizzata in questa procedura, interrompi l'istanza EC2 per ridurre l'utilizzo delle risorse. AWS Per terminare un'istanza EC2, consulta Cessazione di un'istanza nella Guida per l'utente di Amazon EC2.

Replica tra il database esterno e un nuovo database Amazon RDS

È probabile che il database di origine sia stato aggiornato durante la copia e il trasferimento dei dati nel database MariaDB o MySQL. Pertanto, puoi utilizzare la replica per portare il database copiato con il database up-to-date di origine.



Le autorizzazioni necessarie per avviare la replica in un database Amazon RDS sono limitate e non disponibili per l'utente master Amazon RDS. Per questo motivo, assicurati di usare il comando mysql.rds_set_external_master o mysql.rds_set_external_master_gtid di Amazon RDS per configurare la replica e il comando mysql.rds_start_replication per avviare la replica tra il database attivo e il database Amazon RDS.

Per avviare la replica

In precedenza, hai attivato il log binario e impostato un ID server univoco per il database di origine. Ora puoi impostare il database Amazon RDS come replica, utilizzando il database live come istanza di replica di origine.

1. Nella Console di gestione Amazon RDS aggiungi l'indirizzo IP del server che ospita il database di origine al gruppo di sicurezza VPC per il database Amazon RDS. Per ulteriori informazioni sulla modifica di un gruppo di sicurezza VPC, consulta Gruppi di sicurezza per il VPC nella Guida per l'utente di Amazon Virtual Private Cloud.

   Potrebbe essere necessario configurare anche la rete locale per consentire le connessioni dall'indirizzo IP del database Amazon RDS, in modo da poter comunicare con l'istanza di origine. Per individuare l'indirizzo IP del database Amazon RDS, utilizza il comando `host`.

   ```
   host rds_db_endpoint
   ```

   Il nome host corrisponde al nome DNS dell'endpoint del database Amazon RDS, ad esempio `myinstance.123456789012.us-east-1.rds.amazonaws.com`. Puoi trovare il valore dell'endpoint è disponibile nei dettagli dell'istanza, nella console di gestione Amazon RDS.
2. Utilizzando il client scelto, eseguire la connessione all'istanza di origine e creare un utente da utilizzare per la replica. Questo account viene utilizzato unicamente per la replica e deve essere limitato al dominio personale per aumentare la sicurezza. Di seguito è riportato un esempio.

   MySQL 5.5, 5.6 e 5.7

   ```
   CREATE USER 'repl_user'@'mydomain.com' IDENTIFIED BY 'password';
   ```

   MySQL 8.0

   ```
   CREATE USER 'repl_user'@'mydomain.com' IDENTIFIED WITH mysql_native_password BY
    'password';
   ```

> **Note**
>
> Specifica credenziali diverse dai prompt mostrati qui come best practice per la sicurezza.

3. Per l'istanza di origine, concedere i privilegi `REPLICATION CLIENT` e `REPLICATION SLAVE` all'utente di replica. Per concedere ad esempio i privilegi `REPLICATION CLIENT` e `REPLICATION SLAVE` su tutti i database per l'utente "`repl_user`" del proprio dominio, eseguire questo comando.

   MySQL 5.5, 5.6 e 5.7

   ```
   GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user'@'mydomain.com'
    IDENTIFIED BY 'password';
   ```

   MySQL 8.0

   ```
   GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user'@'mydomain.com';
   ```

   > **Note**
   >
   > Specifica credenziali diverse dai prompt mostrati qui come best practice per la sicurezza.

4. Se per creare il file di backup è stato usato il formato SQL e l'istanza esterna non è MariaDB 10.0.24 o superiore, controllare il contenuto del file.

   ```
   cat backup.sql
   ```

   Il file include un commento `CHANGE MASTER TO` che contiene il nome e la posizione del file di log principale. Il commento si trova nel file di backup, se è stata utilizzata l'opzione `--master-data` con mysqldump. Prendere nota dei valori per `MASTER_LOG_FILE` e `MASTER_LOG_POS`.

   ```
   --
   -- Position to start replication or point-in-time recovery from
   --

   -- CHANGE MASTER TO MASTER_LOG_FILE='mysql-bin-changelog.000031', MASTER_LOG_POS=107;
   ```

Se per creare il file di backup è stato usato il formato con testo delimitato e l'istanza esterna non è MariaDB 10.0.24 o superiore, si dovrebbe già disporre delle coordinate del log binario dalla fase 1 della procedura descritta nella sezione “Per creare una copia di backup del database esistente” di questo articolo.

Se l'istanza esterna è MariaDB 10.0.24 o superiore, si dovrebbe già disporre del GTID da cui avviare la replica dalla fase 2 della procedura descritta nella sezione “Per creare una copia di backup del database esistente” di questo articolo.

5. Definisci il database Amazon RDS come replica. Se l'istanza esterna non è MariaDB 10.0.24 o versioni successive, connettiti al database Amazon RDS come utente master e identifica il database di origine come istanza di replica di origine usando il comando mysql.rds_set_external_master. Se si dispone di un file di backup in formato SQL, utilizzare il nome e la posizione del file log principale, recuperati nella fase precedente. Se invece è stato usato il formato con testo delimitato, utilizzare il nome e la posizione determinati al momento di creare i file di backup. Di seguito è riportato un esempio.

```
CALL mysql.rds_set_external_master ('myserver.mydomain.com', 3306,
    'repl_user', 'password', 'mysql-bin-changelog.000031', 107, 0);
```

> **Note**
>
> Specifica credenziali diverse dai prompt mostrati qui come best practice per la sicurezza.

Se l'istanza esterna è MariaDB 10.0.24 o versioni successive, connettiti al database Amazon RDS come utente master e identifica il database di origine come istanza di replica di origine usando il comando mysql.rds_set_external_master_gtid. Utilizzare il GTID determinato nel passaggio 2 della procedura descritta nella sezione “Per creare una copia di backup del database esistente” di questo articolo. Di seguito è riportato un esempio.

```
CALL mysql.rds_set_external_master_gtid ('source_server_ip_address', 3306,
 'ReplicationUser', 'password', 'GTID', 0);
```

`source_server_ip_address` è l' indirizzo IP dell'istanza di replica di origine. Al momento, gli indirizzi DNS privati di EC2 non sono supportati.

> **Note**
>
> Specifica credenziali diverse dai prompt mostrati qui come best practice per la sicurezza.

6. Nel database Amazon RDS esegui il comando mysql.rds_start_replication per avviare la replica.

```
CALL mysql.rds_start_replication;
```

7. Sul database Amazon RDS, esegui il comando SHOW REPLICA STATUS per determinare quando la replica è up-to-date con l'istanza di replica di origine. I risultati del comando `SHOW REPLICA STATUS` includono il campo `Seconds_Behind_Master`. Quando il `Seconds_Behind_Master` campo restituisce 0, la replica si trova up-to-date con l'istanza di replica di origine.

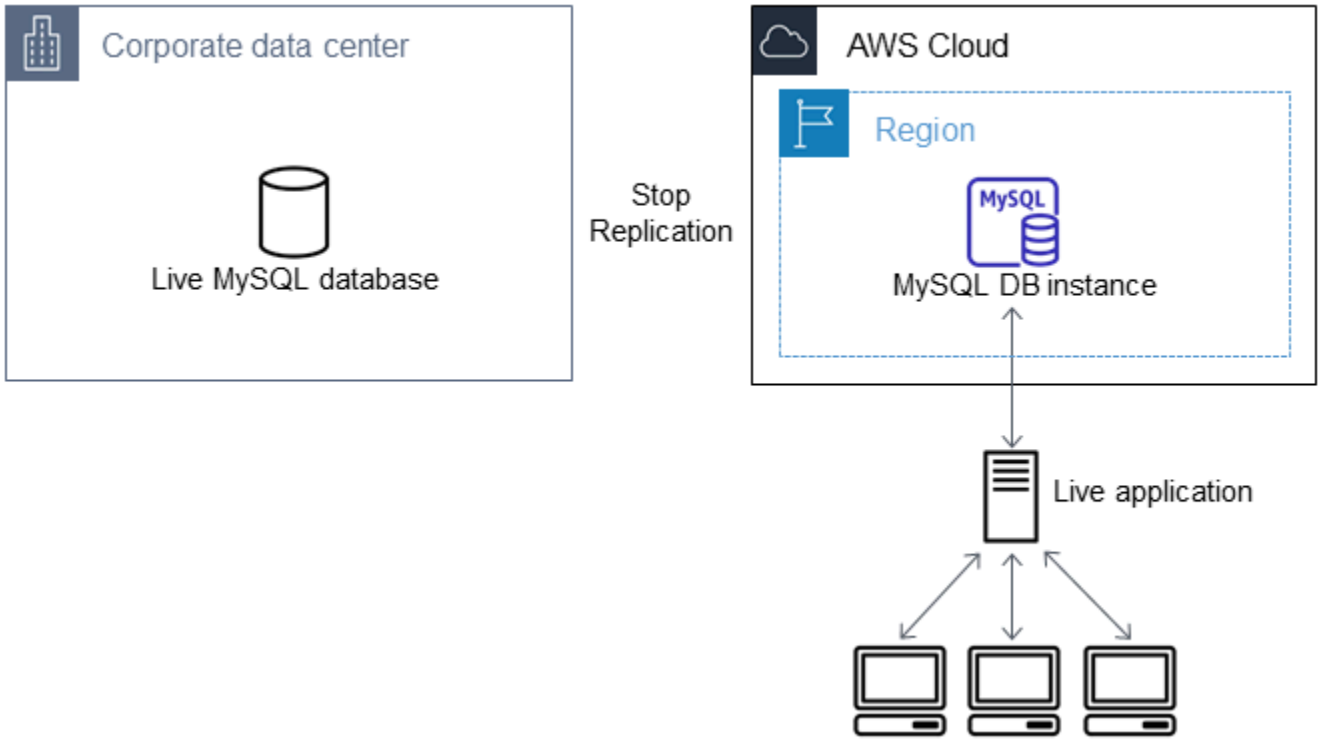
   > **Note**
   >
   > Versioni precedenti di MySQL utilizzate `SHOW SLAVE STATUS` al posto di `SHOW REPLICA STATUS`. Se si utilizza una versione MySQL prima della 8.0.23, utilizzare `SHOW SLAVE STATUS`.

   Per un'istanza database MariaDB 10.5, 10.6 o 10.11, esegui la procedura mysql.rds_replica_status anziché il comando MySQL.

8. Una volta installato il database Amazon RDS up-to-date, attiva i backup automatici in modo da poter ripristinare il database, se necessario. È possibile attivare o modificare i backup automatici per il database Amazon RDS tramite la Console di gestione Amazon RDS. Per ulteriori informazioni, consulta Introduzione ai backup.

Reindirizzamento di un'applicazione attiva nell'istanza di Amazon RDS

Dopo che il up-to-date database MariaDB o MySQL è con l'istanza di replica di origine, ora puoi aggiornare la tua applicazione live per utilizzare l'istanza Amazon RDS.



Per reindirizzare l'applicazione live al database MariaDB o MySQL e arrestare la replica

1. Per aggiungere il gruppo di sicurezza VPC per il database Amazon RDS, immetti l'indirizzo IP del server che ospita l'applicazione. Per ulteriori informazioni sulla modifica di un gruppo di sicurezza VPC, consulta Gruppi di sicurezza per il VPC nella Guida per l'utente di Amazon Virtual Private Cloud.
2. Verifica che il `Seconds_Behind_Master` campo nei risultati del comando SHOW REPLICA STATUS sia 0, il che indica che la replica è con l'istanza di replica di origine. up-to-date

```
SHOW REPLICA STATUS;
```

> **Note**
>
> Versioni precedenti di MySQL utilizzate `SHOW SLAVE STATUS` al posto di `SHOW REPLICA STATUS`. Se si utilizza una versione MySQL prima della 8.0.23, utilizzare `SHOW SLAVE STATUS`.

Per un'istanza database MariaDB 10.5, 10.6 o 10.11, esegui la procedura mysql.rds_replica_status anziché il comando MySQL.

3. Chiudere tutte le connessioni all'origine quando le loro transazioni sono complete.
4. Aggiorna l'applicazione per usare il database Amazon RDS. In genere, l'aggiornamento prevede la modifica delle impostazioni di connessione per identificare il nome host e la porta del database Amazon RDS, l'account utente e la password per eseguire la connessione e il database da utilizzare.
5. Effettua la connessione all'istanza database.

   Per un cluster database multi-AZ, connettiti all'istanza database di scrittura.
6. Interrompere la replica per l'istanza Amazon RDS tramite il comando mysql.rds_stop_replication.

```
CALL mysql.rds_stop_replication;
```

7. Esegui il comando mysql.rds_reset_external_master nel database Amazon RDS per reimpostare la configurazione della replica in modo che l'istanza non venga più identificata come replica.

```
CALL mysql.rds_reset_external_master;
```

8. Attivare le caratteristiche aggiuntive di Amazon RDS, quali il supporto Multi-AZ e le repliche di lettura. Per ulteriori informazioni, consultare Configurazione e gestione di un'implementazione multi-AZ e Uso delle repliche di lettura dell'istanza database.

Importazione dei dati da qualsiasi origine a un'istanza database MariaDB o MySQL

Consigliamo di creare snapshot DB dell'istanza database Amazon RDS di destinazione prima e dopo il caricamento dei dati. Le snapshot DB di Amazon RDS sono backup completi della tua istanza database che puoi utilizzare per ripristinare la tua istanza database in uno stato noto. Quando avvii una snapshot DB, le operazioni I/O dell'istanza database vengono temporaneamente sospese per il backup.

Creando una snapshot DB immediatamente prima di caricare i dati ti consente di ripristinare il database allo stato precedente il caricamento, se fosse necessario. Una snapshot DB effettuata immediatamente dopo il caricamento evita la necessità di caricare nuovamente i dati in caso di problemi e può essere utilizzata per inizializzare nuove istanze database.

Nell'elenco seguente è indicata la procedura da eseguire. Ciascun passaggio della procedura è descritto in modo dettagliato di seguito.

1. Creazione di file flat contenenti i dati da caricare.
2. Arresto delle applicazioni che accedono all'istanza database di destinazione.
3. Creazione di una snapshot DB.
4. Valuta se disattivare i backup automatici di Amazon RDS.
5. Carica i dati.
6. Riattivazione dei backup automatici.

Fase 1: Creazione di file flat contenenti i dati da caricare

Per salvare i dati da caricare, utilizza un formato comune, come ad esempio valori separati da virgola (CSV). Ciascuna tabella deve possedere il proprio file. Non è possibile combinare i dati di più tabelle nello stesso file. Devi fornire a ciascun file lo stesso nome della tabella corrispondente. Il file può avere qualsiasi estensione. Ad esempio, se il nome della tabella è `sales`, il nome del file potrebbe essere `sales.csv` o `sales.txt`, ma non `sales_01.csv`.

Quando possibile, ordina i dati in base alla chiave primaria della tabella da caricare. In questo modo i tempi di caricamento risultano significativamente più rapidi e si riduce il consumo di spazio su disco.

La velocità e l'efficienza di questa procedura dipende dalla capacità di mantenere contenute le dimensioni dei file. Se le dimensioni di un qualsiasi file (non compresso) superano 1 GiB, suddividilo in più file da caricare separatamente.

Nei sistemi di tipo Unix (incluso Linux), puoi utilizzare il comando `split`. Ad esempio, il comando seguente divide il file `sales.csv` in vari file con dimensioni inferiori a 1 GiB. Le divisioni vengono effettuate solo sulle interruzioni di riga (-C 1024m). I nuovi file sono denominati `sales.part_00`, `sales.part_01` e così via.

```
split -C 1024m -d sales.csv sales.part_
```

Utility simili sono disponibili anche per altri sistemi operativi.

Fase 2: Arresto delle applicazioni che accedono all'istanza database di destinazione

Prima di avviare il caricamento di grandi quantità di dati, interrompie le attività di tutte le applicazioni che accedono all'istanza database in cui intendi eseguire il caricamento. Questa operazione è particolarmente consigliata se le altre sessioni modificano le tabelle caricate o quelle di riferimento. In questo modo, puoi ridurre i rischi di violazione dei vincoli e ottimizzare le prestazioni durante

il caricamento. Inoltre, diventa possibile ripristinare l'istanza database al punto immediatamente precedente il caricamento, senza perdere le modifiche apportate dai processi che non sono coinvolti nell'operazione di caricamento.

Ovviamente, ci sono casi in cui l'esecuzione di questa operazione risulta impossibile o poco pratica. Se puoi evitare che alcune applicazioni accedano all'istanza database prima del caricamento, prendi tutte le misure necessarie per assicurare la disponibilità e l'integrità dei dati. Tali misure dipendono in larga parte dal tipo specifico di utilizzo e dai requisiti del sito.

Fase 3: Creazione di una snapshot DB

Se desideri caricare i dati in una nuova istanza database priva di dati, puoi ignorare questa parte. In caso contrario, la creazione di uno snapshot DB dell'istanza database ti consente di ripristinare l'istanza database nel punto immediatamente precedente al caricamento, se fosse necessario. Come spiegato in precedenza, quando avvii uno snapshot DB, le operazioni I/O dell'istanza database vengono sospese per alcuni minuti, mentre ha luogo il backup.

L'esempio seguente utilizza il AWS CLI `create-db-snapshot` comando per creare uno snapshot DB dell'`AcmeRDS`istanza e assegnare all'istantanea del DB l'identificatore. `"preload"`

PerLinux, o: macOS Unix

```
aws rds create-db-snapshot \
    --db-instance-identifier AcmeRDS \
    --db-snapshot-identifier preload
```

Per Windows:

```
aws rds create-db-snapshot ^
    --db-instance-identifier AcmeRDS ^
    --db-snapshot-identifier preload
```

Puoi utilizzare anche la funzione di ripristino da snapshot DB per creare istanze database di prova in cui eseguire test o annullare modifiche apportate durante il caricamento.

Ricorda che il ripristino di un database da una snapshot DB crea una nuova istanza database che, come tutte le istanze database, possiede un identificatore e un endpoint univoci. Per ripristinare l'istanza database senza modificare l'endpoint, devi innanzitutto eliminare l'istanza database, in modo da poter riutilizzare l'endpoint.

Ad esempio, per creare un'istanza database in cui eseguire test di vario tipo, devi assegnare all'istanza database il proprio identificatore. Nell'esempio, l'identificatore è `AcmeRDS-2`". L'esempio si connette all'istanza database utilizzando l'endpoint associato a `AcmeRDS-2`.

Per LinuxmacOS, oUnix:

```
aws rds restore-db-instance-from-db-snapshot \
    --db-instance-identifier AcmeRDS-2 \
    --db-snapshot-identifier preload
```

Per Windows:

```
aws rds restore-db-instance-from-db-snapshot ^
    --db-instance-identifier AcmeRDS-2 ^
    --db-snapshot-identifier preload
```

Per riutilizzare l'endpoint esistente, innanzitutto elimina l'istanza database e fornisci al database ripristinato lo stesso identificatore.

Per LinuxmacOS, oUnix:

```
aws rds delete-db-instance \
    --db-instance-identifier AcmeRDS \
    --final-db-snapshot-identifier AcmeRDS-Final

aws rds restore-db-instance-from-db-snapshot \
    --db-instance-identifier AcmeRDS \
    --db-snapshot-identifier preload
```

Per Windows:

```
aws rds delete-db-instance ^
    --db-instance-identifier AcmeRDS ^
    --final-db-snapshot-identifier AcmeRDS-Final

aws rds restore-db-instance-from-db-snapshot ^
    --db-instance-identifier AcmeRDS ^
    --db-snapshot-identifier preload
```

L'esempio precedente crea uno snapshot DB finale dell'istanza database prima di eliminarla. Questo passaggio è facoltativo, ma è consigliato.

Fase 4: Eventuale disattivazione dei backup automatici di Amazon RDS

> **Warning**
>
> Non disattivare i backup automatici se è necessario eseguire il point-in-time ripristino.

La disattivazione dei backup automatici cancella tutti i backup esistenti, quindi il point-in-time ripristino non è possibile dopo la disattivazione dei backup automatici. La disattivazione dei backup automatici serve a ottimizzare le prestazioni, ma non è indispensabile per il caricamento dei dati. Gli snapshot DB manuali non sono influenzati dalla disattivazione dei backup automatici. Tutti gli snapshot DB esistenti rimangono disponibili per il ripristino.

La disattivazione dei backup automatici velocizza il tempo di caricamento di circa il 25% e riduce la quantità di spazio richiesto. Se devi caricare dati in una nuova istanza database che non contiene altri dati, la disattivazione dei backup rappresenta un'ottima soluzione per velocizzare il caricamento ed evitare di occupare troppo spazio con i backup. Tuttavia, in alcuni casi, potresti pianificare di caricare i dati in un'istanza database che contiene già altri dati. In tal caso, soppesate i vantaggi della disattivazione dei backup rispetto all'impatto della perdita della capacità di esecuzione. point-in-time-recovery

Per impostazione predefinita, i backup sono attivati per le istanze database (con un periodo di conservazione di un giorno). Per disabilitare i backup automatici, imposta il periodo di conservazione del backup su zero. Dopo il caricamento potrai riattivare i backup automatici impostando il periodo di conservazione su un valore diverso da zero. Per attivare o disattivare i backup, Amazon RDS chiude l'istanza database e la riavvia in modo da attivare o disattivare i log MariaDB o MySQL.

Utilizzate il AWS CLI `modify-db-instance` comando per impostare la conservazione dei backup su zero e applicare immediatamente la modifica. Per impostare il periodo di retention su zero è necessario riavviare l'istanza database, quindi prima di continuare dovrai attendere il completamento del riavvio.

Per LinuxmacOS, oUnix:

```
aws rds modify-db-instance \
    --db-instance-identifier AcmeRDS \
    --apply-immediately \
    --backup-retention-period 0
```

Per Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier AcmeRDS ^
    --apply-immediately ^
    --backup-retention-period 0
```

Puoi controllare lo stato della tua istanza DB con il AWS CLI `describe-db-instances` comando. Nell'esempio seguente viene visualizzato lo stato dell'istanza database dell'istanza database `AcmeRDS`.

```
aws rds describe-db-instances --db-instance-identifier AcmeRDS --query "*[].
{DBInstanceStatus:DBInstanceStatus}"
```

Quando lo stato dell'istanza DB è `available`, si è pronti per procedere.

Fase 5: Caricamento dei dati

Usa l'istruzione `LOAD DATA LOCAL INFILE` MySQL per leggere le righe dai tuoi file flat nelle tabelle del database.

L'esempio seguente mostra come caricare i dati da un file denominato `sales.txt` in una tabella denominata `Sales` nel database.

```
mysql> LOAD DATA LOCAL INFILE 'sales.txt' INTO TABLE Sales FIELDS TERMINATED BY ' '
 ENCLOSED BY '' ESCAPED BY '\\';
Query OK, 1 row affected (0.01 sec)
Records: 1  Deleted: 0  Skipped: 0  Warnings: 0
```

Per ulteriori informazioni sulla `LOAD DATA` dichiarazione, consulta la documentazione di MySQL.

Fase 6: Riattivazione dei backup automatici di Amazon RDS

Al termine del caricamento, riattiva i backup automatici di Amazon RDS reimpostando il tempo di conservazione del backup sul valore originale. Come indicato in precedenza, Amazon RDS riavvia l'istanza database, interrompendo brevemente le attività.

L'esempio seguente utilizza il AWS CLI `modify-db-instance` comando per attivare i backup automatici per l'istanza `AcmeRDS` DB e impostare il periodo di conservazione su un giorno.

Per LinuxmacOS, oUnix:

```
aws rds modify-db-instance \
    --db-instance-identifier AcmeRDS \
    --backup-retention-period 1 \
    --apply-immediately
```

Per Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier AcmeRDS ^
    --backup-retention-period 1 ^
    --apply-immediately
```

Uso della replica MySQL in Amazon RDS

Generalmente, per configurare la replica tra le istanze database di Amazon RDS si utilizzano repliche di lettura. Per informazioni generali sulle repliche di lettura, consulta Uso delle repliche di lettura dell'istanza database. Per informazioni specifiche sull'uso di repliche di lettura in Amazon RDS per MySQL, consulta Uso delle repliche di lettura MySQL.

È possibile utilizzare gli ID globali di transazione (GTID) per la replica con RDS per MySQL. Per ulteriori informazioni, consulta Utilizzo della replica basata su GTID.

Puoi anche configurare la replica tra un'istanza database RDS for MySQL e un'istanza MariaDB o MySQL esterna ad Amazon RDS. Per ulteriori informazioni sulla configurazione della replica con un'origine esterna, consulta Configurazione della replica della posizione del file di log binario con un'istanza di origine esterna..

Per qualsiasi opzione di replica, puoi utilizzare la replica basata su riga, basata su istruzioni o quella mista. La replica basata su riga replica solamente le righe modificate che risultano da un'istruzione SQL. La replica basata su istruzioni replica l'intera istruzione SQL. La replica mista utilizza la replica basata su istruzione quando possibile, ma passa alla replica basata su riga quando vengono eseguite le istruzioni SQL che non sono sicure per la replica basata su istruzione. Nella maggior parte dei casi, si consiglia l'utilizzo della replica mista. Il formato di log binario dell'istanza database determina se la replica è basata su riga, su istruzione o è mista. Per informazioni sull'impostazione del formato di log binario, consulta Configurazione del log binario di MySQL.

> **Note**
>
> Puoi configurare la replica per l'importazione di database da un'istanza MariaDB o MySQL esterna ad Amazon RDS o per l'esportazione di database a tali istanze. Per ulteriori informazioni, consultare Importazione dei dati in un database Amazon RDS MariaDB o MySQL con tempi di inattività ridotti e Esportazione di dati da un'istanza database MySQL tramite la replica.

Argomenti

Uso delle repliche di lettura MySQL

Questa sezione contiene informazioni specifiche sull'utilizzo delle repliche di lettura su RDS per MySQL. Per informazioni generali sulle repliche di lettura e istruzioni su come usarle, consulta Uso delle repliche di lettura dell'istanza database.

Argomenti

Configurazione delle repliche di lettura con MySQL

Prima di poter utilizzare un'istanza database MySQL come un'origine delle replica, assicurati di abilitare i backup automatici sull'istanza database di origine. A questo scopo, imposta il periodo di retention dei backup su un valore diverso da zero. Questo requisito si applica anche a una replica di lettura che rappresenta l'istanza database di origine per un'altra replica di lettura. I backup automatici vengono solo supportati per le repliche di lettura che eseguono qualsiasi versione di MySQL. Puoi configurare la replica in base alle coordinate del log binario per un'istanza database MySQL.

Su RDS for MySQL versione 5.7.44 e versioni successive di MySQL 5.7 e RDS for MySQL 8.0.28 e versioni successive 8.0, è possibile configurare la replica utilizzando identificatori di transazione globali (GTID). Per ulteriori informazioni, consulta Utilizzo della replica basata su GTID.

È possibile creare fino a 15 repliche di lettura da un'istanza database nella stessa regione. Per un efficace funzionamento della replica, ciascuna replica di lettura dovrebbe avere la stessa quantità di risorse di calcolo e storage dell'istanza database di origine. Se si dimensiona l'istanza database di origine, si devono dimensionare anche le repliche di lettura.

RDS per MySQL supporta le repliche di lettura a cascata. Per informazioni su come configurare le repliche di lettura a cascata, consulta Utilizzo di repliche di lettura a cascata con RDS per MySQL.

Puoi eseguire più operazioni di creazione ed eliminazione di repliche di lettura simultanee che fanno riferimento alla stessa istanza database di origine. Quando esegui queste operazioni, rimani entro il limite delle 15 repliche di lettura per ogni istanza di origine.

Una replica in lettura di un'istanza DB MySQL non può utilizzare una versione del motore DB inferiore rispetto alla sua istanza DB di origine.

Preparazione delle istanze database MySQL che utilizzano MyISAM

Se l'istanza database MySQL utilizza un motore non transazionale come MyISAM, devi eseguire la seguente procedura per configurare correttamente la replica di lettura. Questa procedura è necessaria per verificare che la replica di lettura contenga una copia coerente dei dati. Non è invece necessaria alcuna procedura se tutte le tabelle usano un motore transazionale come InnoDB.

1. Arresta tutte le operazioni DML (Data Manipulation Language) e DDL (Data Definition Language) sulle tabelle non transazionali nell'istanza database di origine e attendi il loro completamento. Le istruzioni SELECT possono restare in esecuzione.
2. Scarica e blocca le tabelle nell'istanza database di origine.
3. Crea una replica di lettura usando uno dei metodi nelle seguenti sezioni.
4. Verifica lo stato di avanzamento della creazione della replica di lettura utilizzando, ad esempio, l'operazione API `DescribeDBInstances`. Dopo che la replica di lettura è disponibile, sblocca le tabelle dell'istanza database di origine e ripristina le normali operazioni del database.

Configurazione dei filtri di replica con MySQL

Puoi utilizzare i filtri di replica per specificare quali database e tabelle vengono replicati con una replica di lettura. I filtri di replica possono includere database e tabelle nella replica o escluderli dalla replica.

Di seguito sono riportati alcuni casi d'uso per i filtri di replica:

- Per ridurre le dimensioni di una replica di lettura. Con il filtro di replica è possibile escludere i database e le tabelle che non sono necessari nella replica di lettura.
- Per escludere database e tabelle dalle repliche di lettura per motivi di sicurezza.
- Per replicare database e tabelle diversi per casi d'uso specifici in repliche di lettura diverse. Ad esempio, è possibile utilizzare repliche di lettura specifiche per l'analisi o la condivisione.
- Per un'istanza DB che ha letto repliche in diversi, per replicare database o tabelle diversi in diversi. Regioni AWS Regioni AWS

> **Note**
>
> Puoi utilizzare i filtri di replica anche per specificare i database e le tabelle che vengono replicati con un'istanza database MySQL primaria configurata come replica in una topologia di replica in ingresso. Per ulteriori informazioni su questa configurazione, consulta Configurazione della replica della posizione del file di log binario con un'istanza di origine esterna..

Argomenti

Impostazione dei parametri di filtro della replica RDS for MySQL

Per configurare i filtri di replica, impostare i seguenti parametri di filtro replica sulla replica di lettura:

- `replicate-do-db` – Replicare le modifiche ai database specificati. Quando si imposta questo parametro per una replica di lettura, vengono replicati solo i database specificati nel parametro.
- `replicate-ignore-db` – Non replicare le modifiche ai database specificati. Quando il parametro `replicate-do-db` è impostato per una replica di lettura, questo parametro non viene valutato.
- `replicate-do-table` – Replicare le modifiche alle tabelle specificate. Quando si imposta questo parametro per una replica di lettura, vengono replicate solo le tabelle specificate nel parametro. Inoltre, quando viene impostato il parametro `replicate-do-db` o `replicate-ignore-db`, assicurarsi di includere il database che include le tabelle specificate nella replica con la replica di lettura.
- `replicate-ignore-table` – Non replicare le modifiche alle tabelle specificate. Quando il parametro `replicate-do-table` è impostato per una replica di lettura, questo parametro non viene valutato.
- `replicate-wild-do-table` – Replicare le tabelle in base ai modelli di nome del database e della tabella specificati. I caratteri jolly `%` e `_` sono supportati. Quando è impostato il parametro `replicate-do-db` o `replicate-ignore-db`, assicurarsi di includere il database che include le tabelle specificate nella replica con la replica di lettura.

- `replicate-wild-ignore-table` – Non replicare le tabelle in base ai modelli di nomi di database e tabella specificati. I caratteri jolly `%` e `_` sono supportati. Quando è impostato il parametro `replicate-do-table` o `replicate-wild-do-table` per una replica di lettura, questo parametro non viene valutato.

I parametri vengono valutati nell'ordine in cui sono elencati. Per ulteriori informazioni sul funzionamento di questi parametri, consulta la documentazione di MySQL:

- Per informazioni generali, consulta Opzioni e variabili del server di replica.
- Per informazioni sulla modalità di valutazione dei parametri di filtro della replica del database, consulta Valutazione delle opzioni di replica a livello di database e registrazione binaria.
- Per informazioni sulla modalità di valutazione dei parametri di filtro replica delle tabelle, consulta Valutazione delle opzioni di replica a livello di tabella.

Per impostazione predefinita, ognuno di questi parametri ha un valore vuoto. In ogni replica di lettura, è possibile utilizzare questi parametri per impostare, modificare ed eliminare i filtri di replica. Quando viene impostato uno di questi parametri, è necessario separare ogni filtro dagli altri con una virgola.

È possibile utilizzare i caratteri jolly `%` e `_` nei parametri `replicate-wild-do-table` e `replicate-wild-ignore-table`. Il carattere jolly `%` corrisponde a un numero qualsiasi di caratteri e il carattere jolly `_` corrisponde a un solo carattere.

Il formato di registrazione binaria dell'istanza database di origine è importante per la replica perché determina il record delle modifiche ai dati. L'impostazione del parametro `binlog_format` determina se la replica è basata su righe o basata su dichiarazione. Per ulteriori informazioni, consulta Configurazione del log binario di MySQL.

> **Note**
>
> Tutte le istruzioni DDL (Data Definition Language) vengono replicate come istruzioni, indipendentemente dall'impostazione `binlog_format` dell'istanza database di origine.

Limitazioni di filtro delle repliche per RDS per MySQL

Le seguenti limitazioni si applicano al filtro di replica per RDS per MySQL:

- Ogni parametro di filtro della replica ha un limite di 2.000 caratteri.

- Le virgole non sono supportate nei filtri di replica per i valori dei parametri. In un elenco di parametri, le virgole possono essere utilizzate solo come separatori di valori. Ad esempio, `ParameterValue='`a,b`'` non è supportato, ma `ParameterValue='a,b'` lo è.
- Le opzioni MySQL `--binlog-do-db` e `--binlog-ignore-db` per il filtro dei log binari non sono supportate.
- Il filtro delle repliche non supporta le transazioni XA.

  Per ulteriori informazioni, consulta Restrizioni sulle transazioni XA nella documentazione di MySQL.

Esempi di filtri di replica per RDS per MySQL

Per configurare il filtro di replica per una replica di lettura, modificare i parametri di filtro replica nel gruppo di parametri associato alla replica di lettura.

> **Note**
>
> Non è consentito modificare un gruppo di parametri predefinito. Se la replica di lettura usa un gruppo di parametri predefinito, creare un nuovo gruppo di parametri e associarlo alla replica di lettura. Per ulteriori informazioni sui gruppi di parametri database, consulta Utilizzo di gruppi di parametri.

È possibile impostare i parametri in un gruppo di parametri utilizzando l'API AWS Management Console AWS CLI, o RDS. Per informazioni sull'estensione dei parametri consulta Modifica di parametri in un gruppo di parametri del database. Quando si impostano parametri in un gruppo di parametri, tutte le istanze DB associate al gruppo di parametri utilizzano le impostazioni dei parametri. Se si impostano i parametri di filtro della replica in un gruppo di parametri, assicurarsi che il gruppo di parametri sia associato solo alle repliche di lettura. Lasciare vuoti i parametri di filtro di replica per le istanze database di origine.

Negli esempi seguenti vengono impostati i parametri utilizzando AWS CLI. In questi esempi si imposta `ApplyMethod` su `immediate` in modo che le modifiche ai parametri avvengano immediatamente dopo il completamento del comando della CLI. Se si desidera applicare una modifica in sospeso dopo il riavvio della replica di lettura, impostare `ApplyMethod` su `pending-reboot`.

Gli esempi seguenti impostano i filtri di replica:

- Including databases in replication
- Including tables in replication
- Including tables in replication with wildcard characters
- Excluding databases from replication
- Excluding tables from replication
- Excluding tables from replication using wildcard characters

Example Inclusione dei database nella replica

Nell'esempio seguente sono inclusi i database `mydb1` e `mydb2` nella replica.

Per LinuxmacOS, oUnix:

```
aws rds modify-db-parameter-group \
  --db-parameter-group-name myparametergroup \
  --parameters "ParameterName=replicate-do-
db,ParameterValue='mydb1,mydb2',ApplyMethod=immediate"
```

Per Windows:

```
aws rds modify-db-parameter-group ^
  --db-parameter-group-name myparametergroup ^
  --parameters "ParameterName=replicate-do-
db,ParameterValue='mydb1,mydb2',ApplyMethod=immediate"
```

Example Inclusione delle tabelle nella replica

Nell'esempio seguente sono incluse le tabelle `table1` e `table2` nel database `mydb1` nella replica.

Per LinuxmacOS, oUnix:

```
aws rds modify-db-parameter-group \
  --db-parameter-group-name myparametergroup \
  --parameters "ParameterName=replicate-do-
table,ParameterValue='mydb1.table1,mydb1.table2',ApplyMethod=immediate"
```

Per Windows:

```
aws rds modify-db-parameter-group ^
  --db-parameter-group-name myparametergroup ^
  --parameters "ParameterName=replicate-do-
table,ParameterValue='mydb1.table1,mydb1.table2',ApplyMethod=immediate"
```

Example Inclusione di tabelle nella replica utilizzando caratteri jolly

Nell'esempio seguente sono incluse tabelle con nomi che iniziano con `order` e `return` nel database `mydb` nella replica.

Per LinuxmacOS, oUnix:

```
aws rds modify-db-parameter-group \
  --db-parameter-group-name myparametergroup \
  --parameters "ParameterName=replicate-wild-do-table,ParameterValue='mydb.order
%,mydb.return%',ApplyMethod=immediate"
```

Per Windows:

```
aws rds modify-db-parameter-group ^
  --db-parameter-group-name myparametergroup ^
  --parameters "ParameterName=replicate-wild-do-table,ParameterValue='mydb.order
%,mydb.return%',ApplyMethod=immediate"
```

Example Esclusione di database dalla replica

Nell'esempio seguente vengono esclusi i database `mydb5` e `mydb6` dalla replica.

Per LinuxmacOS, oUnix:

```
aws rds modify-db-parameter-group \
  --db-parameter-group-name myparametergroup \
  --parameters "ParameterName=replicate-ignore-
db,ParameterValue='mydb5,mydb6',ApplyMethod=immediate"
```

Per Windows:

```
aws rds modify-db-parameter-group ^
```

```
  --db-parameter-group-name myparametergroup ^
  --parameters "ParameterName=replicate-ignore-
db,ParameterValue='mydb5,mydb6',ApplyMethod=immediate"
```

Example Esclusione di tabelle dalla replica

Nell'esempio seguente vengono escluse dalla replica le tabelle `table1` nel database `mydb5` e `table2` nel database `mydb6`.

Per LinuxmacOS, oUnix:

```
aws rds modify-db-parameter-group \
  --db-parameter-group-name myparametergroup \
  --parameters "ParameterName=replicate-ignore-
table,ParameterValue='mydb5.table1,mydb6.table2',ApplyMethod=immediate"
```

Per Windows:

```
aws rds modify-db-parameter-group ^
  --db-parameter-group-name myparametergroup ^
  --parameters "ParameterName=replicate-ignore-
table,ParameterValue='mydb5.table1,mydb6.table2',ApplyMethod=immediate"
```

Example Esclusione di tabelle dalla replica utilizzando caratteri jolly

Nell'esempio seguente vengono escluse le tabelle con nomi che iniziano con `order` e `return` nel database `mydb7` dalla replica.

Per LinuxmacOS, oUnix:

```
aws rds modify-db-parameter-group \
  --db-parameter-group-name myparametergroup \
  --parameters "ParameterName=replicate-wild-ignore-table,ParameterValue='mydb7.order
%,mydb7.return%',ApplyMethod=immediate"
```

Per Windows:

```
aws rds modify-db-parameter-group ^
```

```
  --db-parameter-group-name myparametergroup ^
  --parameters "ParameterName=replicate-wild-ignore-table,ParameterValue='mydb7.order
%,mydb7.return%',ApplyMethod=immediate"
```

Visualizzazione dei filtri di replica per una replica di lettura

È possibile visualizzare i filtri di replica per una replica di lettura nei seguenti modi:

- Controllare le impostazioni dei parametri di filtro replica nel gruppo di parametri associato alla replica di lettura.

  Per istruzioni, consulta Visualizzazione dei valori dei parametri per un gruppo di parametri del database.
- In un client MySQL, connettersi alla replica di lettura ed eseguire l'istruzione `SHOW REPLICA STATUS`.

  Nell'output, i campi seguenti mostrano i filtri di replica per la replica di lettura:
  - `Replicate_Do_DB`
  - `Replicate_Ignore_DB`
  - `Replicate_Do_Table`
  - `Replicate_Ignore_Table`
  - `Replicate_Wild_Do_Table`
  - `Replicate_Wild_Ignore_Table`

  Per ulteriori informazioni su questi campi, consulta Verifica dello stato della replica nella documentazione di MariaDB.

  > **Note**
  >
  > Versioni precedenti di MySQL utilizzate `SHOW SLAVE STATUS` al posto di `SHOW REPLICA STATUS`. Se si utilizza una versione MySQL prima della 8.0.23, utilizzare `SHOW SLAVE STATUS`.

Configurazione della replica ritardata con MySQL

Puoi usare la replica ritardata come strategia per il disaster recovery. Con la replica ritardata puoi specificare il tempo minimo, in secondi, di ritardo della replica rispetto all'origine nella replica di

lettura. In caso di emergenza, come ad esempio l'eliminazione accidentale di una tabella, completa la seguente procedura per risolvere velocemente il problema:

- Arresta la replica sulla replica di lettura prima che la modifica che ha provocato il problema venga inviata.

  Usa la procedura archiviata mysql.rds_stop_replication per arrestare la replica.
- Avvia la replica e specifica che la replica si arresta automaticamente in corrispondenza di una posizione del file di log.

  Puoi specificare una posizione prima dell'emergenza utilizzando la procedura archiviata mysql.rds_start_replication_until.
- Utilizza le istruzioni contenute in Promozione di una replica di lettura a istanza database standalone per promuovere la replica di lettura a nuova istanza database di origine.

> **Note**
>
> - Su RDS per MySQL 8,0, la replica ritardata è supportata per MySQL 8.0.28 e versioni successive. In RDS for MySQL 5.7, la replica ritardata è supportata per MySQL 5.7.44 e versioni successive.
> - Utilizza le procedure archiviate per configurare la replica ritardata. Non puoi configurare la replica ritardata con AWS Management Console, the o l' AWS CLI API Amazon RDS.
> - Su RDS for MySQL 5.7.44 e versioni successive di MySQL 5.7 e RDS for MySQL 8.0.28 e versioni successive 8.0, è possibile utilizzare la replica basata su identificatori di transazione globali (GTID) in una configurazione di replica ritardata. Se si utilizza la replica basata su GTID, scegliere la stored procedure mysql.rds_start_replication_until_gtid invece della mysql.rds_start_replication_until. Per ulteriori informazioni sulla replica basata su GTID, consultare Utilizzo della replica basata su GTID.

Argomenti

- Configurazione della replica ritardata durante la creazione della replica di lettura
- Modifica della replica ritardata per una replica di lettura esistente
- Definire una posizione per arrestare la replica su una replica di lettura
- Promozione di una replica di lettura

Configurazione della replica ritardata durante la creazione della replica di lettura

Per configurare la replica ritardata per eventuali repliche di lettura future create da un'istanza database, esegui la stored procedure mysql.rds_set_configuration con il parametro `target delay`.

Per configurare le replica ritardata durante la creazione della replica di lettura

1. Utilizzando un client MySQL, connettersi all'istanza database MySQL che sarà l'origine delle repliche di lettura come l'utente master.
2. Eseguire la procedura archiviata mysql.rds_set_configuration con il parametro `target delay`.

   Ad esempio, eseguire la seguente procedura archiviata per specificare che la replica è ritardata per almeno un'ora (3.600 secondi) per le repliche di lettura create dall'istanza database corrente.

   ```
   call mysql.rds_set_configuration('target delay', 3600);
   ```

   > **Note**
   >
   > Dopo aver eseguito questa procedura memorizzata, qualsiasi replica di lettura creata utilizzando l'API AWS CLI o Amazon RDS viene configurata con una replica ritardata del numero di secondi specificato.

Modifica della replica ritardata per una replica di lettura esistente

Per modificare la replica ritardata per una replica di lettura esistente, esegui la stored procedure mysql.rds_set_source_delay.

Per modificare la replica ritardata per una replica di lettura esistente

1. Utilizzando un client MySQL, connettersi alla replica di lettura come utente master.
2. Usa la procedura archiviata mysql.rds_stop_replication per arrestare la replica.
3. Eseguire la procedura archiviata mysql.rds_set_source_delay.

   Ad esempio, eseguire la seguente stored procedure per specificare che la replica sulla replica di lettura è ritardata per almeno un'ora (3600 secondi).

   ```
   call mysql.rds_set_source_delay(3600);
   ```

4. Usare la procedura archiviata mysql.rds_start_replication per avviare la replica.

Definire una posizione per arrestare la replica su una replica di lettura

Dopo aver arrestato la replica sulla replica di lettura, puoi avviare la replica e poi arrestarla in corrispondenza della posizione del file di log binario specificato utilizzando la procedura archiviata mysql.rds_start_replication_until.

Per avviare la replica su una replica di lettura e arrestare la replica in corrispondenza di una posizione specifica

1. Utilizzando un client MySQL, connettersi all'istanza database MySQL di origine come utente master.
2. Eseguire la procedura archiviata mysql.rds_start_replication_until.

   L'esempio seguente avvia la replica e replica le modifiche fino a raggiungere la posizione `120` nel file di log binario `mysql-bin-changelog.000777`. In caso di disaster recovery, presumere che la posizione `120` si riferisca al momento immediatamente precedente l'errore.

   ```
   call mysql.rds_start_replication_until(
     'mysql-bin-changelog.000777',
     120);
   ```

La replica si arresta automaticamente quando viene raggiunto il punto di arresto. Viene generato il seguente evento RDS: `Replication has been stopped since the replica reached the stop point specified by the rds_start_replication_until stored procedure`.

Promozione di una replica di lettura

Dopo l'arresto della replica, in uno scenario di disaster recovery, puoi promuovere la replica di lettura come nuova istanza database di origine. Per informazioni sulla promozione di una replica di lettura, consulta Promozione di una replica di lettura a istanza database standalone.

Aggiornamento di repliche di lettura con MySQL

Le repliche di lettura sono progettate per supportare query di lettura, ma occasionalmente potrebbe essere necessario eseguire aggiornamenti. Ad esempio, potresti dover aggiungere un indice per ottimizzare i tipi specifici di query che accedono alla replica.

Sebbene sia possibile abilitare gli aggiornamenti impostando il parametro `read_only` su `0` nel gruppo di parametri database per la replica di lettura, si consiglia di non farlo perché questa operazione può causare problemi se la replica di lettura diventa non compatibile con l'istanza database di origine. Per le operazioni di manutenzione, si consiglia di utilizzare le implementazioni blu/verde. Per ulteriori informazioni, consulta Utilizzo delle implementazioni blu/verde per gli aggiornamenti del database.

Se disabiliti la modalità di sola lettura su una replica di lettura, modifica il valore del parametro `read_only` impostandolo su `1` il prima possibile.

Utilizzo di implementazioni Multi-AZ di repliche di lettura con MySQL

Puoi creare una replica di lettura da implementazioni Single-AZ o Multi-AZ di istanze database. Puoi usare implementazioni Multi-AZ per migliorare la durabilità e la disponibilità di dati critici, ma non puoi usare l'istanza secondaria Multi-AZ per inviare query di sola lettura. Puoi invece creare repliche di lettura da istanze database Multi-AZ con traffico elevato per l'offload di query di sola lettura. Se viene eseguito il failover dell'istanza di origine di un'implementazione Multi-AZ all'istanza secondaria, tutte le repliche di lettura passeranno automaticamente a usare l'istanza secondaria (ora primaria) come origine della replica. Per ulteriori informazioni, consulta Configurazione e gestione di un'implementazione multi-AZ.

È possibile creare una replica di lettura come istanza database Multi-AZ. Amazon RDS crea una replica di standby in un'altra zona di disponibilità per il supporto del failover per la replica. La creazione della replica di lettura come un'istanza database Multi-AZ non dipende dal fatto che il database di origine sia un'istanza database Multi-AZ.

Utilizzo di repliche di lettura a cascata con RDS per MySQL

RDS per MySQL supporta le repliche di lettura a cascata. Con le repliche di lettura a cascata, puoi dimensionare le letture senza aggiungere sovraccarico all'istanza database RDS per MySQL di origine.

Con le repliche di lettura a cascata, l'istanza database RDS per MySQL invia i dati alla prima replica di lettura della catena. La replica di lettura invia quindi i dati alla seconda replica della catena e così via. Il risultato finale è che tutte le repliche di lettura nella catena includono le modifiche dall'istanza database RDS per MySQL, ma senza sovraccaricare esclusivamente l'istanza database di origine.

È possibile creare una serie di fino a tre repliche di lettura in una catena da un'istanza database RDS per MySQL di origine. Ad esempio, supponi di avere l'istanza database RDS per MySQL `mysql-main`. Puoi eseguire le operazioni indicate di seguito:

- A partire da `mysql-main`, crea la prima replica di lettura nella catena, `read-replica-1`.
- Da `read-replica-1`, crea quindi la successiva replica di lettura nella catena, `read-replica-2`.
- Da `read-replica-2`, crea infine la terza replica di lettura nella catena, `read-replica-3`.

Non è possibile creare un'altra replica di lettura oltre la terza replica di lettura a cascata nella serie per `mysql-main`. Una serie completa di istanze da un'istanza database RDS per MySQL di origine fino alla fine di una serie di repliche di lettura a cascata può essere composta al massimo da quattro istanze database.

Affinché le repliche di lettura a cascata funzionino, ogni istanza database RDS per MySQL di origine deve avere i backup automatici attivati. Per abilitare i backup automatici in una replica di lettura, crea prima di tutto la replica di lettura, quindi modificala in modo da abilitare i backup automatici. Per ulteriori informazioni, consulta Creazione di una replica di lettura.

Come per qualsiasi replica di lettura, puoi promuovere una replica di lettura appartenente a una cascata. La promozione di una replica di lettura all'interno di una catena di repliche di lettura rimuove la replica dalla catena. Ad esempio, supponi di voler spostare parte del carico di lavoro fuori dall'istanza database `mysql-main` in una nuova istanza usata solo dal reparto contabile. Facendo riferimento alla catena di tre repliche di lettura dell'esempio, decidi di promuovere `read-replica-2`. La catena verrà modificata come segue:

- La promozione `read-replica-2` rimuove l'istanza dalla catena di replica.
  - Ora è un'istanza database completa di lettura/scrittura.
  - Continua a replicare su `read-replica-3`, proprio come prima della promozione.
- L'istanza `mysql-main` continua a venire replicata su `read-replica-1`.

Per ulteriori informazioni sulla promozione delle repliche di lettura, consulta Promozione di una replica di lettura a istanza database standalone.

Monitoraggio delle repliche di lettura MySQL

Per le repliche di lettura MySQL, puoi monitorare il ritardo di replica in Amazon CloudWatch visualizzando la metrica Amazon RDS. `ReplicaLag` Il parametro `ReplicaLag` segnala il valore del campo `Seconds_Behind_Master` del comando `SHOW REPLICA STATUS`.

> **Note**
>
> Versioni precedenti di MySQL utilizzate `SHOW SLAVE STATUS` al posto di `SHOW REPLICA STATUS`. Se si utilizza una versione MySQL prima della 8.0.23, utilizzare `SHOW SLAVE STATUS`.

Le cause comuni del ritardo di replica per MySQL sono le seguenti:

- Interruzione della connessione di rete.
- Scrittura su tabelle con indici diversi su una replica di lettura. Se il parametro `read_only` è impostato su `0` sulla replica di lettura, la replica può interrompersi se la replica di lettura diventa incompatibile con l'istanza database di origine. Dopo aver eseguito operazioni di manutenzione sulla replica di lettura, consigliamo di ripristinare il parametro `read_only` a `1`.
- Uso di un motore di storage non transazionale come MyISAM. La replica è supportata solo per il motore di storage InnoDB su MySQL.

Quando il parametro `ReplicaLag` è 0, la replica ha raggiunto l'istanza del database di origine. Se il parametro `ReplicaLag` restituisce -1, la replica non è attualmente attiva. `ReplicaLag` = -1 equivale a `Seconds_Behind_Master` = `NULL`.

Avvio e arresto della replica con repliche di lettura MySQL

Puoi arrestare e riavviare il processo di replica in un'istanza database Amazon RDS chiamando le stored procedure di sistema mysql.rds_stop_replication e mysql.rds_start_replication. Puoi procedere in questo modo quando esegui la replica tra due istanze Amazon RDS per operazioni a esecuzione prolungata, come la creazione di indici di grandi dimensioni. Devi arrestare e avviare la replica anche durante l'importazione o l'esportazione di database. Per ulteriori informazioni, consulta Importazione dei dati in un database Amazon RDS MariaDB o MySQL con tempi di inattività ridotti e Esportazione di dati da un'istanza database MySQL tramite la replica.

Se la replica viene arrestata per più di 30 giorni consecutivi, manualmente o a causa di un errore di replica, Amazon RDS termina la replica tra l'istanza database di origine e tutte le repliche di lettura. Questo avviene per evitare requisiti di storage maggiori sull'istanza database di origine e tempi di failover prolungati. L'istanza database della replica di lettura continua a essere disponibile. Tuttavia, la replica non può essere ripresa, perché i log binari richiesti dalla replica di lettura vengono eliminati

dall'istanza database di origine una volta terminata la replica. Puoi creare una nuova replica di lettura per l'istanza database di origine per ristabilire la replica.

Risoluzione dei problemi relativi a una replica di lettura MySQL

Per le istanze database MySQL, in alcuni casi le repliche di lettura presentano errori o incoerenze (o entrambi) dei dati tra la replica di lettura e la sua istanza database di origine. Questo problema si verifica quando alcuni eventi log binario (binlog) o log redo InnoDB non vengono scaricati durante un errore della replica di lettura o dell'istanza database di origine. In questi casi, elimina e ricrea manualmente le repliche di lettura. Puoi ridurre le possibilità che si verifichi una situazione di questo tipo impostando i seguenti valori dei parametri: `sync_binlog=1` e `innodb_flush_log_at_trx_commit=1`. Tali impostazioni potrebbero ridurre le prestazioni, per cui ti consigliamo di testare il loro impatto prima di implementare le modifiche nell'ambiente di produzione.

> **Warning**
>
> Nel gruppo di parametri associato all'istanza database di origine, consigliamo di mantenere i valori di questi parametri: `sync_binlog=1` e `innodb_flush_log_at_trx_commit=1`. Questi parametri sono dinamici. Se non vuoi utilizzare queste impostazioni, ti consigliamo di impostare temporaneamente tali valori prima di eseguire qualsiasi operazione sull'istanza database di origine che potrebbe causarne il riavvio. Queste operazioni includono, a titolo esemplificativo ma non esaustivo, il riavvio, il riavvio con failover, l'aggiornamento della versione del database e la modifica della classe di istanza database o della relativa archiviazione. Lo stesso suggerimento si applica alla creazione di nuove repliche di lettura per l'istanza database di origine.
>
> Il mancato rispetto di questa guida aumenta il rischio che le repliche di lettura presentino errori o incoerenze dei dati (o entrambe) tra la replica di lettura e la sua istanza database di origine.

Le tecnologie di replica per MySQL sono asincrone. Per questo motivo, devi occasionalmente aspettarti incrementi del parametro `BinLogDiskUsage` per l'istanza database di origine e del parametro `ReplicaLag` per la replica di lettura. Ad esempio, può verificarsi un elevato volume di scrittura in parallelo nell'istanza database di origine. Al contrario, le operazioni di scrittura nella replica di lettura vengono serializzate usando un singolo thread di I/O, causando un ritardo tra l'istanza di origine e la replica di lettura. Per ulteriori informazioni sulle repliche di sola lettura, consulta le informazioni dettagliate sull'implementazione di repliche nella documentazione di MySQL.

Puoi ridurre il ritardo tra gli aggiornamenti di un'istanza database di origine e i successivi aggiornamenti della replica di lettura in diversi modi, ad esempio:

- Dimensionando una replica di lettura in modo che dimensioni di storage e classe dell'istanza database siano equivalenti all'istanza database di origine.
- Assicurandoti che le impostazioni dei parametri nei gruppi di parametri database usati dall'istanza database di origine e dalla replica di lettura siano compatibili. Per ulteriori informazioni e un esempio, consulta la discussione sul parametro `max_allowed_packet` più avanti in questa sezione.

Amazon RDS monitora lo stato delle repliche di lettura e aggiorna il campo `Replication State` dell'istanza della replica di lettura con il valore `Error` se la replica viene arrestata per qualsiasi motivo. Un possibile esempio è quando query DML in esecuzione nella replica di lettura sono in conflitto con gli aggiornamenti eseguiti nell'istanza database di origine.

Puoi esaminare i dettagli dell'errore associato generato dal motore MySQL visualizzando il campo `Replication Error`. Vengono generati anche eventi che indicano lo stato della replica di lettura, inclusi RDS-EVENT-0045, RDS-EVENT-0046 e RDS-EVENT-0047. Per ulteriori informazioni sugli eventi e sulla sottoscrizione a essi, consulta Utilizzo della notifica degli eventi di Amazon RDS. Se viene restituito un messaggio di errore MySQL, verifica il numero di errore nella documentazione dei messaggi di errore MySQL.

Un problema comune che può provocare errori di replica si verifica quando il valore del parametro `max_allowed_packet` per una replica di lettura è minore del parametro `max_allowed_packet` per l'istanza database di origine. Il parametro `max_allowed_packet` è un parametro personalizzato che puoi impostare in un gruppo di parametri database. Utilizza `max_allowed_packet` per specificare la dimensione massima del codice DML che può essere eseguito nel database. In alcuni casi, il valore `max_allowed_packet` nel gruppo dei parametri database associato alla replica di lettura è minore del valore `max_allowed_packet` nel gruppo dei parametri database associato all'istanza database di origine. In questi casi, il processo di replica può generare l'errore `Packet bigger than 'max_allowed_packet' bytes` e interrompere la replica. Per correggere l'errore, impostare l'istanza database di origine e la replica di lettura in modo che utilizzino i gruppi di parametri database con gli stessi valori del parametro `max_allowed_packet`.

Altre situazioni comuni che possono causare errori di replica includono le seguenti:

- Scrittura in tabelle su una replica di lettura. In alcuni casi, potrebbe essere necessario creare indici su una replica di lettura che sono diversi dagli indici nell'istanza database di origine. In tal

caso, imposta il parametro `read_only` su `0` per creare gli indici. Se scrivi in tabelle sulla replica di lettura, questa operazione potrebbe comportare l'interruzione della replica se la replica di lettura diventa incompatibile con l'istanza database di origine. Dopo aver eseguito attività di manutenzione sulla replica di lettura, ti consigliamo di ripristinare il parametro `read_only` su `1`.

- Utilizzo di un motore di storage non transazionale come MyISAM. Le repliche di lettura richiedono un motore di storage transazionale. La replica è supportata solo per il motore di storage InnoDB su MySQL.
- Utilizzo di query non deterministiche non sicure come `SYSDATE()`. Per ulteriori informazioni, consulta la pagina relativa alla determinazione delle istruzioni sicure e non sicure nel log binario.

Se decidi di poter ignorare un errore con certezza, puoi completare la procedura descritta nella sezione Ignorare l'errore di replica corrente. In caso contrario, puoi prima eliminare la replica di lettura. Quindi crea un'istanza utilizzando lo stesso identificatore istanze DB in modo che l'endpoint resti lo stesso di quello della replica di lettura precedente. Quando un problema relativo alla replica viene risolto, il campo `Replication State` (Stato di replica) cambia in replicating (replica in corso).

Utilizzo della replica basata su GTID

Il seguente contenuto spiega come utilizzare gli identificatori di transazione globali (GTID) con la replica di log binari (binlog) tra le istanze DB di Amazon RDS for MySQL.

Se utilizzi la replica binlog e non hai familiarità con la replica basata su GTID con MySQL, consulta Replica con identificatori di transazione globali nella documentazione di MySQL.

La replica basata su GTID è supportata per tutte le versioni di RDS per MySQL 5.7 e RDS per MySQL versione 8.0.26 e versioni successive a MySQL 8.0. Tutte le istanze database MySQL in una configurazione di replica devono rispettare questo requisito.

Argomenti

- Identificatori globali di transazione (GTID)
- Parametri per la replica basata su GTID
- Configurazione della replica basata su GTID per le nuove repliche di lettura
- Configurazione della replica basata su GTID per le repliche di lettura esistenti.
- Disabilitazione della replica basata su GTID per un'istanza database MySQL con repliche di lettura

Identificatori globali di transazione (GTID)

Gli identificatori globali di transazione (GTID) sono identificatori univoci generati per le transazioni MySQL sottoposte a commit. Puoi utilizzare i GTID per semplificare la replica basata sui log binari e facilitare la risoluzione dei problemi.

MySQL utilizza due diversi tipi di transazioni per la replica basata sui log binari:

- Transazioni GTID – Transazioni identificate da un GTID.
- Transazioni anonime – Transazioni a cui non è assegnato un GTID.

In una configurazione di replica, i GTID sono univoci in tutte le istanze database. I GTID semplificano la configurazione della replica perché, quando vengono utilizzati, non è necessario fare riferimento alle posizioni nel file di log. I GTID semplificano anche la registrazione delle transazioni replicate e verificano che l'istanza di origine e le repliche siano coerenti.

Puoi utilizzare la replica basata su GTID per replicare i dati con le repliche di lettura di RDS for MySQL. Puoi configurare la replica basata su GTID quando crei le nuove repliche di lettura oppure puoi convertire le repliche di lettura esistenti in modo che usino la replica basata su GTID.

Puoi utilizzare la replica basata su GTID anche in una configurazione di replica ritardata con RDS for MySQL. Per ulteriori informazioni, consulta Configurazione della replica ritardata con MySQL.

Parametri per la replica basata su GTID

Utilizzare i parametri seguenti per configurare la replica basata su GTID.

Parametro	Valori validi	Descrizione
`gtid_mode`	`OFF`, `OFF_PERMISSIVE` , `ON_PERMISSIVE` , `ON`	`OFF` indica che le nuove transazioni sono anonime, ovvero non hanno GTID, e che una transazione deve essere anonima per poter essere replicata. `OFF_PERMISSIVE` indica che le nuove transazioni sono anonime, ma tutte le transazio ni possono essere replicate.

Parametro	Valori validi	Descrizione
		`ON_PERMISSIVE` indica che le nuove transazioni hanno GTID assegnati, ma tutte le transazioni possono essere replicate. `ON` indica che le nuove transazioni hanno GTID assegnati e che una transazione deve avere un GTID per poter essere replicata.
`enforce_gtid_consistency`	`OFF`, `ON`, `WARN`	`OFF` consente alle transazioni di violare la coerenza GTID. `ON` impedisce alle transazioni di violare la coerenza GTID. `WARN` consente alle transazioni di violare la consistenza GTID, ma genera un avviso quando si verifica una violazione.

> **Note**
>
> Nel, il parametro appare come. AWS Management Console`gtid_mode` `gtid-mode`

Per la replica basata su GTID, utilizza queste impostazioni per il gruppo di parametri dell'istanza database o per la replica di lettura:

- `ON` e `ON_PERMISSIVE` si applicano solo alla replica in uscita da un'istanza database RDS. Entrambi questi valori fanno sì che l'istanza database RDS utilizzi i GTID per le transazioni replicate. `ON` richiede che anche il database di destinazione utilizzi la replica basata su GTID. Per `ON_PERMISSIVE` la replica basata su GTID è opzionale sul database di destinazione.
- `OFF_PERMISSIVE`, se impostato, significa che le istanze database RDS possono accettare la replica in ingresso da un database di origine. Possono farlo indipendentemente dal fatto che il database di origine utilizzi la replica basata su GTID.
- `OFF`, se impostato, significa che le istanze database RDS accettano solo la replica in ingresso da database di origine che non utilizzano la replica basata su GTID.

Per ulteriori informazioni sui gruppi di parametri, consultare Utilizzo di gruppi di parametri.

Configurazione della replica basata su GTID per le nuove repliche di lettura

Quando la replica basata su GTID è abilitata per un'istanza database di RDS for MySQL, la replica basata su GTID viene configurata automaticamente per le repliche di lettura di un'istanza database.

Per abilitare la replica basata su GTID per le nuove repliche di lettura

1. Verificare che il gruppo di parametri associato all'istanza database abbia le impostazioni dei parametri seguenti:

   - `gtid_mode` – `ON` o `ON_PERMISSIVE`
   - `enforce_gtid_consistency` – `ON`

   Per ulteriori informazioni sull'impostazione dei parametri di configurazione mediante i gruppi di parametri, consultare Utilizzo di gruppi di parametri.
2. Se è stato modificato un gruppo di parametri dell'istanza database, riavviare l'istanza. Per ulteriori informazioni su come effettuare questa operazione, consultare Riavvio di un'istanza database.
3. Creare una o più repliche di lettura dell'istanza database. Per ulteriori informazioni su come effettuare questa operazione, consultare Creazione di una replica di lettura.

Amazon RDS prova a stabilire una replica basata su GTID tra l'istanza database di MySQL e le repliche di lettura utilizzando `MASTER_AUTO_POSITION`. Se il tentativo non riesce, Amazon RDS utilizza le posizioni del file di log per la replica con le repliche di lettura. Per ulteriori informazioni su `MASTER_AUTO_POSITION`, consultare l'argomento relativo al posizionamento automatico dei GTID nella documentazione di MySQL.

Configurazione della replica basata su GTID per le repliche di lettura esistenti.

Per un'istanza database di RDS for MySQL esistente con repliche di lettura che non utilizzano la replica basata su GTID, è possibile configurare la replica basata su GTID tra l'istanza database e le repliche di lettura.

Per abilitare la replica basata su GTID per le repliche di lettura esistenti

1. Se l'istanza database o qualsiasi replica di lettura sta utilizzando RDS versione 8.0 for MySQL versione 8.0.26 o precedente, aggiornare l'istanza database o la replica di lettura a MySQL 8.0.26 o versione successiva alla 8.0. Tutte le versioni di RDS per MySQL 5.7 supportano la replica basata su GTID.

   Per ulteriori informazioni, consulta Aggiornamento del motore di database MySQL.

2. (Facoltativo) Reimpostare i parametri GTID e verificare il comportamento dell'istanza database e delle repliche di lettura:

   a. Verificare che il gruppo di parametri associato all'istanza database e ogni replica di lettura abbiano il parametro `enforce_gtid_consistency` impostato su `WARN`.

      Per ulteriori informazioni sull'impostazione dei parametri di configurazione mediante i gruppi di parametri, consultare Utilizzo di gruppi di parametri.

   b. Se è stato modificato un gruppo di parametri dell'istanza database, riavviare l'istanza. Se il gruppo di parametri è stato modificato per una replica di lettura, riavviare la replica.

      Per ulteriori informazioni, consulta Riavvio di un'istanza database.

   c. Eseguire l'istanza database e le repliche di lettura con il normale carico di lavoro e monitorare i file di log.

      Se vengono visualizzati avvisi relativi a transazioni incompatibili con GTID, modificare l'applicazione in modo che usi solo caratteristiche compatibili con GTID. Verificare che l'istanza database non stia generando avvisi relativi a transazioni incompatibili con GTID prima di procedere alla prossima fase.

3. Reimpostare i parametri GTID per la replica basata su GTID che consente le transazioni anonime finché le repliche di lettura non ne completano l'elaborazione.

   a. Verificare che il gruppo di parametri associato all'istanza database e ogni replica di lettura abbiano le impostazioni dei parametri seguenti:

      - `gtid_mode` – `ON_PERMISSIVE`
      - `enforce_gtid_consistency` – `ON`

   b. Se è stato modificato un gruppo di parametri dell'istanza database, riavviare l'istanza. Se il gruppo di parametri è stato modificato per una replica di lettura, riavviare la replica.

4. Attendere il completamento della replica di tutte le transazioni anonime. Per verificare che vengano replicate, procedere come descritto di seguito:

   a. Eseguire questa istruzione sull'istanza DB primaria.

   ```
   SHOW MASTER STATUS;
   ```

   Annotare i valori nelle colonne `File` e `Position`.

   b. In ogni replica di lettura, utilizzare le informazioni su file e posizione presenti nell'istanza di origine menzionata nella fase precedente per eseguire la query seguente.

   ```
   SELECT MASTER_POS_WAIT('file', position);
   ```

   Ad esempio, se il nome del file è `mysql-bin-changelog.000031` e la posizione è `107`, eseguire l'istruzione seguente.

   ```
   SELECT MASTER_POS_WAIT('mysql-bin-changelog.000031', 107);
   ```

   Se la replica di lettura si trova dopo la posizione specificata, la query la restituisce immediatamente. In caso contrario, la funzione entra in attesa. Passare alla fase successiva quando la query restituisce risposte per tutte le repliche di lettura.

5. Reimpostare i parametri GTID solo per la replica basata su GTID.

   a. Verificare che il gruppo di parametri associato all'istanza database e ogni replica di lettura abbiano le impostazioni dei parametri seguenti:

   - `gtid_mode` – `ON`
   - `enforce_gtid_consistency` – `ON`

   b. Riavviare l'istanza database e ogni replica di lettura.

6. In ogni replica di lettura completare la procedura seguente.

   ```
   CALL mysql.rds_set_master_auto_position(1);
   ```

Disabilitazione della replica basata su GTID per un'istanza database MySQL con repliche di lettura

Puoi disabilitare la replica basata su GTID per un un'istanza database MySQL con repliche di lettura.

Per disabilitare la replica basata su GTID per un'istanza database MySQL con repliche di lettura

1. Su ogni replica letta, eseguire la procedura seguente:

   ```
   CALL mysql.rds_set_master_auto_position(0);
   ```

2. Reimpostare `gtid_mode` su `ON_PERMISSIVE`.

   a. Verifica che il gruppo di parametri associato all'istanza database MySQL e ogni replica di lettura abbiano `gtid_mode` impostato su `ON_PERMISSIVE`.

      Per ulteriori informazioni sull'impostazione dei parametri di configurazione mediante i gruppi di parametri, consultare Utilizzo di gruppi di parametri.

   b. Riavvia l'istanza database MySQL e ogni replica di lettura. Per ulteriori informazioni sul riavvio, consultare Riavvio di un'istanza database.

3. Reimpostare `gtid_mode` su `OFF_PERMISSIVE`.

   a. Verifica che il gruppo di parametri associato all'istanza database MySQL e ogni replica di lettura abbiano `gtid_mode` impostato su `OFF_PERMISSIVE`.

   b. Riavvia l'istanza database MySQL e ogni replica di lettura.

4. Attendere che tutte le transazioni GTID vengano applicate a tutte le repliche di lettura. Per verificare che vengano applicate, procedi nel seguente modo:

   a. Sull'istanza database MySQL, esegui il comando `SHOW MASTER STATUS`.

      Il risultato dovrebbe essere simile al seguente.

      ```
      File                      Position
      ----------------------------------
      mysql-bin-changelog.000031      107
      ----------------------------------
      ```

      Annotare il file e la posizione nell'output.

b. In ogni replica letta, utilizzate le informazioni sul file e sulla posizione dalla relativa istanza di origine nel passaggio precedente per eseguire la seguente query:

   Per MySQL 8.0.26 e versioni successive di MySQL 8.0

   ```
   SELECT SOURCE_POS_WAIT('file', position);
   ```

   Per le versioni MySQL 5.7

   ```
   SELECT MASTER_POS_WAIT('file', position);
   ```

   Ad esempio, se il nome del file è `mysql-bin-changelog.000031` e la posizione è`107`, esegui la seguente istruzione:

   Per MySQL 8.0.26 e versioni successive di MySQL 8.0

   ```
   SELECT SOURCE_POS_WAIT('mysql-bin-changelog.000031', 107);
   ```

   Per le versioni MySQL 5.7

   ```
   SELECT MASTER_POS_WAIT('mysql-bin-changelog.000031', 107);
   ```

5. Reimposta i parametri GTID per disabilitare la replica basata su GTID.

   a. Verifica che il gruppo di parametri associato all'istanza database MySQL e ogni replica di lettura abbiano le impostazioni dei parametri seguenti:

      - `gtid_mode` – `OFF`
      - `enforce_gtid_consistency` – `OFF`

   b. Riavvia l'istanza database MySQL e ogni replica di lettura.

Configurazione della replica della posizione del file di log binario con un'istanza di origine esterna.

Puoi impostare la replica fra un'istanza database RDS for MySQL o MariaDB e un’istanza MySQL o MariaDB che è esterna ad Amazon RDS usando la replica del file di log binario.

Argomenti

Prima di iniziare

È possibile configurare la replica utilizzando la posizione del file di log binario delle transazioni replicate.

Le autorizzazioni necessarie per avviare la replica in un'istanza database Amazon RDS sono limitate e non disponibili per l'utente master Amazon RDS. Per questo motivo, assicurati di usare i comandi mysql.rds_set_external_master e mysql.rds_start_replication in Amazon RDS per configurare la replica tra il database live e il database Amazon RDS.

Per impostare il formato di logging binario per un database MySQL o MariaDB, aggiornare il parametro `binlog_format`. Se l'istanza database utilizza il gruppo di parametri di istanza database predefinito, crea un nuovo gruppo di parametri di istanza database per modificare le impostazioni `binlog_format`. Ti consigliamo di mantenere le impostazioni predefinite per `binlog_format`, che è `MIXED`. Tuttavia, puoi anche impostare `binlog_format` su `ROW` o `STATEMENT` se hai bisogno di un formato di registro binario (binlog) specifico. Riavvia l'istanza database affinché venga applicata la modifica.

Per ulteriori informazioni sull'impostazione del parametro `binlog_format`, consulta Configurazione del log binario di MySQL Per ulteriori informazioni sulle implicazioni dei vari tipi di replica MySQL, consulta la pagina relativa a vantaggi e svantaggi della replica basata su istruzioni e basata su riga nella documentazione di MySQL.

> **Note**
>
> A partire dalla versione 8.0.36 di RDS per MySQL, Amazon RDS non replica il database. `mysql` Pertanto, se ci sono utenti nel database esterno di cui hai bisogno nella replica di Amazon RDS, assicurati di crearli manualmente.

Configurazione della replica della posizione del file di log binario con un'istanza di origine esterna.

Seguire queste linee guida quando si imposta un'istanza di origine esterna e una replica su Amazon RDS:

- Monitorare gli eventi di failover per l'istanza database di Amazon RDS che rappresenta la replica. In caso di failover, l'istanza database che rappresenta la replica potrebbe essere ricreata in un nuovo host con un indirizzo di rete diverso. Per informazioni su come monitorare gli eventi di failover, consulta Utilizzo della notifica degli eventi di Amazon RDS.
- Conservare i binlog sull'istanza di origine finché non si ha la conferma che siano stati applicati alla replica. Conservando questi file, si è certi di poter ripristinare l'istanza di origine in caso di errori.
- Attivare i backup automatici sull'istanza database di Amazon RDS. L'attivazione dei backup automatici assicura il ripristino della replica a un punto temporale specifico nel caso fosse necessario risincronizzare l'istanza di origine e la replica. Per informazioni su backup e point-in-time ripristino, consulta. Backup, ripristino ed esportazione dei dati

Per configurare la replica della posizione del file di log binario con un'istanza di origine esterna

1. Rendere l'istanza MySQL o MariaDB di origine di sola lettura.

```
mysql> FLUSH TABLES WITH READ LOCK;
mysql> SET GLOBAL read_only = ON;
```

2. Eseguire il comando `SHOW MASTER STATUS` nell'istanza database di MySQL o MariaDB di origine per determinare la posizione del binlog.

   Viene restituito un output simile all'esempio seguente.

```
File                        Position
-------------------------------------
 mysql-bin-changelog.000031      107
-------------------------------------
```

3. Copiare il database dall'istanza esterna all'istanza database Amazon RDS usando `mysqldump`. Per database di dimensioni particolarmente elevate, è possibile utilizzare la procedura in Importazione dei dati in un database Amazon RDS MariaDB o MySQL con tempi di inattività ridotti.

   Per LinuxmacOS, oUnix:

```
mysqldump --databases database_name \
    --single-transaction \
    --compress \
    --order-by-primary \
```

```
    -u local_user \
    -plocal_password | mysql \
        --host=hostname \
        --port=3306 \
        -u RDS_user_name \
        -pRDS_password
```

Per Windows:

```
mysqldump --databases database_name ^
    --single-transaction ^
    --compress ^
    --order-by-primary ^
    -u local_user ^
    -plocal_password | mysql ^
        --host=hostname ^
        --port=3306 ^
        -u RDS_user_name ^
        -pRDS_password
```

> **Note**
>
> Assicurarsi che non siano presenti spazi tra l'opzione `-p` e la password immessa.

Utilizzare le opzioni `--host`, `--user (-u)`, `--port` e `-p` nel comando `mysql` per specificare il nome host, il nome utente, la porta e la password per la connessione all'istanza database Amazon RDS. Il nome host è il nome DNS (Domain Name Service) dell'endpoint dell'istanza database di Amazon RDS, ad esempio `myinstance.123456789012.us-east-1.rds.amazonaws.com`. È possibile trovare il valore dell'endpoint nei dettagli dell'istanza nella AWS Management Console.

4. Rendere nuovamente scrivibile l'istanza MySQL o MariaDB di origine.

```
mysql> SET GLOBAL read_only = OFF;
mysql> UNLOCK TABLES;
```

Per ulteriori informazioni sulla creazione di backup da utilizzare con la replica, vedere la documentazione di MySQL.

5. Nel AWS Management Console, aggiungi l'indirizzo IP del server che ospita il database esterno al gruppo di sicurezza del cloud privato virtuale (VPC) per l'istanza database Amazon RDS. Per ulteriori informazioni sulla modifica di un gruppo di sicurezza VPC, consulta Gruppi di sicurezza per il VPC nella Guida per l'utente di Amazon Virtual Private Cloud.

   L'indirizzo IP può cambiare quando vengono soddisfatte le seguenti condizioni:

   - Si sta utilizzando un indirizzo IP pubblico per la comunicazione tra l'istanza di origine esterna e l'istanza database.
   - L'istanza di origine esterna è stata arrestata e riavviata.

   Se queste condizioni vengono soddisfatte, verificare l'indirizzo IP prima di aggiungerlo.

   Potrebbe anche essere necessario configurare la rete locale per consentire le connessioni dall'indirizzo IP dell'istanza database di Amazon RDS, affinché possa comunicare con l'istanza MySQL o MariaDB esterna. Per individuare l'indirizzo IP dell'istanza database di Amazon RDS, utilizzare il comando `host`.

   ```
   host db_instance_endpoint
   ```

   Il nome host è il nome DNS dall'endpoint dell'istanza database di Amazon RDS.

6. Utilizzando il client scelto, eseguire la connessione all'istanza esterna e creare un utente da utilizzare per la replica. Utilizza questo account unicamente per la replica e limitalo al dominio personale per aumentare la sicurezza. Di seguito è riportato un esempio.

   ```
   CREATE USER 'repl_user'@'mydomain.com' IDENTIFIED BY 'password';
   ```

   > **Note**
   >
   > Specifica una password diversa dal prompt mostrato qui come best practice per la sicurezza.

7. Per l'istanza esterna, concedere i privilegi `REPLICATION CLIENT` e `REPLICATION SLAVE` all'utente della replica. Per concedere ad esempio i privilegi `REPLICATION CLIENT` e `REPLICATION SLAVE` su tutti i database per l'utente "`repl_user`" del proprio dominio, eseguire questo comando.

```
GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user'@'mydomain.com';
```

8. Definire l'istanza database di Amazon RDS come replica. A tale scopo, connettersi innanzitutto all'istanza database di Amazon RDS come l'utente master. Quindi, identificare il database MySQL o MariaDB esterno come istanza di origine utilizzando il comando mysql.rds_set_external_master. Utilizzare il nome e la posizione del file di log master recuperati nella fase 2. Di seguito è riportato un esempio.

```
CALL mysql.rds_set_external_master ('mymasterserver.mydomain.com', 3306,
 'repl_user', 'password', 'mysql-bin-changelog.000031', 107, 0);
```

> **Note**
>
> In RDS for MySQL puoi scegliere di usare la replica ritardata eseguendo invece la procedura archiviata mysql.rds_set_external_master_with_delay. Su RDS for MySQL, una ragione per utilizzare la replica ritardata è attuare il ripristino di emergenza con la procedura archiviata mysql.rds_start_replication_until. Attualmente RDS for MariaDB supporta la replica ritardata ma non supporta la procedura `mysql.rds_start_replication_until`.

9. Nell'istanza database di Amazon RDS eseguire il comando mysql.rds_start_replication per avviare la replica.

```
CALL mysql.rds_start_replication;
```

Configurazione multi-source-replication per RDS for MySQL

Con la replica da più fonti, puoi configurare un'istanza DB Amazon RDS for MySQL come replica che riceve eventi di log binari da più di un'istanza DB di origine RDS for MySQL. La replica da più fonti è supportata per le istanze DB RDS for MySQL che eseguono le seguenti versioni del motore:

- 8.0.35 e versioni secondarie successive
- 5.7.44 e versioni secondarie successive

Per informazioni sulla replica multi-source MySQL, consulta MySQL Multi-Source Replication nella documentazione di MySQL. La documentazione MySQL contiene informazioni dettagliate su questa

funzionalità, mentre questo argomento descrive come configurare e gestire i canali di replica multi-source sulle istanze DB RDS per MySQL.

Argomenti

Casi d'uso per la replica da più fonti

I seguenti casi sono buoni candidati per l'utilizzo della replica da più fonti su RDS per MySQL:

- Applicazioni che devono unire o combinare più shard su istanze DB separate in un unico shard.
- Applicazioni che devono generare report da dati consolidati da più fonti.
- Requisiti per creare backup consolidati a lungo termine dei dati distribuiti tra più istanze DB RDS for MySQL.

Considerazioni e best practice per la replica da più fonti

Prima di utilizzare la replica da più fonti su RDS for MySQL, esamina le seguenti considerazioni e best practice:

- Assicurati che un'istanza DB configurata come replica da più fonti disponga di risorse sufficienti come throughput, memoria, CPU e IOPS per gestire il carico di lavoro proveniente da più istanze di origine.
- Monitora regolarmente l'utilizzo delle risorse sulla replica da più fonti e regola lo storage o la configurazione dell'istanza per gestire il carico di lavoro senza sovraccaricare le risorse.
- È possibile configurare la replica multithread su una replica da più fonti impostando la variabile di sistema su un valore maggiore di. `replica_parallel_workers` `0` In questo caso, il numero di thread assegnati a ciascun canale è il valore di questa variabile, più un thread di coordinamento per gestire i thread dell'applicatore.

- Configura i filtri di replica in modo appropriato per evitare conflitti. Per replicare un intero database su un altro database su una replica, è possibile utilizzare l'opzione. `--replicate-rewrite-db` Ad esempio, è possibile replicare tutte le tabelle del database A nel database B su un'istanza di replica. Questo approccio può essere utile quando tutte le istanze di origine utilizzano la stessa convenzione di denominazione dello schema. Per informazioni sull'`--replicate-rewrite-db`opzione, consulta Opzioni e variabili del server di replica nella documentazione di MySQL.
- Per evitare errori di replica, evita di scrivere sulla replica. Si consiglia di abilitare il `read_only` parametro sulle repliche da più fonti per bloccare le operazioni di scrittura. In questo modo è possibile eliminare i problemi di replica causati da operazioni di scrittura in conflitto.
- Per aumentare le prestazioni delle operazioni di lettura, ad esempio ordinamenti e join con carichi elevati, eseguite sulla replica da più fonti, prendi in considerazione l'utilizzo di RDS Optimized Reads. Questa funzionalità può essere utile per le interrogazioni che dipendono da tabelle temporanee di grandi dimensioni o da file di ordinamento. Per ulteriori informazioni, consulta the section called “Prestazioni delle query migliorate con RDS Optimized Reads”.
- Per ridurre al minimo il ritardo nella replica e migliorare le prestazioni di una replica da più fonti, è consigliabile abilitare scritture ottimizzate. Per ulteriori informazioni, consulta the section called “Prestazioni di scrittura migliorate con Scritture ottimizzate per Amazon RDS per MySQL”.
- Esegui operazioni di gestione (come la modifica della configurazione) su un canale alla volta ed evita di apportare modifiche a più canali da più connessioni. Queste pratiche possono portare a conflitti nelle operazioni di replica. Ad esempio, l'esecuzione `rds_skip_repl_error_for_channel` simultanea di `rds_start_replication_for_channel` procedure da più connessioni può causare il salto di eventi su un canale diverso da quello previsto.
- Puoi abilitare i backup su un'istanza di replica da più fonti ed esportare i dati da quell'istanza in un bucket Amazon S3 per archiviarli a lungo termine. Tuttavia, è importante configurare anche i backup con una conservazione appropriata sulle singole istanze di origine. Per informazioni sull'esportazione dei dati delle istantanee in Amazon S3, consulta. the section called “Esportazione dei dati dello snapshot DB in Simple Storage Service (Amazon S3)”
- Per distribuire il carico di lavoro di lettura su una replica da più fonti, puoi creare repliche di lettura da una replica da più fonti. È possibile posizionare queste repliche di lettura in diversi modi Regioni AWS in base ai requisiti dell'applicazione. Per ulteriori informazioni sulle repliche di lettura, consulta the section called “Uso delle repliche di lettura MySQL”.

Prerequisiti per la replica da più fonti

Prima di configurare la replica da più fonti, completare i seguenti prerequisiti.

- Assicurati che ogni istanza DB RDS for MySQL di origine abbia i backup automatici abilitati. L'abilitazione dei backup automatici consente la registrazione binaria. Per informazioni su come abilitare i backup automatici, consulta. the section called “Abilitazione dei backup automatici”
- Per evitare errori di replica, si consiglia di bloccare le operazioni di scrittura sulle istanze DB di origine. È possibile farlo impostando il `read-only` parametro su `ON` in un gruppo di parametri personalizzato collegato all'istanza DB di origine RDS for MySQL. È possibile utilizzare AWS Management Console o the AWS CLI per creare un nuovo gruppo di parametri personalizzato o per modificarne uno esistente. Per ulteriori informazioni, consulta the section called “Creazione di un gruppo di parametri del database” e the section called “Modifica di parametri in un gruppo di parametri del database”.
- Per ogni istanza DB di origine, aggiungi l'indirizzo IP dell'istanza al gruppo di sicurezza Amazon Virtual Private Cloud (VPC) per l'istanza DB multi-source. Per identificare l'indirizzo IP di un'istanza DB di origine, puoi eseguire il comando. `dig` *`RDS Endpoint`* Esegui il comando da un'istanza Amazon EC2 nello stesso VPC dell'istanza DB multi-sorgente di destinazione.
- Per ogni istanza DB di origine, utilizza un client per connetterti all'istanza DB e crea un utente del database con i privilegi richiesti per la replica, come nell'esempio seguente.

```
CREATE USER 'repl_user' IDENTIFIED BY 'password';
GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user';
```

Configurazione di canali di replica da più fonti su istanze DB RDS per MySQL

La configurazione dei canali di replica da più fonti è simile alla configurazione della replica da un'unica fonte. Per la replica da più fonti, è innanzitutto necessario attivare la registrazione binaria sull'istanza di origine. Quindi, si importano i dati dalle sorgenti alla replica da più fonti. Quindi, si avvia la replica da ciascuna fonte utilizzando le coordinate del log binario o utilizzando il posizionamento automatico GTID.

Per configurare un'istanza DB RDS for MySQL come replica multi-source di due o più istanze DB RDS for MySQL, procedi nel seguente modo.

Argomenti

- Fase 1: Importazione dei dati dalle istanze DB di origine alla replica multisorgente

- Passaggio 2: avviare la replica dalle istanze DB di origine alla replica multisorgente

Fase 1: Importazione dei dati dalle istanze DB di origine alla replica multisorgente

Esegui i passaggi seguenti su ogni istanza DB di origine.

Prima di importare i dati da un'origine alla replica da più fonti, determina il file di registro binario e la posizione correnti eseguendo il `SHOW MASTER STATUS` comando. Prendi nota di questi dettagli per utilizzarli nel passaggio successivo. In questo output di esempio, il file è `mysql-bin-changelog.000031` e la posizione è`107`.

```
File                        Position
-----------------------------------
mysql-bin-changelog.000031      107
-----------------------------------
```

Ora copia il database dall'istanza DB di origine alla replica multisorgente utilizzando`mysqldump`, come nell'esempio seguente.

```
mysqldump --databases database_name \
 --single-transaction \
 --compress \
 --order-by-primary \
 -u RDS_user_name \
 -p RDS_password \
 --host=RDS Endpoint | mysql \
 --host=RDS Endpoint \
 --port=3306 \
 -u RDS_user_name \
-p RDS_password
```

Dopo aver copiato il database, è possibile impostare il parametro di sola lettura `OFF` su nell'istanza DB di origine.

Passaggio 2: avviare la replica dalle istanze DB di origine alla replica multisorgente

Per ogni istanza DB di origine, utilizza le credenziali dell'utente principale per connetterti all'istanza ed esegui le due stored procedure seguenti. Queste stored procedure configurano la replica su un canale e avviano la replica. Questo esempio utilizza il nome e la posizione del file binlog dell'output di esempio del passaggio precedente.

```
CALL mysql.rds_set_external_source_for_channel('mysourcehost.example.com', 3306,
 'repl_user', 'password', 'mysql-bin-changelog.000031', 107, 0, 'channel_1');
CALL mysql.rds_start_replication_for_channel('channel_1');
```

Per ulteriori informazioni sull'utilizzo di queste stored procedure e di altre per configurare e gestire i canali di replica, vedere. the section called “Gestione della replica da più fonti”

Utilizzo di filtri con replica da più fonti

È possibile utilizzare i filtri di replica per specificare con quali database e tabelle vengono replicati in una replica da più fonti. I filtri di replica possono includere database e tabelle nella replica o escluderli dalla replica. Per ulteriori informazioni sui filtri di replica, vedere. the section called “Configurazione dei filtri di replica con MySQL”

Con la replica da più fonti, puoi configurare i filtri di replica a livello globale o a livello di canale. Il filtraggio a livello di canale è disponibile solo con le istanze DB supportate che eseguono la versione 8.0. Gli esempi seguenti mostrano come configurare i filtri a livello globale o a livello di canale.

Tieni presente i seguenti requisiti e comportamenti con il filtraggio nella replica da più fonti:

- Sono obbligatorie le virgolette (``) attorno ai nomi dei canali.
- Se modificate i filtri di replica nel gruppo di parametri, le repliche da più fonti `sql_thread` per tutti i canali con aggiornamenti vengono riavviate per applicare le modifiche in modo dinamico. Se un aggiornamento riguarda un filtro globale, tutti i canali di replica nello stato di esecuzione vengono riavviati.
- Tutti i filtri globali vengono applicati prima di qualsiasi filtro specifico del canale.
- Se un filtro viene applicato globalmente e a livello di canale, viene applicato solo il filtro a livello di canale. Ad esempio, se i filtri lo sonoreplicate_ignore_db="db1,`channel_22`:db2", allora `replicate_ignore_db` set to `db1` viene applicato a tutti i canali tranne e `channel_22` ignora solo le `channel_22` modifiche apportate da. `db2`

Esempio 1: impostazione di un filtro globale

Nell'esempio seguente, il `temp_data` database è escluso dalla replica in ogni canale.

Per LinuxmacOS, oUnix:

```
aws rds modify-db-parameter-group \
```

```
--db-parameter-group-name myparametergroup \
--parameters "ParameterName=replicate-ignore-
db,ParameterValue='temp_data',ApplyMethod=immediate"
```

Esempio 2: impostazione di un filtro a livello di canale

Nell'esempio seguente, le modifiche dal `sample22` database sono incluse solo nel canale. `channel_22` Allo stesso modo, le modifiche dal `sample99` database sono incluse solo nel canale`channel_99`.

Per LinuxmacOS, oUnix:

```
aws rds modify-db-parameter-group \
--db-parameter-group-name myparametergroup \
--parameters "ParameterName=replicate-do-db,ParameterValue='\`channel_22\`:sample22,
\`channel_99\`:sample99',ApplyMethod=immediate"
```

Monitoraggio dei canali di replica da più fonti

È possibile monitorare singoli canali in una replica da più fonti utilizzando i seguenti metodi:

- Per monitorare lo stato di tutti i canali o di un canale specifico, connettiti alla replica multisorgente ed esegui il comando or. `SHOW REPLICA STATUS` `SHOW REPLICA STATUS FOR CHANNEL '`*`channel_name`*`'` Per ulteriori informazioni, vedere Verifica dello stato della replica nella documentazione di MySQL.
- Per ricevere una notifica quando un canale di replica viene avviato, interrotto o rimosso, utilizza la notifica degli eventi RDS. Per ulteriori informazioni, consulta the section called “Utilizzo della notifica degli eventi di Amazon RDS”.
- Per monitorare il ritardo per un canale specifico, controlla la relativa `ReplicationChannelLag` metrica. I punti dati per questa metrica hanno un periodo di 60 secondi (1 minuto) e sono disponibili per 15 giorni. Per individuare il ritardo del canale di replica per un canale, utilizzate l'identificatore di istanza e il nome del canale di replica. Per ricevere una notifica quando questo ritardo supera una determinata soglia, puoi impostare un allarme. CloudWatch Per ulteriori informazioni, consulta the section called “Monitoraggio di RDS con CloudWatch”.

Limitazioni per la replica da più fonti su RDS for MySQL

Le seguenti limitazioni si applicano alla replica da più fonti su RDS for MySQL:

- Attualmente, RDS for MySQL supporta la configurazione di un massimo di 15 canali per una replica multi-sorgente.
- Un'istanza di replica di lettura non può essere configurata come replica da più fonti.
- Per configurare la replica da più fonti su RDS for MySQL Running Engine versione 5.7, è necessario abilitare Performance Schema sull'istanza di replica. L'attivazione dello schema delle prestazioni è facoltativa su RDS for MySQL con motore in esecuzione versione 8.0.
- Per RDS for MySQL Running Engine versione 5.7, i filtri di replica si applicano a tutti i canali di replica. Per RDS for MySQL con motore in esecuzione versione 8.0, è possibile configurare filtri che si applicano a tutti i canali di replica o ai singoli canali.
- Il ripristino di un'istantanea RDS o l'esecuzione di un P-Restore (PITR) non oint-in-time ripristinano le configurazioni dei canali di replica da più fonti.
- Quando si crea una replica di lettura di una replica da più fonti, vengono replicati solo i dati dell'istanza da più fonti. Non ripristina alcuna configurazione dei canali.
- MySQL non supporta la configurazione di un numero diverso di parallel worker per ogni canale. Ogni canale riceve lo stesso numero di worker paralleli in base al `replica_parallel_workers` valore.

Le seguenti limitazioni aggiuntive si applicano se la destinazione di replica da più fonti è un cluster DB Multi-AZ:

- È necessario configurare un canale per un'istanza RDS for MySQL di origine prima che avvenga qualsiasi scrittura su quell'istanza.
- Ogni istanza di RDS for MySQL di origine deve avere la replica basata su GTID abilitata.
- Un evento di failover sul cluster DB rimuove la configurazione di replica da più fonti. Il ripristino di tale configurazione richiede la ripetizione dei passaggi di configurazione.

Configurazione di cluster active-active per RDS for MySQL

È possibile configurare un cluster active-active per RDS for MySQL utilizzando il plug-in MySQL Group Replication. Il plug-in Group Replication è supportato per le istanze DB RDS for MySQL che eseguono la versione 8.0.35 e versioni secondarie successive.

Per informazioni sulla replica di gruppo MySQL, vedere Replica di gruppo nella documentazione di MySQL. La documentazione di MySQL contiene informazioni dettagliate su questa funzionalità, mentre questo argomento descrive come configurare e gestire il plug-in sulle istanze DB di RDS for MySQL.

> **Note**
>
> Per motivi di brevità, tutte le menzioni di cluster «active-active» in questo argomento si riferiscono ai cluster attivi-attivi che utilizzano il plug-in MySQL Group Replication.

Argomenti

Casi d'uso per cluster attivi-attivi

I seguenti casi sono buoni candidati per l'utilizzo di cluster attivi-attivi:

- Applicazioni che richiedono tutte le istanze DB del cluster per supportare le operazioni di scrittura. Il plug-in Group Replication mantiene i dati coerenti su ogni istanza DB nel cluster active-active. Per ulteriori informazioni su come funziona, consulta Group Replication nella documentazione di MySQL.
- Applicazioni che richiedono la disponibilità continua del database. Con un cluster attivo-attivo, i dati vengono conservati su tutte le istanze DB del cluster. Se un'istanza DB fallisce, l'applicazione può reindirizzare il traffico verso un'altra istanza DB del cluster.
- Applicazioni che potrebbero dover suddividere le operazioni di lettura e scrittura tra diverse istanze DB del cluster per scopi di bilanciamento del carico. Con un cluster attivo-attivo, le applicazioni possono inviare traffico di lettura a istanze DB specifiche e scrivere traffico ad altre. Puoi anche cambiare le istanze DB a cui inviare letture o scritture in qualsiasi momento.

Considerazioni e best practice per i cluster attivi-attivi

Prima di utilizzare i cluster active-active di RDS per MySQL, esamina le seguenti considerazioni e best practice:

- I cluster attivi-attivi non possono avere più di nove istanze DB.
- Con il plug-in Group Replication, puoi controllare le garanzie di coerenza delle transazioni del cluster active-active. Per ulteriori informazioni, consulta Transaction Consistency Guarantees nella documentazione di MySQL.
- I conflitti sono possibili quando diverse istanze DB aggiornano la stessa riga in un cluster attivo-attivo. Per informazioni sui conflitti e sulla risoluzione dei conflitti, vedere Replica di gruppo nella documentazione di MySQL.
- Per la tolleranza agli errori, includi almeno tre istanze DB nel tuo cluster active-active. È possibile configurare un cluster attivo-attivo con solo una o due istanze DB, ma il cluster non sarà tollerante ai guasti. Per informazioni sulla tolleranza agli errori, vedere Fault-Tolerance nella documentazione di MySQL.
- Quando un'istanza DB si unisce a un cluster active-active esistente e utilizza la stessa versione del motore della versione del motore più bassa del cluster, l'istanza DB si unisce in modalità di lettura-scrittura.
- Quando un'istanza DB si unisce a un cluster active-active esistente e utilizza una versione del motore superiore rispetto alla versione del motore più bassa del cluster, l'istanza DB deve rimanere in modalità di sola lettura.

- Se si abilita la replica di gruppo per un'istanza DB impostando il relativo `rds.group_replication_enabled` parametro su `1` Nel gruppo di parametri DB, ma la replica non è iniziata o non è riuscita, l'istanza DB viene messa in modalità per evitare incongruenze tra i dati. super-read-only Per informazioni sulla super-read-only modalità, consulta la documentazione di MySQL.
- È possibile aggiornare un'istanza DB in un cluster active-active, ma l'istanza DB è di sola lettura fino a quando tutte le altre istanze DB del cluster active-active non vengono aggiornate alla stessa versione del motore o a una versione superiore del motore. Quando si aggiorna un'istanza DB, l'istanza DB si unisce automaticamente allo stesso cluster active-active al termine dell'aggiornamento. Per evitare il passaggio involontario alla modalità di sola lettura per un'istanza DB, disabilita gli aggiornamenti automatici delle versioni secondarie dell'istanza. Per informazioni sull'aggiornamento di un'istanza database MySQL, consulta Aggiornamento del motore di database MySQL.
- È possibile aggiungere un'istanza DB in un'implementazione di istanze DB Multi-AZ a un cluster active-active esistente. È inoltre possibile convertire un'istanza DB Single-AZ in un cluster active-active in un'implementazione di istanze DB Multi-AZ. Se un'istanza DB primaria in una distribuzione Multi-AZ fallisce, l'istanza primaria passa all'istanza di standby. La nuova istanza DB primaria si unisce automaticamente allo stesso cluster dopo il completamento del failover. Per ulteriori informazioni sulle implementazioni di istanze DB Multi-AZ, consulta. Implementazioni dell'istanza database Multi-AZ
- È consigliabile che le istanze DB in un cluster attivo-attivo abbiano intervalli di tempo diversi per le finestre di manutenzione. Questa pratica evita che più istanze DB nel cluster vadano offline per la manutenzione contemporaneamente. Per ulteriori informazioni, consulta Finestra di manutenzione Amazon RDS.
- I cluster attivi-attivi possono utilizzare SSL per le connessioni tra istanze DB. Per configurare le connessioni SSL, imposta i parametri group_replication_recovery_use_ssl e group_replication_ssl_mode. I valori di questi parametri devono corrispondere per tutte le istanze DB nel cluster active-active.

  Attualmente, i cluster attivi-attivi non supportano la verifica dell'autorità di certificazione (CA) per le connessioni tra. Regioni AWS Pertanto, il parametro group_replication_ssl_mode deve essere impostato su (impostazione predefinita) o per i cluster interregionali. `DISABLED` `REQUIRED`
- Un cluster active-active RDS per MySQL viene eseguito in modalità multi-primaria. Il valore predefinito di group_replication_enforce_update_everywhere_checks è e il parametro è statico. `0N` Quando questo parametro è impostato su, le applicazioni non possono effettuare inserimenti in una tabella con vincoli di chiave esterna a cascata. `0N`

- Un cluster active-active RDS for MySQL utilizza lo stack di comunicazione MySQL per la sicurezza della connessione anziché XCOM. Per ulteriori informazioni, vedere Communication Stack for Connection Security Management nella documentazione di MySQL.
- Quando un gruppo di parametri DB è associato a un'istanza DB in un cluster attivo-attivo, si consiglia di associare questo gruppo di parametri DB solo ad altre istanze DB presenti nel cluster.
- I cluster active-active supportano solo RDS per istanze DB MySQL. Queste istanze DB devono eseguire versioni supportate del motore DB.
- Quando un'istanza DB in un cluster attivo-attivo presenta un errore imprevisto, RDS avvia automaticamente il ripristino dell'istanza DB. Se l'istanza DB non viene ripristinata, consigliamo di sostituirla con una nuova istanza DB eseguendo un point-in-time ripristino con un'istanza DB sana nel cluster. Per istruzioni, consulta Aggiungere un'istanza DB a un cluster attivo-attivo utilizzando il ripristino point-in-time .
- È possibile eliminare un'istanza DB in un cluster attivo-attivo senza influire sulle altre istanze DB del cluster. Per informazioni sulla creazione di un'istanza database, consulta Eliminazione di un'istanza database.

Prerequisiti per un cluster active-active cross-VPC

Puoi configurare un cluster attivo-attivo con istanze DB in più di un VPC. I VPC possono essere uguali o diversi. Regione AWS Regioni AWS

> **Note**
>
> L'invio di traffico tra più utenti Regioni AWS potrebbe comportare costi aggiuntivi. Per ulteriori informazioni, vedere Panoramica dei costi di trasferimento dei dati per architetture comuni.

Se stai configurando un cluster attivo-attivo in un singolo VPC, puoi saltare questi passaggi e passare a. Configurazione di un cluster attivo-attivo con nuove istanze DB

Per prepararsi a un cluster attivo-attivo con istanze DB in più di un VPC

1. Assicurati che gli intervalli di indirizzi IPv4 nei blocchi CIDR soddisfino i seguenti requisiti:
   - Gli intervalli di indirizzi IPv4 nei blocchi CIDR dei VPC non possono sovrapporsi.
   - *Tutti gli intervalli di indirizzi IPv4 nei blocchi CIDR devono essere inferiori o superiori a 128.0.0.0/ subnet_mask. 128.0.0.0/subnet_mask*

I seguenti intervalli illustrano questi requisiti:

- `10.1.0.0/16`in un VPC e `10.2.0.0/16` nell'altro VPC è supportato.
- `172.1.0.0/16`in un VPC e `172.2.0.0/16` nell'altro VPC è supportato.
- `10.1.0.0/16`in un VPC e `10.1.0.0/16` nell'altro VPC non è supportato perché gli intervalli si sovrappongono.
- `10.1.0.0/16`in un VPC e `172.1.0.0/16` nell'altro VPC non è supportato perché uno è al di sotto `128.0.0.0/`*`subnet_mask`* e l'altro è al di sopra. `128.0.0.0/`*`subnet_mask`*

Per informazioni sui blocchi CIDR, consulta i blocchi CIDR VPC nella Amazon VPC User Guide.

2. In ogni VPC, assicurati che la risoluzione DNS e i nomi host DNS siano entrambi abilitati.

   Per istruzioni, consulta Visualizza e aggiorna gli attributi DNS per il tuo VPC nella Amazon VPC User Guide.

3. Configura i VPC in modo da poter instradare il traffico tra di loro in uno dei seguenti modi:

   - Crea una connessione peering VPC tra i VPC.

     Per istruzioni, consulta Creare una connessione peering VPC nella Amazon VPC Peering Guide. In ogni VPC, assicurati che ci siano regole in entrata per i tuoi gruppi di sicurezza che fanno riferimento ai gruppi di sicurezza nel VPC peerizzato. In questo modo, si consente il traffico verso e da istanze associate al gruppo di sicurezza a cui si fa riferimento nel VPC collegato in peering. Per istruzioni, consulta Aggiorna i tuoi gruppi di sicurezza per fare riferimento ai gruppi di sicurezza peer nella Amazon VPC Peering Guide.

   - Crea un gateway di transito tra i VPC.

     Per istruzioni, consulta Guida introduttiva ai gateway di transito in Amazon VPC Transit Gateways. In ogni VPC, assicurati che ci siano regole in entrata per i tuoi gruppi di sicurezza che consentano il traffico proveniente dall'altro VPC, ad esempio regole in entrata che specificano il CIDR dell'altro VPC. In questo modo si consente al traffico di fluire da e verso le istanze associate al gruppo di sicurezza di riferimento nel cluster active-active. Per ulteriori informazioni, consulta Controlla il traffico verso AWS le tue risorse utilizzando i gruppi di sicurezza nella Amazon VPC User Guide.

Impostazioni dei parametri obbligatorie per i cluster attivi-attivi

Le seguenti impostazioni dei parametri sono necessarie quando si configura un cluster active-active RDS per MySQL.

Parametro	Descrizione	Impostazione richiesta
`binlog_format`	Imposta il formato di registrazione binario. Il valore predefinito per RDS for MySQL è. `MIXED` Per ulteriori informazioni, consulta la documentazione di MySQL.	`ROW`
`enforce_gtid_consistency`	Applica la coerenza GTID per l'esecuzione delle istruzioni. Il valore predefinito per RDS for MySQL è. `OFF` Per ulteriori informazioni, consulta la documentazione di MySQL.	`ON`
`group_replication_group_name`	Imposta il nome della replica di gruppo su un UUID. Il formato UUID è. `11111111-2222-3333-4444-555555555555` È possibile generare un UUID MySQL connettendosi a un'istanza DB MySQL ed eseguendo lo. `SELECT UUID()` Il valore deve essere lo stesso per tutte le istanze DB del cluster active-active. Per ulteriori informazioni, consulta la documentazione di MySQL.	Un UUID MySQL
`gtid-mode`	Controlla la registrazione basata su GTID. Il valore	`ON`

Parametro	Descrizione	Impostazione richiesta
	predefinito per RDS for MySQL è. `OFF_PERMISSIVE` Per ulteriori informazioni, consulta la documentazione di MySQL.	
`rds.custom_dns_resolution`	Specifica se consentire la risoluzione DNS dal server Amazon DNS nel tuo VPC. La risoluzione DNS deve essere abilitata quando la replica di gruppo è abilitata con il parametro. `rds.group_replication_enabled` La risoluzione DNS non può essere abilitata quando la replica di gruppo è disabilitata con il parametro. `rds.group_replication_enabled` Per ulteriori informazioni, consulta il server Amazon DNS nella Amazon VPC User Guide.	1
`rds.group_replication_enabled`	Speciifica se la replica di gruppo è abilitata per un'istanza DB. La replica di gruppo deve essere abilitata su un'istanza DB in un cluster attivo-attivo.	1

Parametro	Descrizione	Impostazione richiesta
`slave_preserve_commit_order`	Controlla l'ordine in cui le transazioni vengono eseguite su una replica. Il valore predefinito per RDS for MySQL è. `ON` Per ulteriori informazioni, consulta la documentazione di MySQL.	`ON`

Conversione di un'istanza DB esistente in un cluster attivo-attivo

La versione del motore DB dell'istanza DB che si desidera migrare in un cluster active-active deve essere MySQL 8.0.35 o superiore. Se è necessario aggiornare la versione del motore, vedere. Aggiornamento del motore di database MySQL

Se stai configurando un cluster attivo-attivo con istanze DB in più di un VPC, assicurati di completare i prerequisiti in. Prerequisiti per un cluster active-active cross-VPC

Completa i seguenti passaggi per migrare un'istanza DB esistente in un cluster active-active per RDS for MySQL.

Argomenti

- Passaggio 1: imposta i parametri del cluster active-active in uno o più gruppi di parametri personalizzati
- Fase 2: Associare l'istanza DB a un gruppo di parametri DB con i parametri di replica di gruppo richiesti impostati
- Fase 3: Creare il cluster active-active
- Fase 4: Creare istanze database RDS per MySQL aggiuntive per il cluster active-active
- Passaggio 5: inizializza il gruppo sull'istanza DB che stai convertendo
- Passaggio 6: avviare la replica sulle altre istanze DB nel cluster active-active
- Passaggio 7: (consigliato) Verifica dello stato del cluster attivo-attivo

Passaggio 1: imposta i parametri del cluster active-active in uno o più gruppi di parametri personalizzati

Le istanze DB RDS for MySQL in un cluster active-active devono essere associate a un gruppo di parametri personalizzato con l'impostazione corretta per i parametri richiesti. Per informazioni sui parametri e sull'impostazione richiesta per ciascuno di essi, vedere. Impostazioni dei parametri obbligatorie per i cluster attivi-attivi

È possibile impostare questi parametri in nuovi gruppi di parametri o in gruppi di parametri esistenti. Tuttavia, per evitare di influire accidentalmente sulle istanze DB che non fanno parte del cluster active-active, ti consigliamo vivamente di creare un nuovo gruppo di parametri personalizzato. Le istanze DB in un cluster attivo-attivo possono essere associate allo stesso gruppo di parametri DB o a gruppi di parametri DB diversi.

È possibile utilizzare AWS Management Console o the AWS CLI per creare un nuovo gruppo di parametri personalizzato. Per ulteriori informazioni, consulta Creazione di un gruppo di parametri del database. L'esempio seguente esegue il create-db-parameter-groupAWS CLIcomando per creare un gruppo di parametri DB personalizzato denominato*myactivepg*:

Per LinuxmacOS, oUnix:

```
aws rds create-db-parameter-group \
  --db-parameter-group-name myactivepg \
  --db-parameter-group-family mysql8.0 \
  --description "Parameter group for active-active clusters"
```

Per Windows:

```
aws rds create-db-parameter-group ^
  --db-parameter-group-name myactivepg ^
  --db-parameter-group-family mysql8.0 ^
  --description "Parameter group for active-active clusters"
```

È inoltre possibile utilizzare il AWS Management Console o AWS CLI per impostare i parametri nel gruppo di parametri personalizzato. Per ulteriori informazioni, consulta Modifica di parametri in un gruppo di parametri del database.

L'esempio seguente esegue il modify-db-parameter-groupAWS CLIcomando per impostare i parametri:

Per LinuxmacOS, oUnix:

```
aws rds modify-db-parameter-group \
  --db-parameter-group-name myactivepg \
  --parameters
 "ParameterName='rds.group_replication_enabled',ParameterValue='1',ApplyMethod=pending-
reboot" \

 "ParameterName='rds.custom_dns_resolution',ParameterValue='1',ApplyMethod=pending-
reboot" \

 "ParameterName='enforce_gtid_consistency',ParameterValue='ON',ApplyMethod=pending-
reboot" \
               "ParameterName='gtid-mode',ParameterValue='ON',ApplyMethod=pending-
reboot" \

 "ParameterName='binlog_format',ParameterValue='ROW',ApplyMethod=immediate" \

 "ParameterName='slave_preserve_commit_order',ParameterValue='ON',ApplyMethod=immediate"
 \

 "ParameterName='group_replication_group_name',ParameterValue='11111111-2222-3333-4444-55555555
reboot"
```

Per Windows:

```
aws rds modify-db-parameter-group ^
  --db-parameter-group-name myactivepg ^
  --parameters
 "ParameterName='rds.group_replication_enabled',ParameterValue='1',ApplyMethod=pending-
reboot" ^

 "ParameterName='rds.custom_dns_resolution',ParameterValue='1',ApplyMethod=pending-
reboot" ^

 "ParameterName='enforce_gtid_consistency',ParameterValue='ON',ApplyMethod=pending-
reboot" ^
               "ParameterName='gtid-mode',ParameterValue='ON',ApplyMethod=pending-
reboot" ^

 "ParameterName='binlog_format',ParameterValue='ROW',ApplyMethod=immediate" ^
```

```
 "ParameterName='slave_preserve_commit_order',ParameterValue='ON',ApplyMethod=immediate"
 ^

 "ParameterName='group_replication_group_name',ParameterValue='11111111-2222-3333-4444-55555555
reboot"
```

Fase 2: Associare l'istanza DB a un gruppo di parametri DB con i parametri di replica di gruppo richiesti impostati

Associate l'istanza DB a un gruppo di parametri creato o modificato nel passaggio precedente. Per istruzioni, consulta Associazione di un gruppo di parametri database a un'istanza database.

Riavvia l'istanza DB per rendere effettive le nuove impostazioni dei parametri. Per istruzioni, consulta Riavvio di un'istanza database.

Fase 3: Creare il cluster active-active

Nel gruppo di parametri DB associato all'istanza DB, imposta il `group_replication_group_seeds` parametro sull'endpoint dell'istanza DB che stai convertendo.

È possibile utilizzare AWS Management Console o the AWS CLI per impostare il parametro. Non è necessario riavviare l'istanza DB dopo aver impostato questo parametro. Per informazioni sull'estensione dei parametri consulta Modifica di parametri in un gruppo di parametri del database.

L'esempio seguente esegue il modify-db-parameter-groupAWS CLIcomando per impostare i parametri:

Per LinuxmacOS, oUnix:

```
aws rds modify-db-parameter-group \
  --db-parameter-group-name myactivepg \
  --parameters
 "ParameterName='group_replication_group_seeds',ParameterValue='myactivedb1.123456789012.us-
east-1.rds.amazonaws.com:3306',ApplyMethod=immediate"
```

Per Windows:

```
aws rds modify-db-parameter-group ^
  --db-parameter-group-name myactivepg ^
```

```
  --parameters
 "ParameterName='group_replication_group_seeds',ParameterValue='myactivedb1.123456789012.us-
east-1.rds.amazonaws.com:3306',ApplyMethod=immediate"
```

Fase 4: Creare istanze database RDS per MySQL aggiuntive per il cluster active-active

Per creare istanze DB aggiuntive per il cluster active-active, point-in-time esegui il ripristino sull'istanza DB che stai convertendo. Per istruzioni, consulta Aggiungere un'istanza DB a un cluster attivo-attivo utilizzando il ripristino point-in-time .

Un cluster attivo-attivo può avere fino a nove istanze DB. Esegui il point-in-time ripristino sull'istanza DB fino a ottenere il numero di istanze DB che desideri per il cluster. Quando esegui point-in-recovery, assicurati di associare l'istanza DB che stai aggiungendo a un gruppo di parametri DB `rds.group_replication_enabled` impostato 1 su. In caso contrario, la replica di gruppo non verrà avviata sull'istanza DB appena aggiunta.

Passaggio 5: inizializza il gruppo sull'istanza DB che stai convertendo

Inizializza il gruppo e avvia la replica:

1. Connect all'istanza DB che stai convertendo in un client SQL. Per ulteriori informazioni sulla connessione a un'istanza DB RDS for MySQL, vedere. Connessione a un'istanza database che esegue il motore di database di MySQL
2. Nel client SQL, esegui le seguenti stored procedure e sostituisci *group_replication_user_password* con la password dell'utente. `rdsgrprepladmin` L'`rdsgrprepladmin`utente è riservato alle connessioni di replica di gruppo in un cluster attivo-attivo. La password per questo utente deve essere la stessa su tutte le istanze DB di un cluster active-active.

   ```
   call mysql.rds_set_configuration('binlog retention hours', 168); -- 7 days binlog
   call mysql.rds_group_replication_create_user('group_replication_user_password');
   call
    mysql.rds_group_replication_set_recovery_channel('group_replication_user_password');
   call mysql.rds_group_replication_start(1);
   ```

   Questo esempio imposta il `binlog retention hours` valore su168, il che significa che i file di log binari vengono conservati per sette giorni sull'istanza DB. È possibile modificare questo valore in base alle proprie esigenze.

Questo esempio specifica 1 nella `mysql.rds_group_replication_start` stored procedure l'inizializzazione di un nuovo gruppo con l'istanza DB corrente.

Per ulteriori informazioni sulle stored procedure richiamate nell'esempio, vedere. Gestione di cluster attivi-attivi

Passaggio 6: avviare la replica sulle altre istanze DB nel cluster active-active

Per ciascuna istanza DB del cluster active-active, utilizza un client SQL per connetterti all'istanza ed esegui le seguenti stored procedure. Sostituisci *`group_replication_user_password con la password dell'utente`*. `rdsgrprepladmin`

```
call mysql.rds_set_configuration('binlog retention hours', 168); -- 7 days binlog
call mysql.rds_group_replication_create_user('group_replication_user_password');
call
 mysql.rds_group_replication_set_recovery_channel('group_replication_user_password');
call mysql.rds_group_replication_start(0);
```

Questo esempio imposta il `binlog retention hours` valore su168, il che significa che i file di log binari vengono conservati per sette giorni su ogni istanza DB. È possibile modificare questo valore in base alle proprie esigenze.

Questo esempio specifica `0` nella `mysql.rds_group_replication_start` stored procedure di aggiungere l'istanza DB corrente a un gruppo esistente.

> **Tip**
>
> Assicurati di eseguire queste stored procedure su tutte le altre istanze DB del cluster active-active.

Passaggio 7: (consigliato) Verifica dello stato del cluster attivo-attivo

Per assicurarti che ogni membro del cluster sia configurato correttamente, controlla lo stato del cluster connettendoti a un'istanza DB nel cluster active-active ed eseguendo il seguente comando SQL:

```
SELECT * FROM performance_schema.replication_group_members;
```

L'output dovrebbe essere visualizzato `ONLINE` per ogni istanza DB, come nell'output di esempio seguente: `MEMBER_STATE`

```
+---------------------------+--------------------------------------
+---------------+-------------+--------------+-------------+---------------
+----------------------------+
| CHANNEL_NAME              | MEMBER_ID                            | MEMBER_HOST    |
 MEMBER_PORT | MEMBER_STATE | MEMBER_ROLE | MEMBER_VERSION | MEMBER_COMMUNICATION_STACK
 |
+---------------------------+--------------------------------------
+---------------+-------------+--------------+-------------+---------------
+----------------------------+
| group_replication_applier | 9854d4a2-5d7f-11ee-b8ec-0ec88c43c251 | ip-10-15-3-137 |
      3306 | ONLINE       | PRIMARY     | 8.0.35         | MySQL                       |
| group_replication_applier | 9e2e9c28-5d7f-11ee-8039-0e5d58f05fef | ip-10-15-3-225 |
      3306 | ONLINE       | PRIMARY     | 8.0.35         | MySQL                       |
| group_replication_applier | a6ba332d-5d7f-11ee-a025-0a5c6971197d | ip-10-15-1-83  |
      3306 | ONLINE       | PRIMARY     | 8.0.35         | MySQL                       |
+---------------------------+--------------------------------------
+---------------+-------------+--------------+-------------+---------------
+----------------------------+
3 rows in set (0.00 sec)
```

Per informazioni sui `MEMBER_STATE` valori possibili, vedere Group Replication Server States nella documentazione di MySQL.

Configurazione di un cluster attivo-attivo con nuove istanze DB

Completa i seguenti passaggi per configurare un cluster attivo-attivo utilizzando nuove istanze DB RDS per MySQL.

Se stai configurando un cluster attivo-attivo con istanze DB in più di un VPC, assicurati di completare i prerequisiti in. Prerequisiti per un cluster active-active cross-VPC

Argomenti

- Passaggio 1: imposta i parametri del cluster attivo-attivo in uno o più gruppi di parametri personalizzati
- Fase 2: Creare nuove istanze DB RDS per MySQL per il cluster active-active
- Fase 4: Specificare le istanze DB nel cluster attivo-attivo
- Passaggio 5: inizializza il gruppo su un'istanza DB e avvia la replica

- Passaggio 6: avviare la replica sulle altre istanze DB nel cluster active-active
- Passaggio 7: (consigliato) Verifica dello stato del cluster attivo-attivo
- Fase 8: (Facoltativo) Importazione dei dati in un'istanza DB nel cluster active-active

Passaggio 1: imposta i parametri del cluster attivo-attivo in uno o più gruppi di parametri personalizzati

Le istanze DB RDS for MySQL in un cluster active-active devono essere associate a un gruppo di parametri personalizzato con l'impostazione corretta per i parametri richiesti. Per informazioni sui parametri e sull'impostazione richiesta per ciascuno di essi, vedere. Impostazioni dei parametri obbligatorie per i cluster attivi-attivi

È possibile impostare questi parametri in nuovi gruppi di parametri o in gruppi di parametri esistenti. Tuttavia, per evitare di influire accidentalmente sulle istanze DB che non fanno parte del cluster active-active, ti consigliamo vivamente di creare un nuovo gruppo di parametri personalizzato. Le istanze DB in un cluster attivo-attivo possono essere associate allo stesso gruppo di parametri DB o a gruppi di parametri DB diversi.

È possibile utilizzare AWS Management Console o the AWS CLI per creare un nuovo gruppo di parametri personalizzato. Per ulteriori informazioni, consulta Creazione di un gruppo di parametri del database. L'esempio seguente esegue il create-db-parameter-groupAWS CLIcomando per creare un gruppo di parametri DB personalizzato denominato*myactivepg*:

Per LinuxmacOS, oUnix:

```
aws rds create-db-parameter-group \
  --db-parameter-group-name myactivepg \
  --db-parameter-group-family mysql8.0 \
  --description "Parameter group for active-active clusters"
```

Per Windows:

```
aws rds create-db-parameter-group ^
  --db-parameter-group-name myactivepg ^
  --db-parameter-group-family mysql8.0 ^
  --description "Parameter group for active-active clusters"
```

È inoltre possibile utilizzare il AWS Management Console o AWS CLI per impostare i parametri nel gruppo di parametri personalizzato. Per ulteriori informazioni, consulta Modifica di parametri in un gruppo di parametri del database.

L'esempio seguente esegue il modify-db-parameter-groupAWS CLIcomando per impostare i parametri:

Per LinuxmacOS, oUnix:

```
aws rds modify-db-parameter-group \
  --db-parameter-group-name myactivepg \
  --parameters
 "ParameterName='rds.group_replication_enabled',ParameterValue='1',ApplyMethod=pending-
reboot" \

 "ParameterName='rds.custom_dns_resolution',ParameterValue='1',ApplyMethod=pending-
reboot" \

 "ParameterName='enforce_gtid_consistency',ParameterValue='ON',ApplyMethod=pending-
reboot" \
              "ParameterName='gtid-mode',ParameterValue='ON',ApplyMethod=pending-
reboot" \

 "ParameterName='binlog_format',ParameterValue='ROW',ApplyMethod=immediate" \

 "ParameterName='slave_preserve_commit_order',ParameterValue='ON',ApplyMethod=immediate"
 \

 "ParameterName='group_replication_group_name',ParameterValue='11111111-2222-3333-4444-55555555
reboot"
```

Per Windows:

```
aws rds modify-db-parameter-group ^
  --db-parameter-group-name myactivepg ^
  --parameters
 "ParameterName='rds.group_replication_enabled',ParameterValue='1',ApplyMethod=pending-
reboot" ^

 "ParameterName='rds.custom_dns_resolution',ParameterValue='1',ApplyMethod=pending-
reboot" ^
```

```
 "ParameterName='enforce_gtid_consistency',ParameterValue='ON',ApplyMethod=pending-
reboot" ^
               "ParameterName='gtid-mode',ParameterValue='ON',ApplyMethod=pending-
reboot" ^

 "ParameterName='binlog_format',ParameterValue='ROW',ApplyMethod=immediate" ^

 "ParameterName='slave_preserve_commit_order',ParameterValue='ON',ApplyMethod=immediate"
 ^

 "ParameterName='group_replication_group_name',ParameterValue='11111111-2222-3333-4444-55555555
reboot"
```

Fase 2: Creare nuove istanze DB RDS per MySQL per il cluster active-active

I cluster active-active sono supportati per la versione 8.0.35 e successive di RDS per le istanze database MySQL. È possibile creare fino a nove nuove istanze DB per il cluster.

È possibile utilizzare AWS Management Console o the AWS CLI per creare nuove istanze DB. Per ulteriori informazioni sulla creazione di un'istanza database, consulta Creazione di un'istanza database Amazon RDS. Quando create l'istanza DB, associatela a un gruppo di parametri DB creato o modificato nel passaggio precedente.

Fase 4: Specificare le istanze DB nel cluster attivo-attivo

Nel gruppo di parametri DB associato a ciascuna istanza DB, imposta il `group_replication_group_seeds` parametro sugli endpoint delle istanze DB che desideri includere nel cluster.

È possibile utilizzare AWS Management Console o the AWS CLI per impostare il parametro. Non è necessario riavviare l'istanza DB dopo aver impostato questo parametro. Per informazioni sull'estensione dei parametri consulta Modifica di parametri in un gruppo di parametri del database.

L'esempio seguente esegue il modify-db-parameter-groupAWS CLIcomando per impostare i parametri:

Per LinuxmacOS, oUnix:

```
aws rds modify-db-parameter-group \
```

```
  --db-parameter-group-name myactivepg \
  --parameters
 "ParameterName='group_replication_group_seeds',ParameterValue='myactivedb1.123456789012.us-
east-1.rds.amazonaws.com:3306,myactivedb2.123456789012.us-
east-1.rds.amazonaws.com:3306,myactivedb3.123456789012.us-
east-1.rds.amazonaws.com:3306',ApplyMethod=immediate"
```

Per Windows:

```
aws rds modify-db-parameter-group ^
  --db-parameter-group-name myactivepg ^
  --parameters
 "ParameterName='group_replication_group_seeds',ParameterValue='myactivedb1.123456789012.us-
east-1.rds.amazonaws.com:3306,myactivedb2.123456789012.us-
east-1.rds.amazonaws.com:3306,myactivedb3.123456789012.us-
east-1.rds.amazonaws.com:3306',ApplyMethod=immediate"
```

> **Tip**
>
> Assicurati di impostare il `group_replication_group_seeds` parametro in ogni gruppo di parametri DB associato a un'istanza DB nel cluster active-active.

Passaggio 5: inizializza il gruppo su un'istanza DB e avvia la replica

È possibile scegliere qualsiasi nuovo DB per inizializzare il gruppo e avviare la replica. Per fare ciò, completa la seguente procedura:

1. Scegli un'istanza DB nel cluster active-active e connettiti a quell'istanza DB in un client SQL. Per ulteriori informazioni sulla connessione a un'istanza DB RDS for MySQL, vedere. Connessione a un'istanza database che esegue il motore di database di MySQL
2. Nel client SQL, esegui le seguenti stored procedure e sostituisci *group_replication_user_password* con la password dell'utente. `rdsgrprepladmin` L'`rdsgrprepladmin`utente è riservato alle connessioni di replica di gruppo in un cluster attivo-attivo. La password per questo utente deve essere la stessa su tutte le istanze DB di un cluster active-active.

   ```
   call mysql.rds_set_configuration('binlog retention hours', 168); -- 7 days binlog
   call mysql.rds_group_replication_create_user('group_replication_user_password');
   ```

```
call
 mysql.rds_group_replication_set_recovery_channel('group_replication_user_password');
call mysql.rds_group_replication_start(1);
```

Questo esempio imposta il `binlog retention hours` valore su168, il che significa che i file di log binari vengono conservati per sette giorni sull'istanza DB. È possibile modificare questo valore in base alle proprie esigenze.

Questo esempio specifica 1 nella `mysql.rds_group_replication_start` stored procedure l'inizializzazione di un nuovo gruppo con l'istanza DB corrente.

Per ulteriori informazioni sulle stored procedure richiamate nell'esempio, vedere. Gestione di cluster attivi-attivi

Passaggio 6: avviare la replica sulle altre istanze DB nel cluster active-active

Per ciascuna istanza DB del cluster active-active, utilizza un client SQL per connetterti all'istanza ed esegui le seguenti stored procedure. Sostituisci *`group_replication_user_password con la password dell'utente`*. `rdsgrprepladmin`

```
call mysql.rds_set_configuration('binlog retention hours', 168); -- 7 days binlog
call mysql.rds_group_replication_create_user('group_replication_user_password');
call
 mysql.rds_group_replication_set_recovery_channel('group_replication_user_password');
call mysql.rds_group_replication_start(0);
```

Questo esempio imposta il `binlog retention hours` valore su168, il che significa che i file di log binari vengono conservati per sette giorni su ogni istanza DB. È possibile modificare questo valore in base alle proprie esigenze.

Questo esempio specifica `0` nella `mysql.rds_group_replication_start` stored procedure di aggiungere l'istanza DB corrente a un gruppo esistente.

> **Tip**
>
> Assicurati di eseguire queste stored procedure su tutte le altre istanze DB del cluster active-active.

Passaggio 7: (consigliato) Verifica dello stato del cluster attivo-attivo

Per assicurarti che ogni membro del cluster sia configurato correttamente, controlla lo stato del cluster connettendoti a un'istanza DB nel cluster active-active ed eseguendo il seguente comando SQL:

```
SELECT * FROM performance_schema.replication_group_members;
```

L'output dovrebbe essere visualizzato `ONLINE` per ogni istanza DB, come nell'output di esempio seguente: `MEMBER_STATE`

```
+---------------------------+--------------------------------------
+---------------+-------------+--------------+-------------+---------------
+----------------------------+
| CHANNEL_NAME              | MEMBER_ID                            | MEMBER_HOST    |
 MEMBER_PORT | MEMBER_STATE | MEMBER_ROLE | MEMBER_VERSION | MEMBER_COMMUNICATION_STACK
 |
+---------------------------+--------------------------------------
+---------------+-------------+--------------+-------------+---------------
+----------------------------+
| group_replication_applier | 9854d4a2-5d7f-11ee-b8ec-0ec88c43c251 | ip-10-15-3-137 |
      3306 | ONLINE       | PRIMARY     | 8.0.35         | MySQL                       |
| group_replication_applier | 9e2e9c28-5d7f-11ee-8039-0e5d58f05fef | ip-10-15-3-225 |
      3306 | ONLINE       | PRIMARY     | 8.0.35         | MySQL                       |
| group_replication_applier | a6ba332d-5d7f-11ee-a025-0a5c6971197d | ip-10-15-1-83  |
      3306 | ONLINE       | PRIMARY     | 8.0.35         | MySQL                       |
+---------------------------+--------------------------------------
+---------------+-------------+--------------+-------------+---------------
+----------------------------+
3 rows in set (0.00 sec)
```

Per informazioni sui `MEMBER_STATE` valori possibili, vedere Group Replication Server States nella documentazione di MySQL.

Fase 8: (Facoltativo) Importazione dei dati in un'istanza DB nel cluster active-active

È possibile importare dati da un database MySQL in un'istanza DB nel cluster active-active. Dopo l'importazione dei dati, Group Replication li replica nelle altre istanze DB del cluster.

Per informazioni sull'importazione dei dati, vedere. Importazione dei dati in un database Amazon RDS MariaDB o MySQL con tempi di inattività ridotti

Aggiungere un'istanza DB a un cluster attivo-attivo

È possibile aggiungere un'istanza DB a un cluster attivo-attivo ripristinando uno snapshot DB o ripristinando un'istanza DB in un point-in-time. Un cluster attivo-attivo può includere fino a nove istanze DB.

Quando si ripristina un'istanza DB in un determinato momento, in genere vengono incluse transazioni più recenti rispetto a un'istanza DB ripristinata da uno snapshot DB. Quando l'istanza DB ha transazioni più recenti, è necessario applicare un numero inferiore di transazioni all'avvio della replica. Pertanto, l'utilizzo del point-in-time ripristino per aggiungere un'istanza DB a un cluster è in genere più rapido rispetto al ripristino da un'istantanea del database.

Argomenti

Aggiungere un'istanza DB a un cluster attivo-attivo utilizzando il ripristino point-in-time

È possibile aggiungere un'istanza DB a un cluster attivo-attivo eseguendo point-in-time il ripristino su un'istanza DB del cluster.

Per informazioni sul ripristino di un'istanza DB in un momento temporale diversoRegione AWS, consulta. Replica dei backup automatici su un altro Regione AWS

Per aggiungere un'istanza DB a un cluster attivo-attivo utilizzando il ripristino point-in-time

1. Crea una nuova istanza DB eseguendo point-in-time il ripristino su un'istanza DB nel cluster active-active.

   È possibile eseguire point-in-time il ripristino su qualsiasi istanza DB del cluster per creare la nuova istanza DB. Per istruzioni, consulta Ripristino a un'ora specifica per un'istanza database.

   > **Important**
   >
   > Durante point-in-time-recovery, associa la nuova istanza DB a un gruppo di parametri DB con i parametri del cluster active-active impostati. In caso contrario, la replica di gruppo non verrà avviata sulla nuova istanza DB. Per informazioni sui parametri e sulle

impostazioni richieste per ciascuno di essi, vedereImpostazioni dei parametri obbligatorie per i cluster attivi-attivi.

> **Tip**
>
> Se si scatta uno snapshot dell'istanza DB prima di iniziare point-in-time il ripristino, potrebbe essere possibile ridurre il tempo necessario per applicare le transazioni sulla nuova istanza DB.

2. Aggiungi l'istanza DB al `group_replication_group_seeds` parametro in ogni gruppo di parametri DB associato a un'istanza DB nel cluster active-active, incluso il gruppo di parametri DB associato alla nuova istanza DB.

   Per informazioni sull'estensione dei parametri consulta Modifica di parametri in un gruppo di parametri del database.

3. In un client SQL, connettiti alla nuova istanza DB e richiama la mysql.rds_group_replication_set_recovery_channel stored procedure. Sostituisci *`group_replication_user_password con la password`* dell'utente. `rdsgrprepladmin`

   ```
   call
    mysql.rds_group_replication_set_recovery_channel('group_replication_user_password');
   ```

4. Utilizzando il client SQL, richiamate la stored procedure per avviare la replica: mysql.rds_group_replication_start

   ```
   call mysql.rds_group_replication_start(0);
   ```

Aggiungere un'istanza DB a un cluster attivo-attivo utilizzando un'istantanea DB

È possibile aggiungere un'istanza DB a un cluster attivo attivo creando uno snapshot DB di un'istanza DB nel cluster e quindi ripristinando lo snapshot DB.

Per informazioni sulla copia di uno snapshot su un altro, consulta. Regione AWS the section called “Copia tra regioni”

Per aggiungere un'istanza DB a un cluster attivo-attivo utilizzando uno snapshot DB

1. Crea uno snapshot DB di un'istanza DB nel cluster active-active.

   È possibile creare uno snapshot DB di qualsiasi istanza DB nel cluster. Per istruzioni, consulta Creazione di uno snapshot DB per un'istanza DB Single-AZ.

2. Ripristina un'istanza DB dallo snapshot DB.

   Durante l'operazione di ripristino dello snapshot, associa la nuova istanza DB a un gruppo di parametri DB con i parametri del cluster active-active impostati. Per informazioni sui parametri e sull'impostazione richiesta per ciascuno di essi, vedere. Impostazioni dei parametri obbligatorie per i cluster attivi-attivi

   Per informazioni sul ripristino di un'istanza DB da un'istantanea del database, vedere. Ripristino da uno snapshot database

3. Aggiungi l'istanza DB al `group_replication_group_seeds` parametro in ogni gruppo di parametri DB associato a un'istanza DB nel cluster active-active, incluso il gruppo di parametri DB associato alla nuova istanza DB.

   Per informazioni sull'estensione dei parametri consulta Modifica di parametri in un gruppo di parametri del database.

4. In un client SQL, connettiti alla nuova istanza DB e richiama la mysql.rds_group_replication_set_recovery_channel stored procedure. Sostituisci *`group_replication_user_password con la password`* dell'utente. `rdsgrprepladmin`

   ```
   call
    mysql.rds_group_replication_set_recovery_channel('group_replication_user_password');
   ```

5. Utilizzando il client SQL, richiamate la stored procedure per avviare la replica: mysql.rds_group_replication_start

   ```
   call mysql.rds_group_replication_start(0);
   ```

Monitoraggio dei cluster attivi-attivi

È possibile monitorare il cluster active-active connettendosi a un'istanza DB nel cluster ed eseguendo il seguente comando SQL:

```
SELECT * FROM performance_schema.replication_group_members;
```

L'output dovrebbe essere visualizzato `ONLINE` per ogni istanza DB, come nell'output di esempio seguente: `MEMBER_STATE`

```
+---------------------------+--------------------------------------
+---------------+-------------+-------------+-------------+---------------
+----------------------------+
| CHANNEL_NAME              | MEMBER_ID                            | MEMBER_HOST    |
 MEMBER_PORT | MEMBER_STATE | MEMBER_ROLE | MEMBER_VERSION | MEMBER_COMMUNICATION_STACK
 |
+---------------------------+--------------------------------------
+---------------+-------------+-------------+-------------+---------------
+----------------------------+
| group_replication_applier | 9854d4a2-5d7f-11ee-b8ec-0ec88c43c251 | ip-10-15-3-137 |
      3306 | ONLINE       | PRIMARY     | 8.0.35         | MySQL                       |
| group_replication_applier | 9e2e9c28-5d7f-11ee-8039-0e5d58f05fef | ip-10-15-3-225 |
      3306 | ONLINE       | PRIMARY     | 8.0.35         | MySQL                       |
| group_replication_applier | a6ba332d-5d7f-11ee-a025-0a5c6971197d | ip-10-15-1-83  |
      3306 | ONLINE       | PRIMARY     | 8.0.35         | MySQL                       |
+---------------------------+--------------------------------------
+---------------+-------------+-------------+-------------+---------------
+----------------------------+
3 rows in set (0.00 sec)
```

Per informazioni sui `MEMBER_STATE` valori possibili, vedere Group Replication Server States nella documentazione di MySQL.

Interruzione della replica di gruppo su un'istanza DB in un cluster attivo-attivo

È possibile interrompere la replica di gruppo su un'istanza DB in un cluster attivo-attivo. Quando si interrompe la replica di gruppo, l'istanza DB viene messa in super-read-only modalità fino al riavvio della replica o alla rimozione dell'istanza DB dal cluster active-active. Per informazioni sulla super-read-only modalità, consulta la documentazione di MySQL.

Per interrompere temporaneamente la replica di gruppo per un cluster attivo-attivo

1. Connect a un'istanza DB nel cluster active-active utilizzando un client SQL.

Per ulteriori informazioni sulla connessione a un'istanza DB RDS for MySQL, vedere. Connessione a un'istanza database che esegue il motore di database di MySQL

2. Nel client SQL, richiamate la mysql.rds_group_replication_stop stored procedure:

```
call mysql.rds_group_replication_stop();
```

Rinominare un'istanza DB in un cluster attivo-attivo

È possibile modificare il nome di un'istanza DB in un cluster attivo-attivo. Per rinominare più di un'istanza DB in un cluster attivo-attivo, esegui l'operazione un'istanza DB alla volta. Quindi, rinomina un'istanza DB e ricongiungila al cluster prima di rinominare l'istanza DB successiva.

Per rinominare un'istanza DB in un cluster attivo-attivo

1. Connect all'istanza DB in un client SQL e richiama la mysql.rds_group_replication_stop stored procedure:

```
call mysql.rds_group_replication_stop();
```

2. Rinomina l'istanza DB seguendo le istruzioni riportate inRidenominazione di un'istanza database.
3. Modifica il `group_replication_group_seeds` parametro in ogni gruppo di parametri DB associato a un'istanza DB nel cluster active-active.

   Nell'impostazione dei parametri, sostituisci il vecchio endpoint dell'istanza DB con il nuovo endpoint dell'istanza DB. Per informazioni sull'estensione dei parametri consulta Modifica di parametri in un gruppo di parametri del database.
4. Connect all'istanza DB in un client SQL e richiama la mysql.rds_group_replication_start stored procedure:

```
call mysql.rds_group_replication_start(0);
```

Rimozione di un'istanza DB da un cluster attivo-attivo

Quando si rimuove un'istanza DB da un cluster attivo attivo, viene ripristinata un'istanza DB autonoma.

Per rimuovere un'istanza DB da un cluster attivo-attivo

1. Connect all'istanza DB in un client SQL e richiama la mysql.rds_group_replication_stop stored procedure:

   ```
   call mysql.rds_group_replication_stop();
   ```

2. Modifica il `group_replication_group_seeds` parametro per le istanze DB che rimarranno nel cluster active-active.

   Nel `group_replication_group_seeds` parametro, eliminate l'istanza DB che state rimuovendo dal cluster active-active. Per informazioni sull'estensione dei parametri consulta Modifica di parametri in un gruppo di parametri del database.

3. Modifica i parametri dell'istanza DB che stai rimuovendo dal cluster active-active in modo che non faccia più parte del cluster.

   È possibile associare l'istanza DB a un gruppo di parametri diverso o modificare i parametri nel gruppo di parametri DB associato all'istanza DB. I parametri da modificare includono `group_replication_group_namerds.group_replication_enabled`, e`group_replication_group_seeds`. Per ulteriori informazioni sui parametri del cluster active-active, vedere. Impostazioni dei parametri obbligatorie per i cluster attivi-attivi

   Se modifichi i parametri in un gruppo di parametri DB, assicurati che il gruppo di parametri DB non sia associato ad altre istanze DB nel cluster active-active.

4. Riavvia l'istanza DB che hai rimosso dal cluster active-active per rendere effettive le nuove impostazioni dei parametri.

   Per istruzioni, consulta Riavvio di un'istanza database.

Limitazioni per i cluster active-active di RDS per MySQL

Le seguenti limitazioni si applicano ai cluster active-active per RDS for MySQL:

- Il nome utente principale non può essere utilizzato `rdsgrprepladmin` per le istanze DB in un cluster attivo-attivo. Questo nome utente è riservato alle connessioni di replica di gruppo.
- Per le istanze DB con repliche di lettura in cluster active-active, uno stato di replica prolungato diverso da 1 `Replicating` può far sì che i file di log superino i limiti di archiviazione. Per informazioni sullo stato delle repliche di lettura, vedere. Monitoraggio della replica di lettura

- Le distribuzioni blu/verdi non sono supportate per le istanze DB in un cluster attivo-attivo. Per ulteriori informazioni, consulta Utilizzo delle implementazioni blu/verde Amazon RDS per gli aggiornamenti del database.
- L'autenticazione Kerberos non è supportata per le istanze DB in un cluster attivo-attivo. Per ulteriori informazioni, consulta Utilizzo dell'autenticazione Kerberos per MySQL.
- Le istanze DB in un cluster DB Multi-AZ non possono essere aggiunte a un cluster active-active.

  Tuttavia, le istanze DB in una distribuzione di istanze DB Multi-AZ possono essere aggiunte a un cluster active-active.

  Per ulteriori informazioni, consulta Configurazione e gestione di un'implementazione multi-AZ.
- Le tabelle che non dispongono di una chiave primaria non vengono replicate in un cluster attivo-attivo perché le scritture vengono rifiutate dal plug-in Group Replication.
- Le tabelle non InnoDB non vengono replicate in un cluster attivo-attivo.
- I cluster attivi-attivi non supportano istruzioni DML e DDL simultanee su diverse istanze DB del cluster.
- Non è possibile configurare un cluster attivo-attivo per utilizzare la modalità primaria singola per la modalità di replica del gruppo. Per questa configurazione, consigliamo invece di utilizzare un cluster DB Multi-AZ. Per ulteriori informazioni, consulta Implementazioni cluster di database multi-AZ.
- La replica da più fonti non è supportata per le istanze DB in un cluster active-active.
- Un cluster active-active interregionale non può imporre la verifica dell'autorità di certificazione (CA) per le connessioni di replica di gruppo.

Esportazione di dati da un'istanza database MySQL tramite la replica

È possibile utilizzare la replica per esportare dati da un'istanza database MySQL a un'istanza di MySQL eseguita esternamente ad Amazon RDS. In questo scenario, l'istanza database MySQL è l'istanza database MySQL di origine e l’istanza MySQL in esecuzione esterna a Amazon RDS è il database MySQL esterno.

Il database MySQL esterno può essere eseguito in locale nel data center o su un'istanza Amazon EC2. Il database MySQL esterno deve eseguire la stessa versione dell'istanza database MySQL di origine o una versione successiva.

La replica in un database MySQL esterno è supportata solo durante il tempo necessario per esportare un database dall'istanza database MySQL di origine. La replica deve essere terminata dopo che i dati sono stati esportati e le applicazioni possono iniziare ad accedere all'istanza MySQL esterna.

Nell'elenco seguente è indicata la procedura da eseguire. Ciascuna fase della procedura è descritta in modo dettagliato nelle sezioni successive.

1. Preparare un'istanza database MySQL esterna.
2. Preparare l'istanza database MySQL di origine per la replica.
3. Utilizzare l'utilità mysqldump per trasferire il database dall'istanza database MySQL di origine al database MySQL esterno.
4. Avviare la replica nel database MySQL esterno.
5. Al termine dell'esportazione, arrestare la replica.

Preparare un database MySQL esterno

Eseguire la procedura seguente per preparare il database MySQL esterno.

Per preparare il database MySQL esterno

1. Installare il database MySQL esterno.
2. Connettiti al database MySQL esterno come utente master. Creare quindi gli utenti necessari per supportare gli amministratori, le applicazioni e i servizi che accedono al database.

3. Seguire le istruzioni nella documentazione MySQL per preparare il database MySQL esterno come replica. Per ulteriori informazioni, consulta la documentazione di MySQL.
4. Configurare una regola di uscita per consentire al database MySQL esterno di funzionare come replica di lettura durante l'esportazione. La regola di uscita consente al database MySQL esterno di connettersi all'istanza database MySQL di origine durante la replica. Specificare una regola in uscita che consenta le connessioni TCP (Transmission Control Protocol) alla porta e all'indirizzo IP dell'istanza database MySQL di origine.

   Specificare le regole di uscita appropriate per l'ambiente in uso:

   - Se il database MySQL esterno è in esecuzione in un'istanza Amazon EC2 in un Virtual Private Cloud (VPC) basato sul servizio Amazon VPC, specificare le regole di uscita in un gruppo di sicurezza VPC. Per ulteriori informazioni, consulta Controllo dell'accesso con i gruppi di sicurezza.
   - Se il database MySQL esterno è installato in locale, specificare le regole di uscita in un firewall.
5. Se il database MySQL esterno è in esecuzione in un VPC, configurare le regole per le regole dell'elenco di controllo accessi VPC (ACL) oltre alla regola di uscita del gruppo di sicurezza:

   - Configurare una regola in ingresso della lista di controllo accessi che abiliti il traffico TCP verso le porte 1024–65535 dall'indirizzo IP dell'istanza database MySQL di origine.
   - Configurare una regola in uscita della lista di controllo accessi che abiliti il traffico TCP in uscita verso la porta e l'indirizzo IP dell'istanza database MySQL di origine.

   Per ulteriori informazioni sulle liste di controllo accessi di rete Amazon VPC, consulta Liste di controllo accessi di rete in Guida per l'utente di Amazon VPC.
6. (Facoltativo) Si consiglia di impostare il parametro `max_allowed_packet` sulla dimensione massima per evitare errori di replica. Si consiglia questa impostazione.

Preparare l'istanza database MySQL di origine

Eseguire la procedura seguente per preparare l'istanza database MySQL di origine come origine di replica.

Per preparare l'istanza database MySQL di origine

1. Assicurarsi che il computer client disponga di spazio su disco sufficiente per salvare i log binari durante la configurazione della replica.
2. Connettersi all'istanza database MySQL di origine e creare un account di replica seguendo le istruzioni in Creazione di un utente per la replica nella documentazione MySQL.
3. Configurare le regole di ingresso sul sistema che esegue l'istanza database MySQL di origine per consentire al database MySQL esterno di connettersi durante la replica. Specificare una regola di ingresso che abiliti connessioni TCP alla porta usata dall'istanza database MySQL di origine dall'indirizzo IP del database MySQL esterno.
4. Specificare le regole di uscita:

   - Se l'istanza database viene eseguita in un VPC, specificare le regole in ingresso in un gruppo di sicurezza VPC. Per ulteriori informazioni, consulta Controllo dell'accesso con i gruppi di sicurezza.
5. Se l'istanza database viene eseguita in un VPC, configurare le regole della lista di controllo accessi VPC oltre alla regola di ingresso del gruppo di sicurezza.

   - Configurare una regola di ingresso della lista di controllo accessi che abiliti le connessioni TCP alla porta utilizzata dall'istanza Amazon RDS dall'indirizzo IP del database MySQL esterno.
   - Configurare una regola di ingresso che abiliti le connessioni TCP dalle porte 1024–65535 all'indirizzo IP del database MySQL esterno.

   Per ulteriori informazioni sulle liste di controllo accessi di rete Amazon VPC, consulta Liste di controllo accessi di rete nella Guida per l'utente di Amazon VPC.
6. È necessario assicurarsi che la durata del periodo di retention dei backup impostata sia sufficiente a garantire che nessun log binario sia eliminato durante l'esportazione. Se eventuali log vengono eliminati prima che l'esportazione sia completata, dovrai riavviare la replica dall'inizio. Per ulteriori informazioni su come impostare il periodo di retention dei backup, consulta Introduzione ai backup.
7. Utilizzare la stored procedure `mysql.rds_set_configuration` per impostare una durata del periodo di retention dei log binari sufficiente a garantire che i log primari non vengano eliminati durante l'esportazione. Per ulteriori informazioni, consulta Accesso ai log binari MySQL.
8. Per garantire ulteriormente che i log binari dell'istanza database MySQL di origine non vengano eliminati, è necessario creare una replica di lettura Amazon RDS dall'istanza database MySQL di origine. Per ulteriori informazioni, consulta Creazione di una replica di lettura.

9. Dopo che la replica di lettura Amazon RDS è stata creata, chiamare la stored procedure `mysql.rds_stop_replication` per arrestare il processo di replica. L'istanza database MySQL di origine non elimina più i file di log binari, quindi questi sono disponibili per il processo di replica.
10. (Facoltativo) Impostare il parametro `max_allowed_packet` e il parametro `slave_max_allowed_packet` sulla dimensione massima per evitare errori di replica. La dimensione massima per entrambi i parametri è 1 GB. Si consiglia questa impostazione per entrambi i parametri. Per informazioni sull'estensione dei parametri consulta Modifica di parametri in un gruppo di parametri del database.

Copia del database

Eseguire la procedura seguente per copiare il database.

Per copiare il database

1. Connettersi alla replica di lettura RDS dell'istanza database MySQL di origine ed eseguire l'istruzione `SHOW REPLICA STATUS\G` MySQL. Prendere nota dei valori per i seguenti elementi:
   - `Master_Host`
   - `Master_Port`
   - `Master_Log_File`
   - `Exec_Master_Log_Pos`

   > **Note**
   >
   > Versioni precedenti di MySQL utilizzate `SHOW SLAVE STATUS` al posto di `SHOW REPLICA STATUS`. Se si utilizza una versione MySQL prima della 8.0.23, utilizzare `SHOW SLAVE STATUS`.

2. Usa l'utilità mysqldump per creare uno snapshot, che copia i dati da Amazon RDS al computer client locale. Assicurarsi che il computer client disponga di spazio sufficiente per contenere i file `mysqldump` dei database da replicare. Questo processo può richiedere diverse ore in caso di database di grandi dimensioni. Seguire le istruzioni in Creazione di uno snapshot di dati utilizzando mysqldump nella documentazione MySQL.

Nell'esempio seguente viene eseguito `mysqldump` su un client e viene scritto il dump in un file.

Per LinuxmacOS, oUnix:

```
mysqldump -h source_MySQL_DB_instance_endpoint \
    -u user \
    -ppassword \
    --port=3306 \
    --single-transaction \
    --routines \
    --triggers \
    --databases  database database2 > path/rds-dump.sql
```

Per Windows:

```
mysqldump -h source_MySQL_DB_instance_endpoint ^
    -u user ^
    -ppassword ^
    --port=3306 ^
    --single-transaction ^
    --routines ^
    --triggers ^
    --databases  database database2 > path\rds-dump.sql
```

È possibile caricare il file di backup nel database MySQL esterno. Per ulteriori informazioni, consulta Reloading SQL-Format Backups (Ricaricamento dei backup in formato SQL) nella documentazione di MySQL. Puoi eseguire un'altra utilità per caricare i dati nel database MySQL esterno.

Completamento dell'esportazione

Per completare l'esportazione, attenersi alla seguente procedura.

Per completare l'esportazione

1. Utilizzare l'istruzione MySQL `CHANGE MASTER` per configurare il database MySQL esterno. Specificare l'ID e la password delle autorizzazioni `REPLICATION SLAVE` concesse all'utente. Specificare i valori di `Master_Host`, `Master_Port`, `Relay_Master_Log_File`e

`Exec_Master_Log_Pos` ottenuti dall'istruzione `SHOW REPLICA STATUS\G` MySQL eseguita sulla replica di lettura RDS. Per ulteriori informazioni, consulta la documentazione di MySQL.

> **Note**
>
> Versioni precedenti di MySQL utilizzate `SHOW SLAVE STATUS` al posto di `SHOW REPLICA STATUS`. Se si utilizza una versione MySQL prima della 8.0.23, utilizzare `SHOW SLAVE STATUS`.

2. Utilizzare il comando `START REPLICA` MySQL per avviare la replica dall'istanza database MySQL di origine al database MySQL esterno.

   In questo modo viene avviata la replica dall'istanza database MySQL di origine e vengono esportate tutte le modifiche di origine che si sono verificate dopo aver interrotto la replica dalla replica di lettura Amazon RDS.

   > **Note**
   >
   > Versioni precedenti di MySQL utilizzate `START SLAVE` al posto di `START REPLICA`. Se si utilizza una versione MySQL prima della 8.0.23, utilizzare `START SLAVE`.

3. Eseguire il comando `SHOW REPLICA STATUS\G` MySQL sul database MySQL esterno per verificare che funzioni come replica di lettura. Per ulteriori informazioni sull'interpretazione dei risultati, consultare la documentazione di MySQL.
4. Dopo che la replica sul database MySQL esterno ha raggiunto l'istanza database MySQL di origine, utilizzare il comando `STOP REPLICA` MySQL per interrompere la replica dall'istanza database MySQL di origine.

   > **Note**
   >
   > Versioni precedenti di MySQL utilizzate `STOP SLAVE` al posto di `STOP REPLICA`. Se si utilizza una versione MySQL prima della 8.0.23, utilizzare `STOP SLAVE`.

5. Nella replica di lettura Amazon RDS, chiamare la stored procedure `mysql.rds_start_replication`. In questo modo Amazon RDS inizierà a eliminare i file di log binari dall'istanza database MySQL di origine.

Opzioni per le istanze database MySQL

Vengono descritte anche le opzioni o le funzionalità aggiuntive disponibili per le istanze di Amazon RDS che eseguono il motore di database MySQL. Per abilitare queste opzioni, puoi aggiungerle a un gruppo di opzioni personalizzato e quindi associare il gruppo di opzioni all'istanza database. Per ulteriori informazioni sull'utilizzo di gruppi di opzioni, consulta Uso di gruppi di opzioni.

Amazon RDS supporta le seguenti opzioni per MySQL:

Opzione	ID opzione	Versioni del motore
Supporto per MySQL del plug-in per audit MariaDB	`MARIADB_AUDIT_PLUGIN`	MySQL 8.0.28 e versioni successive alla 8.0 Tutte le versioni di MySQL 5.7
Supporto per memcached MySQL	`MEMCACHED`	Tutte le versioni di MySQL 5.7 e 8.0

Supporto per MySQL del plug-in per audit MariaDB

Amazon RDS offre un plug-in per audit per le istanze database MySQL basate sul plug-in per audit MariaDB open source. Per ulteriori informazioni, consulta Audit Plugin for MySQL Server GitHub repository.

> **Note**
>
> Il plugin per audit per MySQL è basato sul plugin per audit MariaDB. In questo articolo, lo chiamiamo plugin per audit MariaDB.

Il plug-in per audit MariaDB registra le attività del database, inclusi gli utenti che accedono al database e le query eseguite sul database. Il record con le attività del database è archiviato in un file di log.

> **Note**
>
> Al momento, il plug-in di audit MariaDB è supportato solo nelle seguenti versioni di RDS for MySQL:
>
> - MySQL 8.0.28 e versioni successive alla 8.0
> - Tutte le versioni di MySQL 5.7

Impostazioni dell'opzione relativa al plug-in per audit

Amazon RDS supporta le seguenti impostazioni per l'opzione relativa al plug-in per audit MariaDB.

Impostazione opzioni	Valori validi	Valore predefinito	Descrizione
`SERVER_AUDIT_FILE_PATH`	`/rdsdbdata/log/audit/`	`/rdsdbdata/log/audit/`	La posizione del file di log. Il file di log contiene il record dell'attività specificata in `SERVER_AUDIT_EVENTS` . Per ulteriori informazioni, consulta Visualizzazione ed elenco dei file di log del database e File di log del database MySQL.

Impostazione opzioni	Valori validi	Valore predefinito	Descrizione
SERVER_AUDIT_FILE_ROTATE_SIZE	1–1000000000	1000000	La dimensione in byte che, una volta raggiunta , provoca la rotazione del file. Per ulteriori informazioni, consulta Panoramica dei registri di database RDS per MySQL.
SERVER_AUDIT_FILE_ROTATIONS	0–100	9	Il numero di rotazioni dei log da salvare se server_audit_output_type=file . Se impostata su 0, la rotazione del file di log non viene mai eseguita. Per ulteriori informazioni, consulta Panoramica dei registri di database RDS per MySQL e Download di un file di log di database.

Impostazione opzioni	Valori validi	Valore predefinito	Descrizione
SERVER_AUDIT_EVENTS	CONNECT, QUERY, QUERY_DDL , QUERY_DML , QUERY_DML_NO_SELECT , QUERY_DCL	CONNECT, QUERY	I tipi di attività da registrate nel log. Viene registrata anche l'installazione del plug-in per audit MariaDB. • CONNECT: registrazione delle connessioni al database con esito positivo e negativo e delle disconnessioni dal database. • QUERY: registrazione del testo di tutte le query eseguite sul database. • QUERY_DDL : simile all'evento QUERY, ma restituisce solo le query DDL (Data Definition Language), (CREATE, ALTER e così via). • QUERY_DML : simile all'evento QUERY, ma restituisce solo le query DML (Data Manipulation Language), (INSERT, UPDATE, e così via e anche SELECT). • QUERY_DML_NO_SELECT : simile all'evento QUERY_DML ma non registra le query SELECT. L'impostazione QUERY_DML_NO_SELECT è supportata solo per RDS for MySQL 5.7.34 e versioni successive alla 5.7 e 8.0.25 e successive alle versioni 8.0. • QUERY_DCL : simile all'evento QUERY, ma restituisce solo le query DCL (Data Control Language), (GRANT, REVOKE e così via). Per MySQL, TABLE non è supportato.

Impostazione opzioni	Valori validi	Valore predefinito	Descrizione
SERVER_AUDIT_INCL_USERS	Più valori separati da virgola	Nessuna	Sono incluse solo le attività degli utenti specificati. Per impostazione predefinita, l'attività viene registrata per tutti gli utenti. SERVER_AUDIT_INCL_USERS e SERVER_AUDIT_EXCL_USERS si escludono a vicenda. Se si aggiungono valori a SERVER_AUDIT_INCL_USERS, è necessario assicurarsi che non venga aggiunto alcun valore a SERVER_AUDIT_EXCL_USERS.

Impostazione opzioni	Valori validi	Valore predefinito	Descrizione
SERVER_AUDIT_EXCL_USERS	Più valori separati da virgola	Nessuna	Sono escluse le attività degli utenti specifica ti. Per impostazione predefinita, l'attività viene registrata per tutti gli utenti. SERVER_AU DIT_INCL_USERS e SERVER_AU DIT_EXCL_USERS si escludono a vicenda. Se si aggiungono valori a SERVER_AU DIT_EXCL_USERS , è necessario assicurar si che non venga aggiunto alcun valore a SERVER_AUDIT_INCL_USERS . L'utente rdsadmin esegue query sul database ogni secondo per verificare l'integrità del database. In base alle altre impostazioni, questa attività può causare un rapido ed eccessivo aumento delle dimensioni del file di log. Se non desideri registrare questa attività, aggiungi l'utente rdsadmin all'elenco SERVER_AUDIT_EXCL_USERS . **Note** CONNECTL'attività viene sempre registrata per tutti gli utenti, anche se l'utente è specificato per l'impostazione di questa opzione.
SERVER_AUDIT_LOGGING	ON	ON	La registrazione è attiva. L'unico valore valido è ON. Amazon RDS non supporta la disattiva zione del logging. Se desideri disattivare la registrazione, rimuovi il plug-in per audit MariaDB. Per ulteriori informazioni, consulta Rimozione del plug-in per audit MariaDB.

Impostazione opzioni	Valori validi	Valore predefinito	Descrizione
SERVER_AUDIT_QUERY_LOG_LIMIT	0–2147483647	1.024	Il limite di lunghezza della stringa di query in un record.

Aggiunta del plug-in per audit MariaDB

Di seguito è riportato il processo generale per aggiungere il plug-in per audit MariaDB a un'istanza database:

- Creare un nuovo gruppo di opzioni oppure copiare o modificare un gruppo di opzioni esistente
- Aggiungere l'opzione al gruppo di opzioni
- Associare il gruppo di opzioni a questa istanza database

Dopo aver aggiunto il plug-in per audit MariaDB, non dovrai riavviare la tua istanza database. Non appena il gruppo di opzioni è attivo, inizia immediatamente l'audit.

> **Important**
>
> L'aggiunta del plug-in per audit MariaDB a un'istanza database può causare un errore. Consigliamo di aggiungere il plug-in per audit MariaDB durante una finestra di manutenzione o durante o un periodo di carico di lavoro del database basso.

Per aggiungere il plug-in per audit MariaDB

1. Determinare il gruppo di opzioni che si desidera usare. È possibile creare un nuovo gruppo di opzioni oppure usare un gruppo di opzioni esistente. Se si desidera usare un gruppo di opzioni esistente, puoi passare alla fase successiva. Altrimenti, creare un gruppo di opzioni database personalizzato. Scegliere mysql per Motore e selezionare 5.7 o 8.0 per Versione del motore principale. Per ulteriori informazioni, consulta Creazione di un gruppo di opzioni.
2. Aggiungere l'opzione MARIADB_AUDIT_PLUGIN al gruppo di opzioni e configurare le impostazioni dell'opzione. Per ulteriori informazioni sull'aggiunta di opzioni, consulta Aggiunta di

un'opzione a un gruppo di opzioni. Per ulteriori informazioni su ciascuna impostazione, consulta Impostazioni dell'opzione relativa al plug-in per audit.

3. Applicare il gruppo di opzioni a un'istanza database nuova o esistente.

   - Per una nuova istanza database, si applica il gruppo di opzioni quando viene avviata l'istanza. Per ulteriori informazioni, consulta Creazione di un'istanza database Amazon RDS.
   - Per un'istanza database esistente, viene applicato il gruppo di opzioni modificando l'istanza e collegando il nuovo gruppo di opzioni. Per ulteriori informazioni, consulta Modifica di un'istanza database Amazon RDS.

Formato dei registri di verifica

I file di log sono rappresentati come file con valori delimitati da virgole (CSV) in formato UTF-8.

> **Tip**
>
> Le voci dei log non sono in ordine sequenziale. Per ordinare le voci, utilizza il valore del timestamp. Per visualizzare gli ultimi eventi, potrebbe essere necessario esaminare tutti i file di log. Per una maggiore flessibilità nell'ordinamento e nella ricerca dei dati dei registri, attiva l'impostazione per caricare i registri di verifica su CloudWatch e visualizzarli utilizzando l'interfaccia CloudWatch.
> Per visualizzare i dati di verifica con più tipi di campi e con output in formato JSON, puoi inoltre utilizzare la caratteristica Database Activity Streams (Flussi di attività di database). Per ulteriori informazioni, consulta Monitoraggio di Amazon RDS tramite i flussi di attività del database.

I file dei log di audit includono le seguenti informazioni delimitate da virgola, in righe, nell'ordine specificato:

Campo	Descrizione
timestamp	`YYYYMMDD` seguito da `HH:MI:SS` (orologio a 24 ore) per l'evento registrato.
serverhost	Il nome dell'istanza per cui l'evento viene registrato.
username	Il nome utente connesso dell'utente.

Campo	Descrizione
host	L'host da cui l'utente ha effettuato la connessione.
connectionid	Il numero di ID di connessione per l'operazione registrata.
queryid	Il numero di ID di query che può essere utilizzato per trovare gli eventi di tabella relazionale e le query correlate. Per gli eventi `TABLE`, vengono aggiunte più righe.
operation	Il tipo di operazione registrata. I valori possibili sono `CONNECT`, `QUERY`, `READ`, `WRITE`, `CREATE`, `ALTER`, `RENAME` e `DROP`.
database	Il database attivo, come impostato dal comando `USE`.
oggetto	Per gli eventi `QUERY`, questo valore indica la query eseguita dal database. Per gli eventi `TABLE`, indica il nome di tabella.
retcode	Il codice restituito dell'operazione di registrazione.
connection_type	Stato di sicurezza della connessione al server. I valori possibili sono: • `0`: Undefined • `1`: TCP/IP • `2`: Socket • `3`: Named pipe • `4`: SSL/TLS • `5`: Memoria condivisa Questo campo è incluso solo per RDS for MySQL versione 5.7.34 e versioni 5.7 successive e tutte le versioni 8.0.

Visualizzazione e download del log del plug-in per audit MariaDB

Dopo avere abilitato il plug-in per audit MariaDB, potrai accedere ai risultati nei file di log nello stesso modo in cui accedi a qualsiasi altro file di log basato su testo. I file di log per audit si trovano in `/rdsdbdata/log/audit/`. Per ulteriori informazioni sulla visualizzazione del file di log nella console,

consulta Visualizzazione ed elenco dei file di log del database. Per informazioni sul download del file di log, consulta Download di un file di log di database.

Modifica delle impostazioni del plug-in per audit MariaDB

Dopo aver abilitato il plug-in per audit MariaDB, puoi modificare le impostazioni. Per ulteriori informazioni su come modificare le impostazioni dell'opzione, consulta Modifica di un'impostazione di un'opzione. Per ulteriori informazioni su ciascuna impostazione, consulta Impostazioni dell'opzione relativa al plug-in per audit.

Rimozione del plug-in per audit MariaDB

Amazon RDS non supporta la disattivazione della registrazione nel plug-in per audit MariaDB. Puoi tuttavia rimuovere il plug-in da un'istanza database. Dopo aver rimosso il plug-in per audit MariaDB, l'istanza database viene riavviata automaticamente per arrestare l'audit.

Per rimuovere il plug-in per audit MariaDB da un'istanza database, procedi in uno dei seguenti modi:

- Rimuovere l'opzione relativa al plug-in per audit MariaDB dal gruppo di opzioni a cui appartiene. Questa modifica coinvolge tutte le istanze database che usano il gruppo di opzioni. Per ulteriori informazioni, consulta Rimozione di un'opzione da un gruppo di opzioni
- Modificare l'istanza database e specificare un diverso gruppo di opzioni che non comprenda il plug-in. Questa modifica coinvolge una singola istanza database. È possibile specificare il gruppo di opzioni predefinito (vuoto) o un gruppo di opzioni personalizzato diverso. Per ulteriori informazioni, consulta Modifica di un'istanza database Amazon RDS.

Supporto per memcached MySQL

Amazon RDS supporta l'uso dell'interfaccia memcached in tabelle InnoDB, introdotta in MySQL 5.6. L'API memcached permette alle applicazioni di usare tabelle InnoDB in modo simile ai datastore di coppie chiave/valore NoSQL.

L'interfaccia memcached è una cache semplice basata su chiave. Le applicazioni utilizzano memcached per inserire, manipolare e recuperare dalla cache coppie di dati a chiave-valore. MySQL 5.6 ha introdotto un plugin che implementa un servizio daemon che espone dati di tabelle InnoDB attraverso il protocollo memcached. Per ulteriori informazioni sul plugin memcached MySQL, consulta InnoDB Integration with memcached.

Per abilitare il supporto per memcached per un'istanza database RDS per MySQL

1. Determinare il gruppo di sicurezza da utilizzare per controllare l'accesso all'interfaccia memcached. Se il set di applicazioni che sta già utilizzando l'interfaccia SQL è lo stesso che accederà all'interfaccia memcached, è possibile utilizzare lo stesso gruppo di sicurezza VPC utilizzato dall'interfaccia SQL. Se il set di applicazioni che accederà all'interfaccia memcached è diverso, definire un nuovo gruppo di sicurezza VPC o DB. Per ulteriori informazioni sulla gestione dei gruppi di sicurezza, consulta Controllo dell'accesso con i gruppi di sicurezza
2. Creare un gruppo di opzioni di database personalizzato, selezionando MySQL come tipo di motore e versione. Per ulteriori informazioni sulla creazione di un gruppo di opzioni, consulta Creazione di un gruppo di opzioni.
3. Aggiungere l'opzione MEMCACHED al gruppo di opzioni. Specificare la porta di memcached che l'interfaccia utilizzerà e il gruppo di sicurezza da utilizzare per il controllo dell'accesso all'interfaccia. Per ulteriori informazioni sull'aggiunta di opzioni, consulta Aggiunta di un'opzione a un gruppo di opzioni.
4. Se necessario, modificare le impostazioni di opzione per configurare i parametri di memcached. Per ulteriori informazioni su come modificare le impostazioni dell'opzione, consulta Modifica di un'impostazione di un'opzione.
5. Applica il gruppo di opzioni a un'istanza. Amazon RDS abilita il supporto di memcached per l'istanza quando viene applicato il gruppo di opzioni:

    - Il supporto memcached per una nuova istanza viene abilitato specificando il gruppo di opzioni personalizzato quando viene avviata l'istanza. Per ulteriori informazioni sul lancio di un'istanza MySQL, consulta Creazione di un'istanza database Amazon RDS.

- Il supporto memcached per un'istanza esistente viene abilitato specificando il gruppo di opzioni personalizzato quando si modifica l'istanza. Per ulteriori informazioni sulla modifica di un'istanza database , consulta Modifica di un'istanza database Amazon RDS.

6. Specificare a quali colonne delle tabelle MySQL è possibile accedere attraverso l'interfaccia memcached. Il plugin memcached crea una tabella catalogo chiamata containers all'interno di un database dedicato denominato innodb_memcache. Inserire una riga nella tabella containers per mappare una tabella InnoDB per l'accesso tramite memcached. Specificare una colonna della tabella InnoDB utilizzata per archiviare i valori della chiave memcached e una o più colonne utilizzate per archiviare i valori dei dati associati alla chiave. Specificare anche il nome utilizzato da un'applicazione memcached per fare riferimento a quel set di colonne. Per i dettagli su come inserire righe nella tabella containers, consulta InnoDB memcached Plugin Internals. Per un esempio di mappatura di una tabella InnoDB e di accesso ad essa tramite memcached, consulta Writing Applications for the InnoDB memcached Plugin.
7. Se le applicazioni che accedono all'interfaccia memcached si trovano su computer o istanze EC2 diversi dalle applicazioni che utilizzano l'interfaccia SQL, aggiungere le informazioni di connessione per questi computer al gruppo di sicurezza VPC associato all'istanza MySQL. Per ulteriori informazioni sulla gestione dei gruppi di sicurezza, consulta Controllo dell'accesso con i gruppi di sicurezza.

Disabilitare il supporto memcached per un'istanza modificandola e specificando il gruppo di opzioni predefinite per la versione di MySQL. Per ulteriori informazioni sulla modifica di un'istanza database , consulta Modifica di un'istanza database Amazon RDS.

Considerazioni di sicurezza per memcached di MySQL

Il protocollo memcached non supporta l'autenticazione degli utenti. Per ulteriori informazioni sulle considerazioni sulla sicurezza di MySQL memcached, consulta Considerazioni di sicurezza per il plugin memcached InnoDB nella documentazione di MySQL.

Per migliorare la sicurezza dell'interfaccia memcached, è possibile adottare le seguenti misure:

- Quando aggiungi l'opzione MEMCACHED al gruppo di opzioni, specifica una porta diversa dalla porta 11211 predefinita.
- Verifica di associare l'interfaccia memcached a un gruppo di sicurezza VPC o DB che limiti l'accesso a indirizzi client o istanze EC2 conosciuti e attendibili. Per ulteriori informazioni sulla gestione dei gruppi di sicurezza, consulta Controllo dell'accesso con i gruppi di sicurezza.

Informazioni sulla connessione di memcached a MySQL

Per accedere all'interfaccia `memcached`, un'applicazione deve specificare sia il nome DNS sia l'istanza Amazon RDS e il numero di porta di `memcached`. Ad esempio, se il nome DNS di un'istanza è `my-cache-instance.cg034hpkmmjt.region.rds.amazonaws.com` e l'interfaccia memcached sta usando la porta 11212, le informazioni di connessione specificate in PHP saranno:

```
<?php

$cache = new Memcache;
$cache->connect('my-cache-instance.cg034hpkmmjt.region.rds.amazonaws.com',11212);
?>
```

Per trovare il nome DNS e la porta di memcached di un'istanza database MySQL

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nell'angolo in alto a destra di AWS Management Console, seleziona la regione che contiene l'istanza DB.
3. Nel riquadro di navigazione, scegliere Databases (Database).
4. Scegliere il nome dell'istanza database MySQL per visualizzarne i dettagli.
5. Nella sezione Connect (Connessione), prendere nota del valore del campo Endpoint. Il nome DNS è lo stesso dell'endpoint. Inoltre, tenere presente che la porta nella sezione Connect (Connessione) non si utilizza per accedere all'interfaccia `memcached`.
6. Nella sezione Details (Dettagli), prendere nota del nome elencato nel campo Option Group (Gruppo di opzioni).
7. Nel riquadro di navigazione scegliere Option groups (Gruppi di opzioni).
8. Scegliere il nome del gruppo di opzioni utilizzato dall'istanza database MySQL per visualizzare i dettagli del gruppo di opzioni. Nella sezione Options (Opzioni), prendere nota del valore dell'impostazione Port (Porta) per l'opzione MEMCACHED.

Impostazioni dell'opzione memcached di MySQL

Amazon RDS espone i parametri `memcached` di MySQL come impostazioni di opzione nell'opzione `MEMCACHED` di Amazon RDS.

Parametri di memcached di MySQL

- DAEMON_MEMCACHED_R_BATCH_SIZE: un valore intero che specifica quante operazioni di lettura (get) di memcached è necessario eseguire prima di effettuare un COMMIT per iniziare una nuova transazione. I valori validi sono compresi tra 1 e 4294967295; il valore predefinito è 1. L'opzione non ha effetto fino a quando l'istanza non viene riavviata.
- DAEMON_MEMCACHED_W_BATCH_SIZE: un valore intero che specifica quante operazioni di scrittura di memcached, come add, set o incr, è necessario eseguire prima di effettuare un COMMIT per iniziare una nuova transazione. I valori validi sono compresi tra 1 e 4294967295; il valore predefinito è 1. L'opzione non ha effetto fino a quando l'istanza non viene riavviata.
- INNODB_API_BK_COMMIT_INTERVAL: un valore intero che specifica con quanta frequenza effettuare auto-commit delle connessioni inattive che utilizzano l'interfaccia memcached di InnoDB. I valori validi sono compresi tra 1 e 1073741824; il valore predefinito è 5. L'opzione ha effetto immediato senza necessità di riavviare l'istanza.
- INNODB_API_DISABLE_ROWLOCK: un valore booleano che disabilita (1 (true)) o abilita (0 (false)) l'utilizzo di blocchi di riga quando si utilizza l'interfaccia memcached di InnoDB. Il valore predefinito è 0 (false). L'opzione non ha effetto fino a quando l'istanza non viene riavviata.
- INNODB_API_ENABLE_MDL: un valore booleano che quando impostato su 0 (false) blocca la tabella utilizzata dal plugin memcached di InnoDB, in modo che non possa essere eliminata o modificata da DDL attraverso l'interfaccia SQL. Il valore predefinito è 0 (false). L'opzione non ha effetto fino a quando l'istanza non viene riavviata.
- INNODB_API_TRX_LEVEL: un valore intero che specifica il livello di isolamento della transazione per query elaborate dall'interfaccia memcached. I valori consentiti sono da 0 a 3. Il valore predefinito è 0. L'opzione non ha effetto fino a quando l'istanza non viene riavviata.

Amazon RDS configura questi parametri memcached di MySQL, che non possono essere modificati: DAEMON_MEMCACHED_LIB_NAME, DAEMON_MEMCACHED_LIB_PATH e INNODB_API_ENABLE_BINLOG. I parametri impostati dagli amministratori di MySQL tramite daemon_memcached_options sono disponibili come singole impostazioni dell'opzione MEMCACHED in Amazon RDS.

Parametri daemon_memcached_options di MySQL

- BINDING_PROTOCOL: una stringa che specifica il protocollo binding da utilizzare. I valori consentiti sono auto, ascii o binary. Quello predefinito è auto, vale a dire che il server negozia

automaticamente il protocollo con il client. L'opzione non ha effetto fino a quando l'istanza non viene riavviata.

- `BACKLOG_QUEUE_LIMIT` – un valore intero che specifica quante connessioni di rete possono essere in attesa di elaborazione da parte di `memcached`. L'aumento di questo limite potrebbe ridurre gli errori ricevuti da un client che non riesce a effettuare la connessione all'istanza `memcached`, ma non migliora le prestazioni del server. I valori validi sono compresi tra 1 e 2048; il valore predefinito è 1024. L'opzione non ha effetto fino a quando l'istanza non viene riavviata.
- `CAS_DISABLED`: un valore booleano che abilita (1 (true)) o disabilita (0 (false)) l'utilizzo di compare and swap (CAS), che riduce di 8 byte le dimensioni per voce. Il valore predefinito è 0 (false). L'opzione non ha effetto fino a quando l'istanza non viene riavviata.
- `CHUNK_SIZE`: un valore intero che specifica le dimensioni minime (in byte) del blocco da allocare per chiave, valore e flag della voce più piccola. I valori consentiti sono da 1 a 48. Quello predefinito è 48, ma è possibile migliorare notevolmente l'efficienza della memoria con un valore inferiore. L'opzione non ha effetto fino a quando l'istanza non viene riavviata.
- `CHUNK_SIZE_GROWTH_FACTOR`: un valore float che controlla le dimensioni dei nuovi blocchi. Le dimensioni di un nuovo blocco sono le dimensioni del blocco precedente moltiplicato per `CHUNK_SIZE_GROWTH_FACTOR`. I valori validi sono compresi tra 1 e 2; il valore predefinito è 1,25. L'opzione non ha effetto fino a quando l'istanza non viene riavviata.
- `ERROR_ON_MEMORY_EXHAUSTED`: un valore booleano che, quando è impostato su 1 (true), specifica che `memcached` restituirà un errore anziché eliminare voci quando si esaurisce la memoria per archiviarle. Se impostato su 0 (false), `memcached` eliminerà elementi se la memoria è esaurita. Il valore predefinito è 0 (false). L'opzione non ha effetto fino a quando l'istanza non viene riavviata.
- `MAX_SIMULTANEOUS_CONNECTIONS`: un valore intero che specifica il numero massimo di connessioni simultanee. Impostare questo valore al di sotto di 10 impedisce l'avvio di MySQL. I valori validi sono compresi tra 10 e 1024; il valore predefinito è 1024. L'opzione non ha effetto fino a quando l'istanza non viene riavviata.
- `VERBOSITY`: una stringa che specifica il livello di informazioni registrate dal servizio `memcached` nel log degli errori di MySQL. Il valore predefinito è v. L'opzione non ha effetto fino a quando l'istanza non viene riavviata. I valori consentiti sono:
  - `v` – Avvisi ed errori dei log durante l'esecuzione del loop eventi principale.
  - `vv`: oltre alle informazioni registrate da v, registra anche ogni comando del client e la risposta.
  - `vvv`: oltre alle informazioni registrate da vv, registra anche le transizioni dello stato interno.

Amazon RDS configura questi parametri DAEMON_MEMCACHED_OPTIONS di MySQL, che non possono essere modificati: DAEMON_PROCESS, LARGE_MEMORY_PAGES, MAXIMUM_CORE_FILE_LIMIT, MAX_ITEM_SIZE, LOCK_DOWN_PAGE_MEMORY, MASK, IDFILE, REQUESTS_PER_EVENT, SOCKET e USER.

Parametri per MySQL

Per impostazione predefinita, un'istanza database MySQL utilizza un gruppo di parametri database specifico a un database MySQL. Questo gruppo di parametri contiene parametri per il motore del database MySQL. Per informazioni sull'utilizzo dei gruppi di parametri e sull'impostazione dei parametri, consulta Utilizzo di gruppi di parametri.

I parametri di RDS for MySQL sono impostati sui valori predefiniti del motore di storage selezionati. Per ulteriori informazioni sui parametri di MySQL, consulta la documentazione di MySQL. Per ulteriori informazioni sui motori di storage MySQL, consulta Motori di storage supportati per RDS for MySQL.

È possibile visualizzare i parametri disponibili per una versione RDS for MySQL specifica utilizzando la console RDS o la AWS CLI. Per informazioni sulla visualizzazione dei parametri in un gruppo di parametri MySQL nella console RDS, consulta Visualizzazione dei valori dei parametri per un gruppo di parametri del database.

Utilizzando AWS CLI, è possibile visualizzare i parametri di una versione RDS for MySQL eseguendo il comando `describe-engine-default-parameters`. Indica uno dei valori seguenti per l'opzione `--db-parameter-group-family`:

- `mysql8.0`
- `mysql5.7`

Ad esempio, per visualizzare i parametri supportati per RDS for MySQL versione 8.0 esegui il comando seguente.

```
aws rds describe-engine-default-parameters --db-parameter-group-family mysql8.0
```

L'output avrà un aspetto simile al seguente.

```
{
    "EngineDefaults": {
        "Parameters": [
            {
                "ParameterName": "activate_all_roles_on_login",
                "ParameterValue": "0",
                "Description": "Automatically set all granted roles as active after the
 user has authenticated successfully.",
                "Source": "engine-default",
```

```
                "ApplyType": "dynamic",
                "DataType": "boolean",
                "AllowedValues": "0,1",
                "IsModifiable": true
            },
            {
                "ParameterName": "allow-suspicious-udfs",
                "Description": "Controls whether user-defined functions that have only
 an xxx symbol for the main function can be loaded",
                "Source": "engine-default",
                "ApplyType": "static",
                "DataType": "boolean",
                "AllowedValues": "0,1",
                "IsModifiable": false
            },
            {
                "ParameterName": "auto_generate_certs",
                "Description": "Controls whether the server autogenerates SSL key and
 certificate files in the data directory, if they do not already exist.",
                "Source": "engine-default",
                "ApplyType": "static",
                "DataType": "boolean",
                "AllowedValues": "0,1",
                "IsModifiable": false
            },
        ...
```

Per visualizzare i parametri supportati per RDS for MySQL versione 8.0 esegui il comando seguente.

Per LinuxmacOS, oUnix:

```
aws rds describe-engine-default-parameters --db-parameter-group-family mysql8.0 \
   --query 'EngineDefaults.Parameters[?IsModifiable==`true`]'
```

Per Windows:

```
aws rds describe-engine-default-parameters --db-parameter-group-family mysql8.0 ^
   --query "EngineDefaults.Parameters[?IsModifiable==`true`]"
```

Attività di log DBA comuni per istanze database MySQL

Nel seguente contenuto, puoi trovare le descrizioni delle implementazioni specifiche di Amazon RDS di alcune attività DBA comuni per le istanze DB che eseguono il motore di database MySQL. Per offrire un'esperienza di servizio gestito, Amazon RDS non fornisce accesso shell alle istanze database. Limita anche l'accesso ad alcune procedure di sistema e tabelle che richiedono privilegi avanzati.

Per informazioni sull'uso di file di log MySQL in Amazon RDS, consult File di log del database MySQL.

Argomenti

- Comprendere gli utenti predefiniti
- Privilegio basato sui ruoli
- Terminare una sessione o una query
- Ignorare l'errore di replica corrente
- Lavorare con gli spazi tabella InnoDB per migliorare i tempi di ripristino dopo un arresto anomalo
- Gestione della cronologia di stato globale

Comprendere gli utenti predefiniti

Amazon RDS crea automaticamente diversi utenti predefiniti con nuove istanze DB RDS per MySQL. Gli utenti predefiniti e i relativi privilegi non possono essere modificati. Non è possibile eliminare, rinominare o modificare i privilegi per questi utenti predefiniti. Qualsiasi tentativo comporta la generazione di un errore.

- rdsadmin: utente creato per gestire molte delle attività di gestione che l'amministratore con `superuser` privilegi eseguirebbe su un database MySQL autonomo. Questo utente viene utilizzato internamente da RDS for MySQL per molte attività di gestione.
- rdsrepladmin: utente utilizzato internamente da Amazon RDS per supportare le attività di replica su istanze e cluster RDS for MySQL DB.

Privilegio basato sui ruoli

A partire dalla versione 8.0.36 di RDS for MySQL, non è possibile modificare direttamente le tabelle nel database. `mysql` In particolare, non è possibile creare utenti del database eseguendo operazioni

DML (Data Manipulation Language) sulle tabelle. `grant` Si utilizzano invece istruzioni di gestione degli account MySQL `CREATE USER` come`GRANT`, `REVOKE` e per concedere privilegi basati sui ruoli agli utenti. Inoltre, nel database `mysql`, non è possibile creare altri tipi di oggetti come le stored procedure. È comunque possibile interrogare le tabelle di `mysql`. Se si utilizza la replica dei log binari, le modifiche apportate direttamente alle `mysql` tabelle sull'istanza DB di origine non vengono replicate nel cluster di destinazione.

In alcuni casi, l'applicazione potrebbe utilizzare scorciatoie per creare utenti o altri oggetti inserendoli nelle tabelle di `mysql`. In tal caso, modifica il codice dell'applicazione per utilizzare le istruzioni corrispondenti come `CREATE USER`.

Per esportare i metadati per gli utenti del database durante la migrazione da un database MySQL esterno, utilizzare uno dei seguenti metodi:

- Utilizza l'utilità di dump delle istanze di MySQL Shell con un filtro per escludere utenti, ruoli e concessioni. L'esempio seguente mostra la sintassi del comando da utilizzare. Assicurati che `outputUrl` sia vuoto.

  ```
  mysqlsh user@host -- util.dumpInstance(outputUrl,{excludeSchemas:['mysql'],users:
   true})
  ```

  Per ulteriori informazioni, vedere Instance Dump Utility, Schema Dump Utility e Table Dump Utility nel MySQL Reference Manual.
- Usa l'utilità client. `mysqlpump` Questo esempio include tutte le tabelle ad eccezione delle tabelle del database `mysql` di sistema. Include anche istruzioni `CREATE USER` e `GRANT` per riprodurre tutti gli utenti MySQL nel database migrato.

  ```
  mysqlpump --exclude-databases=mysql --users
  ```

Per semplificare la gestione delle autorizzazioni per molti utenti o applicazioni, è possibile utilizzare l'istruzione `CREATE ROLE` per creare un ruolo con una serie di autorizzazioni. Puoi quindi utilizzare le istruzioni `GRANT` e `SET ROLE` e la funzione `current_role` per assegnare ruoli a utenti o applicazioni, cambiare il ruolo corrente e verificare quali ruoli sono in vigore. Per ulteriori informazioni sul sistema di autorizzazione basato sui ruoli in MySQL 8.0, consultare Utilizzo di ruoli nel Manuale di riferimento di MySQL.

> **Important**
>
> Si consiglia di non utilizzare l'utente master direttamente nelle applicazioni. Rispetta piuttosto la best practice di utilizzare un utente del database creato con i privilegi minimi richiesti per l'applicazione.

A partire dalla versione 8.0.36, RDS for MySQL include un ruolo speciale con tutti i seguenti privilegi. Il ruolo è denominato `rds_superuser_role`. Questo ruolo è già assegnato all'utente amministrativo principale di ogni istanza DB. Il ruolo `rds_superuser_role` include i seguenti privilegi per tutti gli oggetti del database:

- `ALTER`
- `APPLICATION_PASSWORD_ADMIN`
- `ALTER ROUTINE`
- `CREATE`
- `CREATE ROLE`
- `CREATE ROUTINE`
- `CREATE TEMPORARY TABLES`
- `CREATE USER`
- `CREATE VIEW`
- `DELETE`
- `DROP`
- `DROP ROLE`
- `EVENT`
- `EXECUTE`
- `INDEX`
- `INSERT`
- `LOCK TABLES`
- `PROCESS`
- `REFERENCES`
- `RELOAD`
- `REPLICATION CLIENT`

- REPLICATION SLAVE
- ROLE_ADMIN
- SET_USER_ID
- SELECT
- SHOW DATABASES
- SHOW VIEW
- TRIGGER
- UPDATE
- XA_RECOVER_ADMIN

La definizione del ruolo include anche la WITH GRANT OPTION in modo che un utente amministrativo possa concedere tale ruolo ad altri utenti. In particolare, l'amministratore deve concedere tutti i privilegi necessari per eseguire la replica dei log binari con il cluster MySQL come destinazione.

> **Tip**
>
> Per visualizzare i dettagli completi delle autorizzazioni, utilizzare la seguente dichiarazione.
>
> ```
> SHOW GRANTS FOR rds_superuser_role@'%';
> ```

Quando concedi l'accesso utilizzando i ruoli in RDS for MySQL versione 8.0.36 e successive, attivi il ruolo anche utilizzando l'istruzione or. SET ROLE *role_name* SET ROLE ALL L'esempio seguente mostra come. Sostituire il nome del ruolo appropriato per CUSTOM_ROLE.

```
# Grant role to user
mysql> GRANT CUSTOM_ROLE TO 'user'@'domain-or-ip-address'

# Check the current roles for your user. In this case, the CUSTOM_ROLE role has not
 been activated.
# Only the rds_superuser_role is currently in effect.
mysql> SELECT CURRENT_ROLE();
+--------------------------+
| CURRENT_ROLE()           |
+--------------------------+
```

```
| `rds_superuser_role`@`%` |
+--------------------------+
1 row in set (0.00 sec)

# Activate all roles associated with this user using SET ROLE.
# You can activate specific roles or all roles.
# In this case, the user only has 2 roles, so we specify ALL.
mysql> SET ROLE ALL;
Query OK, 0 rows affected (0.00 sec)

# Verify role is now active
mysql> SELECT CURRENT_ROLE();
+-------------------------------------------------+
| CURRENT_ROLE()                                  |
+-------------------------------------------------+
| `CUSTOM_ROLE`@`%`,`rds_superuser_role`@`%` |
+-------------------------------------------------+
```

Terminare una sessione o una query

Puoi terminare query o sessioni utente sulle istanze database utilizzando i comandi `rds_kill` e `rds_kill_query`. Connettiti alla tua istanza database MySQL, quindi immetti il comando appropriato come mostrato di seguito. Per ulteriori informazioni, consulta Connessione a un'istanza database che esegue il motore di database di MySQL.

```
CALL mysql.rds_kill(thread-ID)
CALL mysql.rds_kill_query(thread-ID)
```

Ad esempio, per terminare la sessione in esecuzione sul thread 99, dovresti digitare:

```
CALL mysql.rds_kill(99);
```

Per terminare la query in esecuzione sul thread 99, dovresti digitare:

```
CALL mysql.rds_kill_query(99);
```

Ignorare l'errore di replica corrente

È possibile ignorare un errore sulle repliche di lettura se l'errore provoca il blocco della replica di lettura e non compromette l'integrità dei dati.

> **Note**
>
> Dovrai prima verificare che sia sicuro ignorare l'errore. In una utility MySQL, connettiti alla replica di lettura ed esegui il seguente comando MySQL.
>
> ```
> SHOW REPLICA STATUS\G
> ```
>
> Per informazioni sui valori restituiti, vedere la documentazione di MySQL.
> Le versioni precedenti di MySQL utilizzavano `SHOW SLAVE STATUS` anziché `SHOW REPLICA STATUS`. Se si utilizza una versione MySQL prima della 8.0.23, utilizzare `SHOW SLAVE STATUS`.

È possibile saltare un errore nella replica di lettura nei seguenti modi.

Argomenti

- Chiamata della procedura mysql.rds_skip_repl_error
- Impostazione del parametro slave_skip_errors

Chiamata della procedura mysql.rds_skip_repl_error

Amazon RDS fornisce una stored procedure che puoi chiamare per saltare un errore nelle repliche di lettura. In primo luogo, collegati alla replica di lettura, quindi emetti i comandi appropriati come mostrato qui di seguito. Per ulteriori informazioni, consulta Connessione a un'istanza database che esegue il motore di database di MySQL.

Per ignorare l'errore, puoi emettere il seguente comando.

```
CALL mysql.rds_skip_repl_error;
```

Questo comando non ha alcun effetto se lo esegui sull'istanza database di origine o in una replica di lettura che non ha riscontrato un errore di replica.

Per ulteriori informazioni, ad esempio per conoscere le versioni di MySQL che supportano `mysql.rds_skip_repl_error`, consulta mysql.rds_skip_repl_error.

> **Important**
>
> Se tenti di chiamare `mysql.rds_skip_repl_error` e incontri questo errore: `ERROR 1305 (42000): PROCEDURE mysql.rds_skip_repl_error does not exist`, aggiorna l'istanza database MySQL alla versione secondaria più recente o a una delle versioni secondarie minime indicate in mysql.rds_skip_repl_error.

Impostazione del parametro slave_skip_errors

Per saltare uno o più errori, puoi impostare il parametro statico `slave_skip_errors` sulla replica di lettura. Puoi impostare questo parametro per saltare uno o più codici di errore di replica specifici. Attualmente, puoi impostare questo parametro solo per le istanze DB per RDS for MySQL 5.7. Dopo aver modificato l'impostazione per questo parametro, accertati di riavviare l'istanza database per rendere effettiva la nuova impostazione. Per ulteriori informazioni su questo parametro, consulta la documentazione di MySQL.

Consigliamo di impostare questo parametro in un gruppo di parametri database a parte. Puoi associare questo gruppo di parametri database solo alle repliche di lettura che devono saltare gli errori. Seguendo questa best practice riduci il potenziale impatto su altre istanze database e sulle repliche di lettura.

> **Important**
>
> L'impostazione di un valore non predefinito per questo parametro può causare incoerenza nella replica. Impostare questo parametro su un valore non predefinito solo se sono state esaurite le altre opzioni per risolvere il problema e si è sicuri del potenziale impatto sui dati della replica di lettura.

Lavorare con gli spazi tabella InnoDB per migliorare i tempi di ripristino dopo un arresto anomalo

Ogni tabella in MySQL è formata da una definizione della tabella, dati e indici. Il motore di storage InnoDB MySQL salva gli indici e i dati della tabella in uno spazio tabella. InnoDB crea uno spazio tabella globale condiviso che contiene un dizionario di dati e altri metadati rilevanti e che può inoltre contenere indici e dati della tabella. InnoDB può anche creare degli spazi tabella per ciascuna tabella

e partizione. Questi spazi tabella separati vengono salvati in file con estensione .ibd e l'intestazione di ciascuno spazio tabella contiene un numero identificativo univoco.

Amazon RDS fornisce un parametro in un gruppo di parametri MySQL denominato `innodb_file_per_table`. Questo parametro controlla se InnoDB aggiunge nuovi dati e indici di tabella allo spazio di tabella condiviso (impostando il valore del parametro su 0) o a singoli spazi di tabella (impostando il valore del parametro su 1). Amazon RDS imposta il valore predefinito per il parametro `innodb_file_per_table` su 1, che consente di eliminare singole tabelle InnoDB e recuperare l'archiviazione utilizzata da tali tabelle per l'istanza database. Nella maggior parte dei casi d'uso l'impostazione del parametro `innodb_file_per_table` su 1 rappresenta l'opzione consigliata.

Dovresti impostare il parametro `innodb_file_per_table` su 0 quando hai un numero elevato di tabelle, ad esempio oltre 1.000 tabelle quando utilizzi lo storage standard (magnetico) o lo storage General Purpose SSD oppure oltre 10.000 tabelle quando utilizzi lo storage Provisioned IOPS Quando imposti questo parametro su 0, non vengono creati singoli spazi tabella, pertanto il ripristino dopo un arresto anomalo del database viene completato in minor tempo.

MySQL elabora ciascun file dei metadati, che include spazi tabella durante il ciclo di recupero dopo un arresto anomalo. Il tempo richiesto da MySQL per elaborare le informazioni dei metadati negli spazi tabella condivisi è trascurabile rispetto al tempo necessario per elaborare migliaia di file di spazi tabella quando sono presenti più spazi tabella. Poiché il numero di spazi tabella viene salvato nell'intestazione di ciascun file, il tempo complessivo per leggere tutti i file degli spazi tabella può essere di diverse ore. Ad esempio, per elaborare un milione di spazi tabella InnoDB nello storage standard per un ciclo di ripristino dopo un arresto anomalo potrebbero essere necessarie da cinque a otto ore. In alcuni casi, InnoDB può reputare la necessità di una pulizia aggiuntiva dopo un ciclo di ripristino dopo un arresto anomalo che attiverà un altro ciclo di ripristino dopo un arresto anomalo, rendendo più lungo il tempo di ripristino. Ricorda che un ciclo di ripristino dopo un arresto anomalo, oltre all'elaborazione delle informazioni degli spazi tabella, implica transazioni di rollback, riparazioni delle pagine non funzionanti e altre operazioni.

Poiché il parametro `innodb_file_per_table` risiede in un gruppo di parametri, puoi cambiare il valore del parametro modificando il gruppo di parametri utilizzato dalla tua istanza database senza riavviarla. Dopo aver cambiato l'impostazione, ad esempio da 1 (per creare tabelle individuali) a 0 (per utilizzare gli spazi tabella condivisi), allo spazio tabella condiviso saranno aggiunte altre tabelle InnoDB, mentre quelle esistenti continueranno ad avere degli spazi tabella individuali. Per spostare una tabella InnoDB in uno spazio tabella condiviso, devi utilizzare il comando `ALTER TABLE`.

Migrazione di più spazi tabella in uno spazio tabella condiviso

Puoi spostare i metadati di una tabella InnoDB dal loro spazio tabella allo spazio tabella condiviso che ricompilerà i metadati della tabella secondo l'impostazione del parametro `innodb_file_per_table`. Connettiti innanzitutto alla tua istanza database MySQL, quindi utilizza i comandi appropriati come mostrato di seguito. Per ulteriori informazioni, consulta Connessione a un'istanza database che esegue il motore di database di MySQL.

```
ALTER TABLE table_name ENGINE = InnoDB, ALGORITHM=COPY;
```

Ad esempio, la seguente query restituisce un'istruzione `ALTER TABLE` per ogni tabella InnoDB non presente nello spazio tabella condiviso.

Per le istanze database MySQL 5.7:

```
SELECT CONCAT('ALTER TABLE `',
REPLACE(LEFT(NAME , INSTR((NAME), '/') - 1), '`', '``'), '`.`',
REPLACE(SUBSTR(NAME FROM INSTR(NAME, '/') + 1), '`', '``'), '` ENGINE=InnoDB,
 ALGORITHM=COPY;') AS Query
FROM INFORMATION_SCHEMA.INNODB_SYS_TABLES
WHERE SPACE <> 0 AND LEFT(NAME, INSTR((NAME), '/') - 1) NOT IN ('mysql','');
```

Per le istanze database MySQL 8.0:

```
SELECT CONCAT('ALTER TABLE `',
REPLACE(LEFT(NAME , INSTR((NAME), '/') - 1), '`', '``'), '`.`',
REPLACE(SUBSTR(NAME FROM INSTR(NAME, '/') + 1), '`', '``'), '` ENGINE=InnoDB,
 ALGORITHM=COPY;') AS Query
FROM INFORMATION_SCHEMA.INNODB_TABLES
WHERE SPACE <> 0 AND LEFT(NAME, INSTR((NAME), '/') - 1) NOT IN ('mysql','');
```

La ricompilazione di una tabella MySQL per spostare i metadati della tabella nello spazio tabella condiviso richiede uno spazio di storage ulteriore disponibile nell'istanza database. Durante la ricompilazione, la tabella è bloccata e inaccessibile alle query. Per le tabelle di piccole dimensioni o le tabelle a cui non si accede di frequente, questo potrebbe non essere un problema. Per tabelle di grandi dimensioni o tabelle a cui si accede di frequente in un ambiente con molti processi simultanei, puoi ricompilare le tabelle su una replica di lettura.

Puoi creare una replica di lettura e migrare i metadati della tabella nello spazio tabella condiviso sulla replica di lettura. Mentre l'istruzione ALTER TABLE blocca l'accesso sulla replica di lettura, l'istanza

database di origine non viene interessata. L'istanza database di origine continuerà a generare i suoi log binari, mentre la replica di lettura sarà in ritardo durante il processo di ricompilazione della tabella. Poiché la ricompilazione richiede spazio di storage aggiuntivo e il file di log di riproduzione può essere di grandi dimensioni, dovresti creare una replica di lettura con uno storage allocato più grande rispetto all'istanza database di origine.

Per creare una replica di lettura e ricompilare le tabelle InnoDB per utilizzare lo spazio tabella condiviso, procedere come indicato di seguito:

1. Assicurarsi che la retention dei backup sia abilitata sull'istanza database di origine in modo che sia abilitato il log binario.
2. Utilizzate AWS Management Console o AWS CLI per creare una replica di lettura per l'istanza DB di origine. Poiché la creazione di una replica di lettura richiede molti degli stessi processi di un ripristino dopo un arresto anomalo, il processo di creazione potrebbe richiedere diverso tempo se sono presenti numerosi spazi tabella InnoDB. Allocare più spazio di storage sulla replica di lettura rispetto a quello attualmente utilizzato sull'istanza database.
3. Una volta creata la replica di lettura, creare un gruppo di parametri con le impostazioni dei parametri `read_only = 0` e `innodb_file_per_table = 0`. Quindi, associare il gruppo di parametri alla replica di lettura.
4. Immettere la seguente istruzione SQL per tutte le tabelle di cui si desidera eseguire la migrazione nella replica:

   ```
   ALTER TABLE name ENGINE = InnoDB
   ```

5. Quando tutte le istruzioni `ALTER TABLE` sono state completate sulla replica di lettura, verificare che quest'ultima sia connessa all'istanza database di origine e che le due istanze siano sincronizzate.
6. Utilizzare la console o CLI per promuovere la replica di lettura a istanza. Assicurarsi che il gruppo di parametri utilizzato per la nuova istanza DB standalone abbia il parametro `innodb_file_per_table` impostato su 0. Modificare il nome della nuova istanza DB standalone e puntare tutte le applicazioni alla nuova istanza DB standalone.

Gestione della cronologia di stato globale

> **Tip**
>
> Per analizzare le prestazioni del database, puoi anche utilizzare la funzionalità Approfondimenti sulle prestazioni su Amazon RDS. Per ulteriori informazioni, consulta Monitoraggio del carico DB con Performance Insights su Amazon RDS.

MySQL mantiene molte variabili di stato che forniscono informazioni sul suo funzionamento. Il loro valore può aiutarti a rilevare problemi di blocco o di memoria su un'istanza database. I valori di queste variabili di stato sono cumulativi dal momento dell'ultimo avvio dell'istanza database. Puoi reimpostare la maggiore parte delle variabili di stato su 0 utilizzando il comando `FLUSH STATUS`.

Per consentire il monitoraggio di questi valori nel tempo, Amazon RDS fornisce un insieme di procedure che effettuano una snapshot dei valori di queste variabili di stato nel tempo e li scrivono su una tabella insieme alle modifiche eseguite dall'ultima snapshot. Questa infrastruttura, denominata GoSH (Global Status History, cronologia di stato globale), viene installata in tutte le istanze database MySQL a partire dalle versioni 5.5.23. La funzione GoSH è disabilitata per impostazione predefinita.

Per abilitare la funzione GoSH dovrai prima abilitare il pianificatore di eventi da un gruppo di parametri database impostando il parametro `event_scheduler` su `ON`. Anche per le istanze database MySQL su cui è in esecuzione MySQL 5.7, imposta il parametro `show_compatibility_56` su 1. Per informazioni sulla creazione e la modifica di un gruppo di parametri database, consulta Utilizzo di gruppi di parametri. Per informazioni sugli effetti collaterali dell'abilitazione di questo parametro, consulta show_compatibility_56 nel Manuale di riferimento di MySQL 5.7.

Puoi utilizzare le procedure riportate nella seguente tabella per abilitare e configurare la funzione GoSH. Connettiti innanzitutto alla tua istanza database MySQL, quindi utilizza i comandi appropriati come mostrato di seguito. Per ulteriori informazioni, consulta Connessione a un'istanza database che esegue il motore di database di MySQL. Per ogni procedura, digita:

```
CALL procedure-name;
```

Dove procedure-name è una delle procedure nella tabella.

Procedura	Descrizione
`mysql.rds_enable_gsh_collector`	Abilita la funzione GoSH per acquisire le snapshot per impostazione predefinita a intervalli specificati da `rds_set_gsh_collector` .
`mysql.rds_set_gsh_collector`	Specifica l'intervallo, in minuti, tra gli snapshot. Il valore predefinito è 5.
`mysql.rds_disable_gsh_collector`	Disabilita gli snapshot.
`mysql.rds_collect_global_status_history`	Acquisisce una snapshot a richiesta.
`mysql.rds_enable_gsh_rotation`	Abilita la rotazione dei contenuti della tabella `mysql.rds_global_status_history` su `mysql.rds_global_status_history_old` a intervalli specificati da `rds_set_gsh_rotation` .
`mysql.rds_set_gsh_rotation`	Specifica l'intervallo, in giorni, tra le conversioni delle tabelle. Il valore predefinito è 7.
`mysql.rds_disable_gsh_rotation`	Disabilita la rotazione delle tabelle.
`mysql.rds_rotate_global_status_history`	Ruota i contenuti della tabella `mysql.rds_global_status_history` su `mysql.rds_global_status_history_old` a richiesta.

Quando la funzione GoSH è in esecuzione, puoi inviare query alle tabelle su cui viene completata la scrittura. Ad esempio, per inviare una query per il numero di riscontri del buffer pool InnoDB, dovresti inviare la seguente query:

```
select a.collection_end, a.collection_start, (( a.variable_Delta-b.variable_delta)/
a.variable_delta)*100 as "HitRatio"
    from mysql.rds_global_status_history as a join mysql.rds_global_status_history as b
 on a.collection_end = b.collection_end
```

```
    where a. variable_name = 'Innodb_buffer_pool_read_requests' and b.variable_name =
'Innodb_buffer_pool_reads'
```

Fuso orario locale per le istanze database MySQL

Per impostazione predefinita, il fuso orario per un'istanza database MySQL è in formato Universal Time Coordinated (UTC). Puoi impostare il fuso orario per l'istanza database sul fuso orario locale dell'applicazione.

Per impostare il fuso orario locale per un'istanza database, imposta il parametro `time_zone` nel gruppo di parametri per l'istanza database su uno dei valori supportati elencati più avanti in questa sezione. Quando imposti il parametro `time_zone` per un gruppo di parametri, tutte le istanze database e le repliche di lettura che utilizzano tale gruppo di parametri cambiano per utilizzare il nuovo fuso orario locale. Per informazioni sull'impostazione dei parametri in un gruppo di parametri, consulta Utilizzo di gruppi di parametri.

Dopo aver impostato il fuso orario locale, tutte le nuove connessioni al database riflettono la modifica. Se ci sono connessioni aperte al database quando modifichi il fuso orario locale, questo non viene aggiornato fino a quando non chiudi la connessione e ne apri una nuova.

Puoi impostare un fuso orario locale diverso per un'istanza database e una o più delle relative repliche di lettura. A tale scopo, utilizza un gruppo di parametri diverso per l'istanza database e la replica o le repliche e imposta il parametro `time_zone` in ogni gruppo di parametri su un fuso orario locale diverso.

Se esegui la replica tra Regioni AWS, l'istanza database di origine e la replica di lettura utilizzano gruppi di parametri diversi (i gruppi di parametri sono univoci per una Regione AWS). Per utilizzare lo stesso fuso orario locale per ogni istanza, imposta il parametro `time_zone` nei gruppi di parametri dell'istanza e della replica di lettura.

Quando ripristini un'istanza database da uno snapshot DB, il fuso orario locale è impostato su UTC. Puoi aggiornare il fuso orario impostandolo sul fuso orario locale dopo il completamento del ripristino. Se ripristini un'istanza database a un punto nel tempo, il fuso orario locale per l'istanza database ripristinata corrisponde all'impostazione del fuso orario per il gruppo di parametri dell'istanza database ripristinata.

Internet Assigned Numbers Authority (IANA) pubblica nuovi fusi orari all'indirizzo https://www.iana.org/time-zones più volte all'anno. Ogni volta che RDS rilascia una nuova versione di manutenzione secondaria di MySQL, la versione viene fornita con i dati sul fuso orario più recenti al momento del rilascio. Quando utilizzi le versioni più recenti di RDS per MySQL, hai a disposizione i dati recenti relativi ai fusi orari di RDS. Per assicurarti che l'istanza DB disponga dei dati più aggiornati relativi ai fusi orari, ti consigliamo di eseguire l'aggiornamento a una versione superiore

del motore DB. In alternativa, puoi modificare manualmente le tabelle dei fusi orari nelle istanze DB MariaDB. A tale scopo, puoi utilizzare i comandi SQL o eseguire lo strumento mysql_tzinfo_to_sql in un client SQL. Dopo l'aggiornamento manuale dei dati dei fusi orari, avvia l'istanza database per applicare le modifiche. RDS non modifica né ripristina i dati dei fusi orari delle istanze DB in esecuzione. I nuovi dati dei fusi orari vengono installati solo quando si esegue un aggiornamento della versione del motore di database.

Puoi impostare il fuso orario locale su uno dei valori seguenti.

Africa/Cairo	Asia/Riyadh
Africa/Casablanca	Asia/Seoul
Africa/Harare	Asia/Shanghai
Africa/Monrovia	Asia/Singapore
Africa/Nairobi	Asia/Taipei
Africa/Tripoli	Asia/Tehran
Africa/Windhoek	Asia/Tokyo
America/Araguaina	Asia/Ulaanbaatar
America/Asuncion	Asia/Vladivostok
America/Bogota	Asia/Yakutsk
America/Buenos_Aires	Asia/Yerevan
America/Caracas	Atlantic/Azores
America/Chihuahua	Australia/Adelaide
America/Cuiaba	Australia/Brisbane
America/Denver	Australia/Darwin
America/Fortaleza	Australia/Hobart

America/Guatemala	Australia/Perth
America/Halifax	Australia/Sydney
America/Manaus	Brazil/East
America/Matamoros	Canada/Newfoundland
America/Monterrey	Canada/Saskatchewan
America/Montevideo	Canada/Yukon
America/Phoenix	Europe/Amsterdam
America/Santiago	Europe/Athens
America/Tijuana	Europe/Dublin
Asia/Amman	Europe/Helsinki
Asia/Ashgabat	Europe/Istanbul
Asia/Baghdad	Europe/Kaliningrad
Asia/Baku	Europe/Moscow
Asia/Bangkok	Europe/Paris
Asia/Beirut	Europe/Prague
Asia/Calcutta	Europe/Sarajevo
Asia/Damascus	Pacific/Auckland
Asia/Dhaka	Pacific/Fiji
Asia/Irkutsk	Pacific/Guam
Asia/Jerusalem	Pacific/Honolulu
Asia/Kabul	Pacific/Samoa

Asia/Karachi	US/Alaska
Asia/Kathmandu	US/Central
Asia/Krasnoyarsk	US/Eastern
Asia/Magadan	US/East-Indiana
Asia/Muscat	US/Pacific
Asia/Novosibirsk	UTC

Problemi e limitazioni note per Amazon RDS for MySQL

I problemi e le limitazioni note per l'utilizzo di Amazon RDS for MySQL sono sotto riportarti.

Argomenti

Parola riservata InnoDB

`InnoDB` è una parola riservata per RDS per MySQL. Non è possibile utilizzare questo nome per un database MySQL.

Comportamento in caso di storage pieno per Amazon RDS for MySQL

Quando lo storage diventa pieno per un'istanza di database MySQL, possono verificarsi delle incongruenze di metadati, disallineamenti del dizionario e tabelle orfane. Per evitare questi problemi, Amazon RDS interrompe automaticamente un'istanza database che raggiunge lo stato `storage-full`.

Un'istanza MySQL database raggiunge lo stato `storage-full` nei seguenti casi:

- L'istanza database ha meno di 20.000 MiB di storage e lo storage disponibile raggiunge almeno 200 MiB.

- L'istanza database ha più di 102.400 MiB di storage e lo storage disponibile raggiunge 1024 MiB o meno.
- L'istanza database ha tra 20.000 MiB e 102.400 MiB di storage e ha meno dell'1% di storage disponibile.

Dopo che Amazon RDS interrompe automaticamente un'istanza database perché ha raggiunto lo stato `storage-full`, è comunque possibile modificarla. Per riavviare l'istanza database, completa almeno una delle seguenti operazioni:

- Modifica l'istanza database per abilitare il dimensionamento automatico dello storage.

  Per ulteriori informazioni sul dimensionamento automatico dello storage, consulta Gestione della capacità automaticamente con Auto Scaling dello storage Amazon RDS.
- Modifica l'istanza database per aumentarne la capacità di storage.

  Per ulteriori informazioni sull'aumento della capacità di storage, consulta Aumento della capacità di storage dell'istanza database.

Dopo aver apportato una di queste modifiche, l'istanza database viene riavviata automaticamente. Per ulteriori informazioni sulla modifica di un'istanza di database, consulta Modifica di un'istanza database Amazon RDS.

Dimensione del pool di buffer Innodb incoerente

Per MySQL 5.7, esiste attualmente un bug nel modo in cui viene gestita la dimensione del pool di buffer InnoDB. MySQL 5.7 potrebbe regolare il valore del parametro `innodb_buffer_pool_size` in un valore maggiore che possa far sì che il pool di buffer InnoDB diventi troppo grande e utilizzi troppa memoria. Questo effetto può causare l'arresto del motore database MySQL o impedirne l'avvio. Questo problema è più comune per le classi di istanze database con meno memoria disponibile.

Per risolvere questo problema, impostare il valore del parametro `innodb_buffer_pool_size` su un multiplo del prodotto del valore del parametro `innodb_buffer_pool_instances` e del valore del parametro `innodb_buffer_pool_chunk_size`. Ad esempio, si potrebbe impostare il valore del parametro `innodb_buffer_pool_size` su un multiplo di otto volte del prodotto del valore del parametro `innodb_buffer_pool_instances` e `innodb_buffer_pool_chunk_size`, come mostrato nel seguente esempio.

```
innodb_buffer_pool_chunk_size = 536870912
innodb_buffer_pool_instances = 4
innodb_buffer_pool_size = (536870912 * 4) * 8 = 17179869184
```

Per informazioni dettagliate su questo bug di MySQL 5.7, consulta https://bugs.mysql.com/bug.php?id=79379 nella documentazione di MySQL.

L'ottimizzazione dell'unione dell'indice restituisce risultati errati

Le query che usano l'ottimizzazione dell'unione dell'indice possono restituire risultati errati a causa di un bug nello strumento Query Optimizer MySQL introdotto in MySQL 5.5.37. Quando esegui una query su una tabella con più indici, lo strumento di ottimizzazione analizza gli intervalli di riga in base a più indici, ma non unisce i risultati correttamente. Per ulteriori informazioni sul bug di Query Optimizer, consulta http://bugs.mysql.com/bug.php?id=72745 e http://bugs.mysql.com/bug.php?id=68194 nel database dei bug di MySQL.

Ad esempio, si consideri una query su una tabella con due indici in cui gli argomenti di ricerca fanno riferimento alle colonne indicizzate.

```
SELECT * FROM table1
WHERE indexed_col1 = 'value1' AND indexed_col2 = 'value2';
```

In questo caso, la ricerca verrà effettuata su entrambi gli indici. Tuttavia, a causa del bug, i risultati uniti non sono corretti.

Per risolvere il problema, è possibile procedere in uno dei seguenti modi:

- Imposta il parametro `optimizer_switch` su `index_merge=off` nel gruppo di parametri di database per l'istanza di database MySQL. Per informazioni sull'impostazione dei parametri appartenenti a un gruppo di parametri di database, consulta Utilizzo di gruppi di parametri.
- Aggiorna l'istanza database MySQL a MySQL versione 5.7 o 8.0. Per ulteriori informazioni, consulta Aggiornamento del motore di database MySQL.
- Se non riesci ad aggiornare l'istanza o a modificare il parametro `optimizer_switch`, puoi aggirare il bug identificando in modo esplicito un indice per la query, ad esempio:

  ```
  SELECT * FROM table1
  USE INDEX covering_index
  WHERE indexed_col1 = 'value1' AND indexed_col2 = 'value2';
  ```

Per ulteriori informazioni, consulta Index merge optimization nella documentazione di MySQL.

Eccezioni dei parametri di MySQL per le istanze database Amazon RDS

Alcuni parametri MySQL richiedono considerazioni speciali quando vengono utilizzati con un'istanza database Amazon RDS.

lower_case_table_names

Poiché Amazon RDS usa un file system che fa distinzione tra maiuscole e minuscole, l'impostazione del valore del parametro server `lower_case_table_names` su 2 (nomi archiviati come dati ma confrontati in lettere minuscole) non è supportata. Di seguito sono riportati i valori supportati per le istanze database Amazon RDS for MySQL:

- 0 (nomi archiviati come dati e confronti con distinzione tra minuscole e maiuscole) è supportato per tutte le versioni di per MySQL.
- 1 (nomi memorizzati in minuscolo e confronti non fanno distinzione tra maiuscole e minuscole) è supportato per RDS per MySQL versione 5.7 e 8.0.28 e versioni successive alla 8.0.

Prima di creare un'istanza database, il parametro `lower_case_table_names` dovrebbe essere impostato come parte di un gruppo di parametri database personalizzato. Quindi, specificare il gruppo di parametri database personalizzato quando viene creata l'istanza database.

Quando un gruppo di parametri è associato a un'istanza database MySQL con una versione inferiore a 8.0, si consiglia di evitare di modificare il parametro `lower_case_table_names` nel gruppo di parametri. La modifica potrebbe causare incongruenze nei backup di point-in-time ripristino e nella lettura delle istanze DB di replica.

Quando un gruppo di parametri è associato a un'istanza database MySQL versione 8.0, non è possibile modificare il parametro `lower_case_table_names` nel gruppo di parametri.

Le repliche di lettura dovrebbero sempre usare lo stesso valore del parametro `lower_case_table_names` in qualità di istanza DB di origine.

long_query_time

È possibile impostare il parametro `long_query_time` su un valore del punto variabile che consenta di registrare query lente nel log delle query lente MySQL con risoluzione al microsecondo. È possibile impostare un valore come 0,1 secondi, che sarebbe 100 millisecondi, per aiutare durante il debug delle transazioni lente che richiedono meno di un secondo.

Limiti delle dimensioni dei file MySQL in Amazon RDS

Per le istanze DB MySQL, il limite massimo di archiviazione assegnato limita la dimensione di una tabella a una dimensione massima di 16 TB quando si utilizzano tablespace InnoDB. file-per-table Questo valore limita anche il tablespace di sistema a una dimensione massima di 16 TB. I file-per-table tablespace InnoDB (con tabelle ciascuna nel proprio tablespace) sono impostati di default per le istanze DB MySQL.

> **Note**
>
> Alcune istanze database esistenti hanno un limite inferiore. Ad esempio, le istanze database MySQL create prima di aprile 2014 hanno un limite di dimensioni di file e tabelle di 2 TB. Questo limite di 2 TB delle dimensioni si applica anche alle istanze database o alle repliche di lettura create da snapshot DB effettuati prima di aprile 2014, a prescindere dalla data di creazione dell'istanza database.

L'utilizzo dei file-per-table tablespace InnoDB presenta vantaggi e svantaggi, a seconda dell'applicazione. Per determinare l'approccio migliore per la tua applicazione, consulta i ile-per-table tablespace F nella documentazione di MySQL.

Non è consigliabile consentire alle tabelle di crescere fino alla dimensione massima del file. In generale, una pratica migliore consiste nel partizionare i dati in tabelle più piccole, che possano migliorare le prestazioni e i tempi di ripristino.

Un'opzione che è possibile utilizzare per suddividere una tabella di grandi dimensioni in tabelle più piccole è rappresentata dal partizionamento. Il partizionamento distribuisce porzioni della tabella di grandi dimensioni in file separati in base alle regole specificate. Ad esempio, se si archiviano le transazioni per data, è possibile creare regole di partizionamento che distribuiscono le transazioni meno recenti in file separati mediante il partizionamento. Quindi, periodicamente, è possibile archiviare i dati storici della transazione che non devono essere prontamente disponibili per l'applicazione. Per ulteriori informazioni, consulta Partitioning nella documentazione MySQL.

Poiché non esiste un'unica tabella o vista di sistema che fornisca le dimensioni di tutte le tabelle e dello spazio tabella del sistema InnoDB, è necessario eseguire query su più tabelle per determinare la dimensione degli spazi tabella.

Determinazione della dimensione dello spazio tabella del sistema InnoDB e dello spazio tabella del dizionario dati

- Utilizza il seguente comando SQL per stabilire se qualche spazio tabella supera le dimensioni consentite e può essere scelta per il partizionamento.

> **Note**
>
> Lo spazio tabella del dizionario dati è specifico di MySQL 8.0.

```
select FILE_NAME,TABLESPACE_NAME, ROUND(((TOTAL_EXTENTS*EXTENT_SIZE)
/1024/1024/1024), 2)  as "File Size (GB)" from information_schema.FILES
where tablespace_name in ('mysql','innodb_system');
```

Determinazione della dimensione delle tabelle utente di InnoDB al di fuori dello spazio tabella del sistema InnoDB (per le versioni 5.7 di MySQL)

- Utilizza il seguente comando SQL per stabilire se qualche tabella supera le dimensioni consentite e può essere scelta per il partizionamento.

```
SELECT SPACE,NAME,ROUND((ALLOCATED_SIZE/1024/1024/1024), 2)
as "Tablespace Size (GB)"
FROM information_schema.INNODB_SYS_TABLESPACES ORDER BY 3 DESC;
```

Determinazione della dimensione delle tabelle utente di InnoDB al di fuori dello spazio tabella del sistema InnoDB (per le versioni 8.0 di MySQL)

- Utilizza il seguente comando SQL per stabilire se qualche tabella supera le dimensioni consentite e può essere scelta per il partizionamento.

```
SELECT SPACE,NAME,ROUND((ALLOCATED_SIZE/1024/1024/1024), 2)
as "Tablespace Size (GB)"
FROM information_schema.INNODB_TABLESPACES ORDER BY 3 DESC;
```

Determinazione della dimensione delle tabelle utente non InnoDB

- Utilizza il seguente comando SQL per stabilire se qualche tabella utente non InnoDB ha dimensioni eccessive.

```
SELECT TABLE_SCHEMA, TABLE_NAME, round(((DATA_LENGTH + INDEX_LENGTH+DATA_FREE)
/ 1024 / 1024/ 1024), 2) As "Approximate size (GB)" FROM information_schema.TABLES
WHERE TABLE_SCHEMA NOT IN ('mysql', 'information_schema', 'performance_schema')
and ENGINE<>'InnoDB';
```

Per abilitare i tablespace InnoDB file-per-table

- Imposta il parametro innodb_file_per_table su 1 nel gruppo di parametri dell'istanza database.

Per disabilitare i tablespace InnoDB file-per-table

- Imposta il parametro innodb_file_per_table su `0` nel gruppo di parametri dell'istanza database.

Per informazioni sull'aggiornamento di un gruppo di parametri database, consulta Utilizzo di gruppi di parametri.

Dopo aver abilitato o disabilitato file-per-table i tablespace InnoDB, puoi emettere un `ALTER TABLE` comando per spostare una tabella dal tablespace globale al proprio tablespace o dal proprio tablespace al tablespace globale come mostrato nell'esempio seguente:

```
ALTER TABLE table_name ENGINE=InnoDB;
```

Plugin Keyring MySQL non supportato

Al momento, Amazon RDS per MySQL non supporta il plugin Keyring Amazon Web Services MySQL `keyring_aws`.

Porte personalizzate

Amazon RDS blocca le connessioni alla porta personalizzata 33060 per il motore MySQL. Scegli una porta diversa per il motore MySQL.

Limitazioni delle stored procedure di MySQL

Le stored procedure mysql.rds_kill e mysql.rds_kill_query non possono terminare sessioni o query di proprietà di utenti MySQL con nomi utente più lunghi di 16 caratteri nelle seguenti versioni di RDS per MySQL:

- 8.0.32 e versioni precedenti alla 8
- 5.7.41 e versioni precedenti alla 5.7

Replica basata su GTID con un'istanza di origine esterna

Amazon RDS non supporta la replica basata su identificatori di transazione globali (GTID) da un'istanza MySQL esterna in un'istanza database Amazon RDS per MySQL che richiede l'impostazione del parametro GTID_PURGED durante la configurazione.

Plugin di autenticazione MySQL predefinito

RDS per MySQL versione 8.0.34 e successive utilizza il plugin. `mysql_native_password` Non è possibile modificare l'impostazione `default_authentication_plugin`.

Sovrascrivere innodb_buffer_pool_size

Con classi di istanze DB micro o piccole, il valore predefinito per il `innodb_buffer_pool_size` parametro potrebbe differire dal valore restituito eseguendo il comando seguente:

```
mysql> SELECT @@innodb_buffer_pool_size;
```

Questa differenza può verificarsi quando Amazon RDS deve sovrascrivere il valore predefinito come parte della gestione delle classi di istanze DB. Se necessario, puoi sovrascrivere il valore predefinito e impostarlo su un valore supportato dalla classe dell'istanza DB. Per determinare un valore valido, aggiungi l'utilizzo della memoria e la memoria totale disponibile sull'istanza DB. Per ulteriori informazioni, consulta i tipi di istanze di Amazon RDS.

Se la tua istanza DB ha solo 4 GB di memoria, non puoi `innodb_buffer_pool_size` impostarla su 8 GB ma potresti essere in grado di impostarla su 3 GB, a seconda della quantità di memoria allocata per altri parametri.

Se il valore immesso è troppo grande, Amazon RDS lo riduce ai seguenti limiti:

- Classi di istanze Micro DB: 256 MB
- classi di istanze db.t4g.micro DB: 128 MB

Riferimento delle stored procedure RDS per MySQL

In questi argomenti vengono descritte le stored procedure di sistema disponibili per le istanze Amazon RDS che eseguono il motore di database MySQL. La procedura deve essere eseguita dall'utente master.

Argomenti

Configurazione

Le seguenti stored procedure impostano e mostrano i parametri di configurazione, ad esempio per la conservazione dei file di log binari.

Argomenti

- mysql.rds_set_configuration
- mysql.rds_show_configuration

mysql.rds_set_configuration

Specifica il numero di ore di conservazione dei log binari o il numero di secondi di ritardo della replica.

Sintassi

```
CALL mysql.rds_set_configuration(name,value);
```

Parametri

name

Il nome del parametro di configurazione da impostare.

value

Il valore del parametro di configurazione.

Note per l'utilizzo

la procedura archiviata `mysql.rds_set_configuration` supporta i parametri di configurazione seguenti:

- binlog retention hours
- Ritardo dell'origine
- target delay

I parametri di configurazione vengono archiviati in modo permanente e restano effettivi dopo qualsiasi riavvio o failover dell'istanza database.

binlog retention hours

Il parametro `binlog retention hours` viene utilizzato per specificare il numero di ore di conservazione dei file di log binari. Amazon RDS elimina in genere un log binario non appena possibile, tuttavia il log potrebbe continuare a essere necessario per la replica con un database MySQL esterno a RDS.

Il valore predefinito di `binlog retention hours` è `NULL`. Per RDS per MySQL, `NULL` significa che i log binari non vengono mantenuti (0 ore).

Per specificare il numero di ore per mantenere i log binari in un'istanza, usa la stored procedure `mysql.rds_set_configuration` e specifica un periodo con tempo sufficiente per l'esecuzione della replica, come mostrato nell'esempio seguente.

```
call mysql.rds_set_configuration('binlog retention hours', 24);
```

> **Note**
>
> Non puoi utilizzare il valore `0` per `binlog retention hours`.

Per le istanze database MySQL, il valore `binlog retention hours` massimo è 168 (7 giorni).

Dopo l'impostazione del periodo di retention, monitora l'utilizzo dello storage per l'istanza database per verificare che i log binari conservati non occupino troppo spazio di storage.

Ritardo dell'origine

Usa il parametro `source delay` in una replica di lettura per specificare il numero di secondi per cui ritardare la replica di lettura rispetto all'istanza database di origine. Amazon RDS in genere replica le modifiche non appena possibile, ma in alcuni ambienti è possibile che la replica venga ritardata. Ritardando la replica, ad esempio, è possibile effettuare il roll forward di una replica di lettura ritardata al momento immediatamente precedente a un errore. Se una tabella viene eliminata accidentalmente, puoi usare la replica ritardata per recuperarla rapidamente. Il valore predefinito di `target delay` è `0` (la replica non viene ritardata).

Quando utilizzato, questo parametro esegue mysql.rds_set_source_delay e applica CHANGE primary TO MASTER_DELAY = valore di input. In caso di esito positivo, la procedura salva il parametro `source delay` nella tabella `mysql.rds_configuration`.

Per specificare il numero di secondi per cui Amazon RDS deve ritardare la replica in un'istanza database di origine, usa la stored procedure `mysql.rds_set_configuration` e specifica il numero di secondi per il ritardo della replica. Nell'esempio seguente la replica viene ritardata di almeno un'ora (3600 secondi).

```
call mysql.rds_set_configuration('source delay', 3600);
```

La procedura quindi esegue `mysql.rds_set_source_delay(3600)`.

Il limite per il parametro `source delay` è un giorno (86400 secondi).

> **Note**
>
> Il parametro `source delay` non è supportato per RDS per MySQL versione 8.0 o MariaDB versioni precedenti alla 10.2.

target delay

Usa il parametro `target delay` per specificare il numero di secondi per cui ritardare la replica tra un'istanza database e le repliche di lettura future gestite da RDS create dall'istanza. Questo parametro viene ignorato per le repliche di lettura non gestite da RDS. Amazon RDS in genere replica le modifiche non appena possibile, ma in alcuni ambienti è possibile che la replica venga ritardata. Ritardando la replica, ad esempio, è possibile effettuare il roll forward di una replica di lettura ritardata al momento immediatamente precedente a un errore. Se una tabella viene eliminata accidentalmente, puoi usare la replica ritardata per recuperarla rapidamente. Il valore predefinito di `target delay` è `0` (la replica non viene ritardata).

Per il disaster recovery puoi usare questo parametro di configurazione con la stored procedure mysql.rds_start_replication_until o mysql.rds_start_replication_until_gtid. Per effettuare il roll forward delle modifiche a una replica di lettura ritardata al momento immediatamente precedente a un errore, puoi eseguire la procedura `mysql.rds_set_configuration` con questo parametro impostato. Dopo che la procedura `mysql.rds_start_replication_until` o `mysql.rds_start_replication_until_gtid` arresta la replica, puoi promuovere la replica di lettura come nuova istanza database master seguendo le istruzioni in Promozione di una replica di lettura a istanza database standalone.

Per usare la procedura `mysql.rds_rds_start_replication_until_gtid`, è necessario che sia abilitata la replica basata su GTID. Per passare a una specifica transazione

basata su GTID che notoriamente causa un problema, puoi usare la stored procedure mysql.rds_skip_transaction_with_gtid. Per ulteriori informazioni sull'utilizzo della replica basata su GTID, consulta Utilizzo della replica basata su GTID.

Per specificare il numero di secondi per cui Amazon RDS deve ritardare la replica in una replica di lettura, usa la stored procedure `mysql.rds_set_configuration` e specifica il numero di secondi per il ritardo della replica. L'esempio seguente specifica che la replica viene ritardata di almeno un'ora (3600 secondi).

```
call mysql.rds_set_configuration('target delay', 3600);
```

Il limite per il parametro `target delay` è un giorno (86400 secondi).

> **Note**
>
> Il parametro `target delay` non è supportato per RDS per MySQL versione 8.0 o MariaDB versioni precedenti alla 10.2.

mysql.rds_show_configuration

Il numero di ore di retention dei log binari.

Sintassi

```
CALL mysql.rds_show_configuration;
```

Note per l'utilizzo

Per verificare il numero di ore per cui Amazon RDS deve conservare i log binari, usa la stored procedure `mysql.rds_show_configuration`.

Esempi

L'esempio seguente visualizza il periodo di retention:

```
call mysql.rds_show_configuration;
                name                          value     description
                binlog retention hours        24        binlog retention hours specifies
 the duration in hours before binary logs are automatically deleted.
```

Terminare una sessione o una query

Le seguenti stored procedure terminano una sessione o una query.

Argomenti

- mysql.rds_kill
- mysql.rds_kill_query

mysql.rds_kill

Termina una connessione al server MySQL.

Sintassi

```
CALL mysql.rds_kill(processID);
```

Parametri

processID

L'identità del thread di connessione da terminare.

Note per l'utilizzo

Ogni connessione al server MySQL viene eseguita in un thread distinto. Per terminare una connessione, utilizza la procedura `mysql.rds_kill` e passa l'ID di thread di quella connessione. Per ottenere l'ID di thread, utilizza il comando MySQL SHOW PROCESSLIST.

Per informazioni sulle limitazioni, consulta Limitazioni delle stored procedure di MySQL.

Esempi

L'esempio seguente termina una connessione con l'ID di thread 4243:

```
CALL mysql.rds_kill(4243);
```

mysql.rds_kill_query

Termina una query in esecuzione sul server MySQL.

Sintassi

```
CALL mysql.rds_kill_query(processID);
```

Parametri

processID

L'identità del processo o del thread che esegue la query da terminare.

Note per l'utilizzo

Per arrestare una query in esecuzione nel server MySQL, utilizza la procedura `mysql_rds_kill_query` e invia l'ID di connessione del thread che sta eseguendo la query. La procedura termina quindi la connessione.

Per ottenere l'ID, esegui una query sulla tabella INFORMATION_SCHEMA PROCESSLIST MySQL o utilizza il comando MySQL SHOW PROCESSLIST. Il valore nella colonna ID da `SHOW PROCESSLIST` o `SELECT * FROM INFORMATION_SCHEMA.PROCESSLIST` è *processID*.

Per informazioni sulle limitazioni, consulta Limitazioni delle stored procedure di MySQL.

Esempi

L'esempio seguente arresta una query con l'ID di thread di query 230040:

```
CALL mysql.rds_kill_query(230040);
```

Registrazione

Le seguenti stored procedure ruotano i log MySQL nelle tabelle di backup. Per ulteriori informazioni, consulta File di log del database MySQL.

Argomenti

- mysql.rds_rotate_general_log
- mysql.rds_rotate_slow_log

mysql.rds_rotate_general_log

Converte la tabella `mysql.general_log` in una tabella di backup.

Sintassi

```
CALL mysql.rds_rotate_general_log;
```

Note per l'utilizzo

Puoi convertire la tabella `mysql.general_log` in una tabella di backup chiamando la procedura `mysql.rds_rotate_general_log`. Quando le tabelle di log sono convertite, la tabella di log corrente è copiata in una tabella di logo di backup e le voci nella tabella di log corrente sono eliminate. Se una tabella di log di backup esiste, viene eliminata prima che la tabella di log corrente sia copiata nel backup. Puoi eseguire una query sulla tabella di log di backup, se necessario. La tabella di log di backup per la tabella `mysql.general_log` è denominata `mysql.general_log_backup`.

È possibile eseguire questa procedura solo quando il parametro `log_output` è impostato su `TABLE`.

mysql.rds_rotate_slow_log

Converte la tabella `mysql.slow_log` in una tabella di backup.

Sintassi

```
CALL mysql.rds_rotate_slow_log;
```

Note per l'utilizzo

Puoi convertire la tabella `mysql.slow_log` in una tabella di backup chiamando la procedura `mysql.rds_rotate_slow_log`. Quando le tabelle di log sono convertite, la tabella di log corrente è copiata in una tabella di logo di backup e le voci nella tabella di log corrente sono eliminate. Se una tabella di log di backup esiste, viene eliminata prima che la tabella di log corrente sia copiata nel backup.

Puoi eseguire una query sulla tabella di log di backup, se necessario. La tabella di log di backup per la tabella `mysql.slow_log` è denominata `mysql.slow_log_backup`.

Gestione di cluster attivi-attivi

Le seguenti stored procedure configurano e gestiscono i cluster active-active di RDS per MySQL. Per ulteriori informazioni, consulta the section called “Configurazione di cluster attivi-attivi”.

Queste stored procedure sono disponibili solo con le istanze DB RDS for MySQL che eseguono la versione 8.0.35 e versioni secondarie successive.

Argomenti

mysql.rds_group_replication_advance_gtid

Crea GTID segnaposto sull'istanza DB corrente.

Sintassi

```
CALL mysql.rds_group_replication_advance_gtid(
begin_id
, end_id
, server_uuid
);
```

Parametri

begin_id

L'ID della transazione iniziale da creare.

end_id

L'ID della transazione finale da creare.

begin_id

Il `group_replication_group_name` per la transazione da creare. `group_replication_group_name`è specificato come UUID nel gruppo di parametri DB associato all'istanza DB.

Note per l'utilizzo

In un cluster attivo-attivo, affinché un'istanza DB possa entrare a far parte di un gruppo, tutte le transazioni GTID eseguite sulla nuova istanza DB devono esistere sugli altri membri del cluster. In casi insoliti, una nuova istanza DB potrebbe avere più transazioni quando le transazioni vengono eseguite prima di aggiungere l'istanza al gruppo. In questo caso, non puoi rimuovere alcuna transazione esistente, ma puoi utilizzare questa procedura per creare i GTID segnaposto corrispondenti sulle altre istanze DB del gruppo. Prima di farlo, verifica che le transazioni non influiscano sui dati replicati.

Quando si richiama questa procedura, le transazioni GTID di `server_uuid:begin_id-end_id` vengono create con contenuto vuoto. Per evitare problemi di replica, non utilizzate questa procedura in altre condizioni.

> **Important**
>
> Evita di chiamare questa procedura quando il cluster active-active funziona normalmente. Non richiamate questa procedura se non comprendete le possibili conseguenze delle transazioni che state creando. La chiamata a questa procedura potrebbe generare dati non coerenti.

Esempio

L'esempio seguente crea GTID segnaposto sull'istanza DB corrente. :

```
CALL mysql.rds_group_replication_advance_gtid(5, 6,
 '11111111-2222-3333-4444-555555555555');
```

mysql.rds_group_replication_create_user

Crea l'utente di replica `rdsgrprepladmin` per la replica di gruppo sull'istanza DB.

Sintassi

```
CALL mysql.rds_group_replication_create_user(
replication_user_password
);
```

Parametri

replication_user_password

La password dell'utente di replica. `rdsgrprepladmin`

Note per l'utilizzo

- La password dell'utente di replica `rdsgrprepladmin` deve essere la stessa su tutte le istanze DB di un cluster attivo-attivo.
- Il nome `rdsgrprepladmin` utente è riservato alle connessioni di replica di gruppo. Nessun altro utente, incluso l'utente principale, può avere questo nome utente.

Esempio

L'esempio seguente crea l'utente di replica `rdsgrprepladmin` per la replica di gruppo sull'istanza DB:

```
CALL mysql.rds_group_replication_create_user('password');
```

mysql.rds_group_replication_set_recovery_channel

Imposta il canale per un cluster `group_replication_recovery` attivo-attivo. La procedura utilizza l'`rdsgrprepladmin`utente riservato per configurare il canale.

Sintassi

```
CALL mysql.rds_group_replication_set_recovery_channel(
replication_user_password);
```

Parametri

replication_user_password

La password dell'utente `rdsgrprepladmin` di replica.

Note per l'utilizzo

La password dell'utente di replica `rdsgrprepladmin` deve essere la stessa su tutte le istanze DB di un cluster attivo-attivo. Una chiamata a specifica la password. `mysql.rds_group_replication_create_user`

Esempio

L'esempio seguente imposta il `group_replication_recovery` canale per un cluster attivo-attivo:

```
CALL mysql.rds_group_replication_set_recovery_channel('password');
```

mysql.rds_group_replication_start

Avvia la replica di gruppo sull'istanza DB corrente.

Sintassi

```
CALL mysql.rds_group_replication_start(
bootstrap
);
```

Parametri

bootstrap

Un valore che specifica se inizializzare un nuovo gruppo o unirsi a un gruppo esistente. `1`inizializza un nuovo gruppo con l'istanza DB corrente. `0`unisce l'istanza DB corrente a un gruppo esistente connettendosi agli endpoint definiti nel `group_replication_group_seeds` parametro nel gruppo di parametri DB associato all'istanza DB.

Esempio

L'esempio seguente inizializza un nuovo gruppo con l'istanza DB corrente:

```
CALL mysql.rds_group_replication_start(1);
```

mysql.rds_group_replication_stop

Interrompe la replica di gruppo sull'istanza DB corrente.

Sintassi

```
CALL mysql.rds_group_replication_stop();
```

Note per l'utilizzo

Quando si interrompe la replica su un'istanza DB, ciò non influisce su nessun'altra istanza DB nel cluster active-active.

Gestione della replica da più fonti

Le seguenti stored procedure configurano e gestiscono i canali di replica su una replica multi-sorgente RDS for MySQL. Per ulteriori informazioni, consulta the section called “Configurazione della replica da più fonti”.

Queste stored procedure sono disponibili solo con le istanze DB RDS for MySQL che eseguono le seguenti versioni del motore:

- 8.0.35 e versioni secondarie successive
- 5.7.44 e versioni secondarie successive

> **Note**
>
> Sebbene questa documentazione faccia riferimento alle istanze DB di origine come RDS per istanze DB MySQL, queste procedure funzionano anche per le istanze MySQL eseguite esternamente ad Amazon RDS.

Argomenti

mysql.rds_next_source_log_for_channel

Modifica la posizione del log dell'istanza DB di origine all'inizio del log binario successivo sull'istanza DB di origine per il canale. Utilizzate questa procedura solo se ricevete l'errore di I/O di replica 1236 su una replica da più fonti.

Sintassi

```
CALL mysql.rds_next_source_log_for_channel(
curr_master_log,
channel_name
);
```

Parametri

curr_master_log

L'indice del file di log di origine corrente. Ad esempio, se il file corrente è denominato `mysql-bin-changelog.012345`, l'indice è 12345. Per determinare il nome del file di log di origine corrente, esegui il comando `SHOW REPLICA STATUS FOR CHANNEL '`*channel_name*`'` e visualizza il campo `Source_Log_File`.

> **Note**
>
> Versioni precedenti di MySQL utilizzate `SHOW SLAVE STATUS` al posto di `SHOW REPLICA STATUS`. Se si utilizza una versione MySQL prima della 8.0.23, utilizzare `SHOW SLAVE STATUS`.

nome_canale

Il nome del canale di replica sulla replica multisorgente. Ogni canale di replica riceve gli eventi di log binario da un'unica istanza DB RDS for MySQL di origine in esecuzione su un host e una porta specifici.

Note per l'utilizzo

La procedura `mysql.rds_next_source_log_for_channel` deve essere eseguita dall'utente master. Se si verifica un errore IO_Thread, ad esempio, è possibile utilizzare questa procedura per

ignorare tutti gli eventi nel file di registro binario corrente e riprendere la replica dal file di registro binario successivo per il canale specificato in. `channel_name`

Esempio

Supponiamo che la replica fallisca su un canale su una replica con più sorgenti. L'esecuzione `SHOW REPLICA STATUS FOR CHANNEL 'channel_1'\G` sulla replica da più fonti restituisce il seguente risultato:

```
mysql> SHOW REPLICA STATUS FOR CHANNEL 'channel_1'\G
*************************** 1. row ***************************
             Replica_IO_State: Waiting for source to send event
                  Source_Host: myhost.XXXXXXXXXXXXXX.rr-rrrr-1.rds.amazonaws.com
                  Source_User: ReplicationUser
                  Source_Port: 3306
                Connect_Retry: 60
              Source_Log_File: mysql-bin-changelog.012345
          Read_Source_Log_Pos: 1219393
               Relay_Log_File: replica-relay-bin.000003
                Relay_Log_Pos: 30223388
        Relay_Source_Log_File: mysql-bin-changelog.012345
           Replica_IO_Running: No
          Replica_SQL_Running: Yes
              Replicate_Do_DB:.
              .
              .
                Last_IO_Errno: 1236
                Last_IO_Error: Got fatal error 1236 from master when reading data from
 binary log: 'Client requested master to start replication from impossible position;
 the first event 'mysql-bin-changelog.013406' at 1219393, the last event read from
 '/rdsdbdata/log/binlog/mysql-bin-changelog.012345' at 4, the last byte read from '/
rdsdbdata/log/binlog/mysql-bin-changelog.012345' at 4.'
               Last_SQL_Errno: 0
               Last_SQL_Error:
               .
               .
                 Channel_name: channel_1
              .
              .
 -- Some fields are omitted in this example output
```

Il campo `Last_IO_Errno` mostra che l'istanza riceve l'errore I/O 1236. Il campo `Source_Log_File` mostra che il nome di file è `mysql-bin-changelog.012345`, il che significa che l'indice del file di log è `12345`. Per risolvere l'errore, puoi chiamare `mysql.rds_next_source_log_for_channel` con i seguenti parametri:

```
CALL mysql.rds_next_source_log_for_channel(12345,'channel_1');
```

> **Note**
>
> Versioni precedenti di MySQL utilizzate `SHOW SLAVE STATUS` al posto di `SHOW REPLICA STATUS`. Se si utilizza una versione MySQL prima della 8.0.23, utilizzare `SHOW SLAVE STATUS`.

mysql.rds_reset_external_source_for_channel

Interrompe il processo di replica sul canale specificato e rimuove il canale e le configurazioni associate dalla replica da più fonti.

> **Important**
>
> Per eseguire questa procedura, è necessario abilitare `autocommit`. Per abilitarlo, impostare il parametro `autocommit` su 1. Per ulteriori informazioni sulla modifica dei parametri, consulta Modifica di parametri in un gruppo di parametri del database.

Sintassi

```
CALL mysql.rds_reset_external_source_for_channel (channel_name);
```

Parametri

`nome_canale`

Il nome del canale di replica sulla replica multisorgente. Ogni canale di replica riceve gli eventi di log binario da un'unica istanza DB RDS for MySQL di origine in esecuzione su un host e una porta specifici.

Note per l'utilizzo

La procedura `mysql.rds_reset_external_source_for_channel` deve essere eseguita dall'utente master. Questa procedura elimina tutti i log di inoltro che appartengono al canale da rimuovere.

mysql.rds_set_external_source_for_channel

Configura un canale di replica su un'istanza DB RDS for MySQL per replicare i dati da un'altra istanza DB RDS for MySQL.

> **Important**
>
> Per eseguire questa procedura, è necessario abilitare `autocommit`. Per abilitarlo, impostare il parametro `autocommit` su `1`. Per ulteriori informazioni sulla modifica dei parametri, consulta Modifica di parametri in un gruppo di parametri del database.

> **Note**
>
> È possibile utilizzare invece la the section called "mysql.rds_set_external_source_with_delay_for_channel" stored procedure per configurare questo canale con la replica ritardata.

Sintassi

```
CALL mysql.rds_set_external_source_for_channel (
  host_name
  , host_port
  , replication_user_name
  , replication_user_password
  , mysql_binary_log_file_name
  , mysql_binary_log_file_location
  , ssl_encryption
  , channel_name
);
```

Parametri

host_name

Il nome host o l'indirizzo IP dell'istanza DB di origine RDS for MySQL.

host_port

La porta utilizzata dall'istanza DB di origine RDS per MySQL. Se la configurazione della rete include la replica della porta Secure Shell (SSH) che converte il numero di porta, specifica il numero di porta esposto da SSH.

replication_user_name

L'ID di un utente con `REPLICATION CLIENT` e `REPLICATION SLAVE` le autorizzazioni sull'istanza DB di origine RDS for MySQL. Si consiglia di fornire un account utilizzato esclusivamente per la replica con l'istanza DB di origine.

replication_user_password

La password dell'ID utente specificata in `replication_user_name`.

mysql_binary_log_file_name

Il nome del log binario sull'istanza DB di origine che contiene le informazioni di replica.

mysql_binary_log_file_location

La posizione nel log binario `mysql_binary_log_file_name` a partire dalla quale la replica inizia a leggere le informazioni a essa relative.

È possibile determinare il nome e la posizione del file binlog eseguendolo `SHOW MASTER STATUS` sull'istanza DB di origine.

ssl_encryption

Un valore che specifica se la crittografia Secure Socket Layer (SSL) è utilizzata sulla connessione di replica. 1 indica che la crittografia SSL deve essere utilizzata; 0 specifica che la crittografia non deve essere utilizzata. Il valore predefinito è 0.

> **Note**
>
> L'opzione `MASTER_SSL_VERIFY_SERVER_CERT` non è supportata. Questa opzione è impostata su 0, il che significa che la connessione è crittografata, ma i certificati non sono verificati.

nome_canale

Il nome del canale di replica. Ogni canale di replica riceve gli eventi di log binario da un'unica istanza DB RDS for MySQL di origine in esecuzione su un host e una porta specifici.

Note per l'utilizzo

La procedura `mysql.rds_set_external_source_for_channel` deve essere eseguita dall'utente master. Questa procedura deve essere eseguita sull'istanza DB RDS for MySQL di destinazione su cui si sta creando il canale di replica.

Prima dell'esecuzione`mysql.rds_set_external_source_for_channel`, configura un utente di replica sull'istanza DB di origine con i privilegi richiesti per la replica da più fonti. Per connettere la replica multisorgente all'istanza DB di origine, è necessario specificare `replication_user_password` i valori di un utente di replica che dispone `REPLICATION CLIENT` delle autorizzazioni `replication_user_name` e delle autorizzazioni per l'istanza DB di origine. `REPLICATION SLAVE`

Per configurare un utente di replica sull'istanza DB di origine

1. Utilizzando il client MySQL di tua scelta, connettiti all'istanza DB di origine e crea un account utente da utilizzare per la replica. Di seguito è riportato un esempio.

   > **Important**
   >
   > Come procedura consigliata in materia di sicurezza, specificate una password diversa dal valore segnaposto mostrato negli esempi seguenti.

   MySQL 8.0

   ```
   CREATE USER 'repl_user'@'example.com' IDENTIFIED WITH mysql_native_password BY
    'password';
   ```

   MySQL 5.7

   ```
   CREATE USER 'repl_user'@'example.com' IDENTIFIED BY 'password';
   ```

2. Sull'istanza DB di origine, concedi `REPLICATION CLIENT` e `REPLICATION SLAVE` privilegi all'utente di replica. L'esempio seguente concede i privilegi `REPLICATION CLIENT` e `REPLICATION SLAVE` su tutti i database per l'utente "repl_user" del dominio:

```
GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user'@'example.com';
```

Per utilizzare la replica crittografata, configura l'istanza DB di origine per utilizzare connessioni SSL.

Dopo aver chiamato `mysql.rds_set_external_source_for_channel` per configurare questo canale di replica, puoi chiamare la replica per avviare il processo di replica mysql.rds_start_replication_for_channel sul canale. È possibile effettuare una chiamata the section called “mysql.rds_reset_external_source_for_channel” per interrompere la replica sul canale e rimuovere la configurazione del canale dalla replica.

Quando effettui una chiamata`mysql.rds_set_external_source_for_channel`, Amazon RDS registra l'ora, l'utente e un'azione `set channel source` nella `mysql.rds_history` tabella senza dettagli specifici del canale e nella `mysql.rds_replication_status` tabella, con il nome del canale. Queste informazioni vengono registrate solo per uso interno e scopi di monitoraggio. Per registrare l'intera procedura chiamata ai fini del controllo, prendete in considerazione la possibilità di abilitare i registri di controllo o i registri generali, in base ai requisiti specifici dell'applicazione.

Esempi

Quando viene eseguito su un'istanza DB RDS for MySQL, l'esempio seguente configura un canale di replica `channel_1` denominato su questa istanza DB per replicare i dati dall'origine specificata da host e porta. `sourcedb.example.com` `3306`

```
call mysql.rds_set_external_source_for_channel(
  'sourcedb.example.com',
  3306,
  'repl_user',
  'password',
  'mysql-bin-changelog.0777',
  120,
  0,
  'channel_1');
```

mysql.rds_set_external_source_with_auto_position_for_channel

Configura un canale di replica su un'istanza DB RDS for MySQL con un ritardo di replica opzionale. La replica si basa su identificatori di transazione globali (GTID).

> **Important**
>
> Per eseguire questa procedura, è necessario abilitare `autocommit`. Per abilitarlo, impostare il parametro `autocommit` su 1. Per ulteriori informazioni sulla modifica dei parametri, consulta Modifica di parametri in un gruppo di parametri del database.

Sintassi

```
CALL mysql.rds_set_external_source_with_auto_position_for_channel (
  host_name
  , host_port
  , replication_user_name
  , replication_user_password
  , ssl_encryption
  , delay
  , channel_name
);
```

Parametri

host_name

Il nome host o l'indirizzo IP dell'istanza DB di origine RDS for MySQL.

host_port

La porta utilizzata dall'istanza DB di origine RDS per MySQL. Se la configurazione della rete include la replica della porta Secure Shell (SSH) che converte il numero di porta, specifica il numero di porta esposto da SSH.

replication_user_name

L'ID di un utente con `REPLICATION CLIENT` e `REPLICATION SLAVE` le autorizzazioni sull'istanza DB di origine RDS for MySQL. Si consiglia di fornire un account utilizzato esclusivamente per la replica con l'istanza DB di origine.

replication_user_password

La password dell'ID utente specificata in `replication_user_name`.

ssl_encryption

Un valore che specifica se la crittografia Secure Socket Layer (SSL) è utilizzata sulla connessione di replica. 1 indica che la crittografia SSL deve essere utilizzata; 0 specifica che la crittografia non deve essere utilizzata. Il valore predefinito è 0.

> **Note**
>
> L'opzione `MASTER_SSL_VERIFY_SERVER_CERT` non è supportata. Questa opzione è impostata su 0, il che significa che la connessione è crittografata, ma i certificati non sono verificati.

delay

Il numero minimo di secondi per ritardare la replica dall'istanza DB di origine.

Il limite per questo parametro è un giorno (86400 secondi).

nome_canale

Il nome del canale di replica. Ogni canale di replica riceve gli eventi di log binario da un'unica istanza DB RDS for MySQL di origine in esecuzione su un host e una porta specifici.

Note per l'utilizzo

La procedura `mysql.rds_set_external_source_with_auto_position_for_channel` deve essere eseguita dall'utente master. Questa procedura deve essere eseguita sull'istanza DB RDS for MySQL di destinazione su cui si sta creando il canale di replica.

Prima dell'esecuzione`rds_set_external_source_with_auto_position_for_channel`, configura un utente di replica sull'istanza DB di origine con i privilegi richiesti per la replica da più fonti. Per connettere la replica multisorgente all'istanza DB di origine, è necessario specificare `replication_user_password` i valori di un utente di replica che dispone `REPLICATION CLIENT` delle autorizzazioni `replication_user_name` e delle autorizzazioni per l'istanza DB di origine. `REPLICATION SLAVE`

Per configurare un utente di replica sull'istanza DB di origine

1. Utilizzando il client MySQL di tua scelta, connettiti all'istanza DB di origine e crea un account utente da utilizzare per la replica. Di seguito è riportato un esempio.

   > **Important**
   >
   > Come procedura consigliata in materia di sicurezza, specificate una password diversa dal valore segnaposto mostrato negli esempi seguenti.

   MySQL 8.0

   ```
   CREATE USER 'repl_user'@'example.com' IDENTIFIED WITH mysql_native_password BY
    'password';
   ```

   MySQL 5.7

   ```
   CREATE USER 'repl_user'@'example.com' IDENTIFIED BY 'password';
   ```

2. Sull'istanza DB di origine, concedi `REPLICATION CLIENT` e `REPLICATION SLAVE` privilegi all'utente di replica. L'esempio seguente concede i privilegi `REPLICATION CLIENT` e `REPLICATION SLAVE` su tutti i database per l'utente "repl_user" del dominio:

   ```
   GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user'@'example.com';
   ```

Per utilizzare la replica crittografata, configura l'istanza DB di origine per utilizzare connessioni SSL.

Dopo aver chiamato `mysql.rds_set_external_source_with_auto_position_for_channel` per configurare un'istanza DB Amazon RDS come replica di lettura su un canale specifico, puoi chiamare la replica di lettura per avviare il processo di replica the section called “mysql.rds_start_replication_per_channel” su quel canale.

Dopo aver chiamato `mysql.rds_set_external_source_with_auto_position_for_channel` per configurare questo canale di replica, puoi chiamare la replica per avviare il processo di replica mysql.rds_start_replication_for_channel sul canale. È possibile effettuare una chiamata the section

called “mysql.rds_reset_external_source_for_channel” per interrompere la replica sul canale e rimuovere la configurazione del canale dalla replica.

Esempi

Quando viene eseguito su un'istanza DB RDS for MySQL, l'esempio seguente configura un canale di replica `channel_1` denominato su questa istanza DB per replicare i dati dall'origine specificata dall'`sourcedb.example.com`host e dalla `3306` porta. Imposta il ritardo minimo di replica a un'ora (3.600 secondi). Ciò significa che una modifica dall'istanza DB RDS for MySQL di origine non viene applicata alla replica multisorgente per almeno un'ora.

```
call mysql.rds_set_external_source_with_auto_position_for_channel(
  'sourcedb.example.com',
  3306,
  'repl_user',
  'password',
  0,
  3600,
  'channel_1');
```

mysql.rds_set_external_source_with_delay_for_channel

Configura un canale di replica su un'istanza DB RDS for MySQL con un ritardo di replica specificato.

> **Important**
>
> Per eseguire questa procedura, è necessario abilitare `autocommit`. Per abilitarlo, impostare il parametro `autocommit` su `1`. Per ulteriori informazioni sulla modifica dei parametri, consulta Modifica di parametri in un gruppo di parametri del database.

Sintassi

```
CALL mysql.rds_set_external_source_with_delay_for_channel (
  host_name
  , host_port
  , replication_user_name
  , replication_user_password
  , mysql_binary_log_file_name
```

```
  , mysql_binary_log_file_location
  , ssl_encryption
  , delay
  , channel_name
);
```

Parametri

host_name

Il nome host o l'indirizzo IP dell'istanza DB di origine RDS for MySQL.

host_port

La porta utilizzata dall'istanza DB di origine RDS per MySQL. Se la configurazione della rete include la replica della porta Secure Shell (SSH) che converte il numero di porta, specifica il numero di porta esposto da SSH.

replication_user_name

L'ID di un utente con `REPLICATION CLIENT` e `REPLICATION SLAVE` le autorizzazioni sull'istanza DB di origine RDS for MySQL. Si consiglia di fornire un account utilizzato esclusivamente per la replica con l'istanza DB di origine.

replication_user_password

La password dell'ID utente specificata in `replication_user_name`.

mysql_binary_log_file_name

Il nome del log binario sull'istanza DB di origine contiene le informazioni di replica.

mysql_binary_log_file_location

Posizione nel log binario `mysql_binary_log_file_name` a partire dalla quale la replica inizierà a leggere le informazioni di replica.

È possibile determinare il nome e la posizione del file binlog in esecuzione `SHOW MASTER STATUS` sull'istanza del database di origine.

ssl_encryption

Un valore che specifica se la crittografia Secure Socket Layer (SSL) è utilizzata sulla connessione di replica. 1 indica che la crittografia SSL deve essere utilizzata; 0 specifica che la crittografia non deve essere utilizzata. Il valore predefinito è 0.

> **Note**
>
> L'opzione `MASTER_SSL_VERIFY_SERVER_CERT` non è supportata. Questa opzione è impostata su 0, il che significa che la connessione è crittografata, ma i certificati non sono verificati.

`delay`

Il numero minimo di secondi per ritardare la replica dall'istanza DB di origine.

Il limite per questo parametro è un giorno (86400 secondi).

`nome_canale`

Il nome del canale di replica. Ogni canale di replica riceve gli eventi di log binario da un'unica istanza DB RDS for MySQL di origine in esecuzione su un host e una porta specifici.

Note per l'utilizzo

La procedura `mysql.rds_set_external_source_with_delay_for_channel` deve essere eseguita dall'utente master. Questa procedura deve essere eseguita sull'istanza DB RDS for MySQL di destinazione su cui si sta creando il canale di replica.

Prima dell'esecuzione`mysql.rds_set_external_source_with_delay_for_channel`, configura un utente di replica sull'istanza DB di origine con i privilegi richiesti per la replica da più fonti. Per connettere la replica multisorgente all'istanza DB di origine, è necessario specificare `replication_user_password` i valori di un utente di replica che dispone `REPLICATION CLIENT` delle autorizzazioni `replication_user_name` e delle autorizzazioni per l'istanza DB di origine. `REPLICATION SLAVE`

Per configurare un utente di replica sull'istanza DB di origine

1. Utilizzando il client MySQL di tua scelta, connettiti all'istanza DB di origine e crea un account utente da utilizzare per la replica. Di seguito è riportato un esempio.

   > **Important**
   >
   > Come procedura consigliata in materia di sicurezza, specificate una password diversa dal valore segnaposto mostrato negli esempi seguenti.

MySQL 8.0

```
CREATE USER 'repl_user'@'example.com' IDENTIFIED WITH mysql_native_password BY
 'password';
```

MySQL 5.7

```
CREATE USER 'repl_user'@'example.com' IDENTIFIED BY 'password';
```

2. Sull'istanza DB di origine, concedi `REPLICATION CLIENT` e `REPLICATION SLAVE` privilegi all'utente di replica. L'esempio seguente concede i privilegi `REPLICATION CLIENT` e `REPLICATION SLAVE` su tutti i database per l'utente "repl_user" del dominio:

```
GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user'@'example.com';
```

Per utilizzare la replica crittografata, configura l'istanza DB di origine per utilizzare connessioni SSL.

Dopo aver chiamato `mysql.rds_set_external_source_with_delay_for_channel` per configurare questo canale di replica, puoi chiamare la replica per avviare il processo di replica mysql.rds_start_replication_for_channel sul canale. È possibile effettuare una chiamata the section called “mysql.rds_reset_external_source_for_channel” per interrompere la replica sul canale e rimuovere la configurazione del canale dalla replica.

Quando effettui una chiamata`mysql.rds_set_external_source_with_delay_for_channel`, Amazon RDS registra l'ora, l'utente e un'azione `set channel source` nella `mysql.rds_history` tabella senza dettagli specifici del canale e nella `mysql.rds_replication_status` tabella, con il nome del canale. Queste informazioni vengono registrate solo per uso interno e scopi di monitoraggio. Per registrare l'intera procedura chiamata ai fini del controllo, prendete in considerazione la possibilità di abilitare i registri di controllo o i registri generali, in base ai requisiti specifici dell'applicazione.

Esempi

Quando viene eseguito su un'istanza DB RDS for MySQL, l'esempio seguente configura un canale di replica `channel_1` denominato su questa istanza DB per replicare i dati dall'origine specificata dall'`sourcedb.example.com`host e dalla `3306` porta. Imposta il ritardo minimo di replica a un'ora

(3.600 secondi). Ciò significa che una modifica dall'istanza DB RDS for MySQL di origine non viene applicata alla replica multisorgente per almeno un'ora.

```
call mysql.rds_set_external_source_with_delay_for_channel(
  'sourcedb.example.com',
  3306,
  'repl_user',
  'password',
  'mysql-bin-changelog.000777',
  120,
  0,
  3600,
  'channel_1');
```

mysql.rds_set_source_auto_position_for_channel

Imposta la modalità di replica per il canale specificato in modo che sia basata sulle posizioni dei file di registro binari o sugli identificatori di transazione globali (GTID).

Sintassi

```
CALL mysql.rds_set_source_auto_position_for_channel (
auto_position_mode
 , channel_name
);
```

Parametri

auto_position_mode

Valore che indica se usare la replica basata sulla posizione del file di log o la replica basata su GTID:

- `0` – Usa il metodo di replica basato sulla posizione del file di log binario. Il valore di default è `0`.
- 1 – Usa il metodo di replica basato su GTID.

nome_canale

Il nome del canale di replica sulla replica multisorgente. Ogni canale di replica riceve gli eventi di log binario da un'unica istanza DB RDS for MySQL di origine in esecuzione su un host e una porta specifici.

Note per l'utilizzo

La procedura `mysql.rds_set_source_auto_position_for_channel` deve essere eseguita dall'utente master. Questa procedura riavvia la replica sul canale specificato per applicare la modalità di posizionamento automatico specificata.

Esempi

L'esempio seguente imposta la modalità di posizionamento automatico per channel_1 per utilizzare il metodo di replica basato su GTID.

```
call mysql.rds_set_source_auto_position_for_channel(1,'channel_1');
```

mysql.rds_set_source_delay_for_channel

Imposta il numero minimo di secondi per ritardare la replica dall'istanza del database di origine alla replica multisorgente per il canale specificato.

Sintassi

```
CALL mysql.rds_set_source_delay_for_channel(delay, channel_name);
```

Parametri

`delay`

Il numero minimo di secondi per ritardare la replica dall'istanza DB di origine.

Il limite per questo parametro è un giorno (86400 secondi).

`nome_canale`

Il nome del canale di replica sulla replica multisorgente. Ogni canale di replica riceve gli eventi di log binario da un'unica istanza DB RDS for MySQL di origine in esecuzione su un host e una porta specifici.

Note per l'utilizzo

La procedura `mysql.rds_set_source_delay_for_channel` deve essere eseguita dall'utente master. Per utilizzare questa procedura, è necessario innanzitutto effettuare una chiamata `mysql.rds_stop_replication_for_channel` per interrompere la replica. Quindi, richiamate

questa procedura per impostare il valore del ritardo di replica. Quando il ritardo è impostato, chiama `mysql.rds_start_replication_for_channel` per riavviare la replica.

Esempi

L'esempio seguente imposta il ritardo per la replica dall'istanza del database `channel_1` di origine sulla replica multisorgente per almeno un'ora (3.600 secondi).

```
CALL mysql.rds_set_source_delay_for_channel(3600,'channel_1');
```

mysql.rds_skip_repl_error_for_channel

Ignora un evento di log binario ed elimina un errore di replica su una replica multisorgente MySQL DB per il canale specificato.

Sintassi

```
CALL mysql.rds_skip_repl_error_for_channel(channel_name);
```

Parametri

`nome_canale`

Il nome del canale di replica sulla replica multisorgente. Ogni canale di replica riceve gli eventi di log binario da un'unica istanza DB RDS for MySQL di origine in esecuzione su un host e una porta specifici.

Note per l'utilizzo

La procedura `mysql.rds_skip_repl_error_for_channel` deve essere eseguita dall'utente master su una replica di lettura. È possibile utilizzare questa procedura in modo `mysql.rds_skip_repl_error` simile a quello utilizzato per ignorare un errore in una replica di lettura. Per ulteriori informazioni, consulta Chiamata della procedura mysql.rds_skip_repl_error.

> **Note**
>
> Per ignorare gli errori nella replica basata su GTID, si consiglia di utilizzare invece la procedura. the section called “mysql.rds_skip_transaction_with_gtid”

Per determinare se ci sono errori, esegui il comando MySQL `SHOW REPLICA STATUS FOR CHANNEL 'channel_name'\G`. Se un errore di replica non è critico, puoi eseguire `mysql.rds_skip_repl_error_for_channel` per ignorare l'errore. Se sono presenti più errori, `mysql.rds_skip_repl_error_for_channel` elimina il primo errore sul canale di replica specificato, quindi avvisa che ne sono presenti altri. Puoi quindi utilizzare `SHOW REPLICA STATUS FOR CHANNEL 'channel_name'\G` per determinare l'operazione corretta per l'errore successivo. Per informazioni sui valori restituiti, consulta Istruzione SHOW REPLICA STATUS nella documentazione di MySQL.

mysql.rds_start_replication_for_channel

Avvia la replica da un'istanza DB RDS for MySQL a una replica multisorgente sul canale specificato.

> **Note**
>
> Puoi usare la stored procedure mysql.rds_start_replication_until_for_channel o mysql.rds_start_replication_until_gtid_for_channel per avviare la replica da un'istanza database RDS for MySQL e arrestare la replica in corrispondenza della posizione del file di log binario specificato.

Sintassi

```
CALL mysql.rds_start_replication_for_channel(channel_name);
```

Parametri

nome_canale

Il nome del canale di replica sulla replica multisorgente. Ogni canale di replica riceve gli eventi di log binario da un'unica istanza DB RDS for MySQL di origine in esecuzione su un host e una porta specifici.

Note per l'utilizzo

La procedura `mysql.rds_start_replication_for_channel` deve essere eseguita dall'utente master. Dopo aver importato i dati dall'istanza DB RDS for MySQL di origine, esegui questo comando sulla replica multi-source per avviare la replica sul canale specificato.

Esempi

L'esempio seguente avvia la replica sulla replica da più fonti. `channel_1`

```
CALL mysql.rds_start_replication_for_channel('channel_1');
```

mysql.rds_start_replication_until_for_channel

Avvia la replica da un'istanza DB RDS for MySQL sul canale specificato e interrompe la replica nella posizione specificata del file di log binario.

Sintassi

```
CALL mysql.rds_start_replication_until_for_channel (
replication_log_file
  , replication_stop_point
  , channel_name
);
```

Parametri

`replication_log_file`

Il nome del log binario sull'istanza DB di origine contiene le informazioni di replica.

`replication_stop_point`

Posizione nel log binario `replication_log_file` in corrispondenza di cui la replica verrà arrestata.

`nome_canale`

Il nome del canale di replica sulla replica multisorgente. Ogni canale di replica riceve gli eventi di log binario da un'unica istanza DB RDS for MySQL di origine in esecuzione su un host e una porta specifici.

Note per l'utilizzo

La procedura `mysql.rds_start_replication_until_for_channel` deve essere eseguita dall'utente master. Con questa procedura, la replica viene avviata e quindi interrotta quando viene

raggiunta la posizione del file binlog specificata. Per la versione 8.0, la procedura interrompe solo il. `SQL_Thread` Per la versione 5.7, la procedura arresta sia la che `SQL_Thread` la. `IO_Thread`

Il nome di file specificato per il `replication_log_file` parametro deve corrispondere al nome del file binlog dell'istanza DB di origine.

Quando il `replication_stop_point` parametro specifica una posizione di arresto che appartiene al passato, la replica viene interrotta immediatamente.

Esempi

L'esempio seguente avvia la replica e replica le modifiche fino a raggiungere la posizione `120` nel file di registro binario. `channel_1` `mysql-bin-changelog.000777`

```
call mysql.rds_start_replication_until_for_channel(
  'mysql-bin-changelog.000777',
  120,
  'channel_1'
  );
```

mysql.rds_start_replication_until_gtid_for_channel

Avvia la replica sul canale specificato da un'istanza DB RDS for MySQL e interrompe la replica in corrispondenza dell'identificatore di transazione globale (GTID) specificato.

Sintassi

```
CALL mysql.rds_start_replication_until_gtid_for_channel(gtid,channel_name);
```

Parametri

`gtid`

Il GTID dopo il quale interrompere la replica.

`nome_canale`

Il nome del canale di replica sulla replica multisorgente. Ogni canale di replica riceve gli eventi di log binario da un'unica istanza DB RDS for MySQL di origine in esecuzione su un host e una porta specifici.

Note per l'utilizzo

La procedura `mysql.rds_start_replication_until_gtid_for_channel` deve essere eseguita dall'utente master. La procedura avvia la replica sul canale specificato e applica tutte le modifiche fino al valore GTID specificato. Quindi, interrompe la replica sul canale.

Quando il parametro `gtid` specifica una transazione che è già stata eseguita dalla replica, la procedura viene arrestata immediatamente.

Prima di eseguire questa procedura, è necessario disabilitare la replica multithread impostando il valore di o su. `replica_parallel_workers` `slave_parallel_workers` `0`

Esempi

L'esempio seguente avvia la replica su `channel_1` e replica le modifiche fino a raggiungere GTID. `3E11FA47-71CA-11E1-9E33-C80AA9429562:23`

```
call mysql.rds_start_replication_until_gtid_for_channel('3E11FA47-71CA-11E1-9E33-
C80AA9429562:23','channel_1');
```

mysql.rds_stop_replication_for_channel

Interrompe la replica da un'istanza DB MySQL sul canale specificato.

Sintassi

```
CALL mysql.rds_stop_replication_for_channel(channel_name);
```

Parametri

`nome_canale`

Il nome del canale di replica sulla replica multisorgente. Ogni canale di replica riceve gli eventi di log binario da un'unica istanza DB RDS for MySQL di origine in esecuzione su un host e una porta specifici.

Note per l'utilizzo

La procedura `mysql.rds_stop_replication_for_channel` deve essere eseguita dall'utente master.

Esempi

L'esempio seguente interrompe la replica sulla `channel_1` replica da più fonti.

```
CALL mysql.rds_stop_replication_for_channel('channel_1');
```

Gestione della cronologia di stato globale

Amazon RDS fornisce una serie di procedure che acquisiscono uno snapshot dei valori di queste variabili di stato nel tempo e li scrivono in una tabella insieme alle modifiche eseguite dopo l'ultimo snapshot. Questa infrastruttura si chiama cronologia di stato globale. Per ulteriori informazioni, consulta Gestione della cronologia di stato globale.

Le seguenti stored procedure gestiscono il modo in cui la cronologia di stato globale viene raccolta e gestita.

Argomenti

mysql.rds_collect_global_status_history

Acquisisce uno snapshot on demand per la cronologia di stato globale.

Sintassi

```
CALL mysql.rds_collect_global_status_history;
```

mysql.rds_disable_gsh_collector

Disabilita gli snapshot creati mediante la cronologia di stato globale.

Sintassi

```
CALL mysql.rds_disable_gsh_collector;
```

mysql.rds_disable_gsh_rotation

Disattiva la rotazione della tabella `mysql.global_status_history`.

Sintassi

```
CALL mysql.rds_disable_gsh_rotation;
```

mysql.rds_enable_gsh_collector

Abilita la cronologia di stato globale per acquisire snapshot predefiniti agli intervalli specificati da `rds_set_gsh_collector`.

Sintassi

```
CALL mysql.rds_enable_gsh_collector;
```

mysql.rds_enable_gsh_rotation

Attiva la rotazione dei contenuti della tabella `mysql.global_status_history` su `mysql.global_status_history_old` agli intervalli specificati da `rds_set_gsh_rotation`.

Sintassi

```
CALL mysql.rds_enable_gsh_rotation;
```

mysql.rds_rotate_global_status_history

Ruota i contenuti della tabella `mysql.global_status_history` su `mysql.global_status_history_old` a richiesta.

Sintassi

```
CALL mysql.rds_rotate_global_status_history;
```

mysql.rds_set_gsh_collector

Specifica l'intervallo, espresso in minuti, tra gli snapshot acquisiti mediante la cronologia di stato globale.

Sintassi

```
CALL mysql.rds_set_gsh_collector(intervalPeriod);
```

Parametri

intervalPeriod

L'intervallo, in minuti, tra gli snapshot. Il valore predefinito è 5.

mysql.rds_set_gsh_rotation

Specifica l'intervallo, in giorni, tra le conversioni della tabella `mysql.global_status_history`.

Sintassi

```
CALL mysql.rds_set_gsh_rotation(intervalPeriod);
```

Parametri

intervalPeriod

L'intervallo, in giorni, tra le conversioni delle tabelle. Il valore predefinito è 7.

Replica

Queste stored procedure controllano il modo in cui le transazioni vengono replicate da un database esterno in RDS per MySQL o viceversa. Per informazioni su come utilizzare la replica in base agli ID globali di transazione (GTID) con RDS per MySQL, consulta Utilizzo della replica basata su GTID.

Argomenti

- mysql.rds_next_master_log
- mysql.rds_reset_external_master
- mysql.rds_set_external_master
- mysql.rds_set_external_master_with_auto_position
- mysql.rds_set_external_master_with_delay
- mysql.rds_set_master_auto_position
- mysql.rds_set_source_delay
- mysql.rds_skip_transaction_with_gtid
- mysql.rds_skip_repl_error
- mysql.rds_start_replication
- mysql.rds_start_replication_until
- mysql.rds_start_replication_until_gtid
- mysql.rds_stop_replication

mysql.rds_next_master_log

Cambia la posizione del log dell'istanza database di origine all'inizio del successivo log binario nell'istanza database di origine. Utilizza questa procedura solo se ricevi un errore I/O di replica 1236 su una replica di lettura.

Sintassi

```
CALL mysql.rds_next_master_log(
curr_master_log
);
```

Parametri

curr_master_log

L'indice del file di log master corrente. Ad esempio, se il file corrente è denominato `mysql-bin-changelog.012345`, l'indice è 12345. Per determinare il nome del file di log master corrente, esegui il comando `SHOW REPLICA STATUS` e visualizza il campo `Master_Log_File`.

> **Note**
>
> Versioni precedenti di MySQL utilizzate `SHOW SLAVE STATUS` al posto di `SHOW REPLICA STATUS`. Se si utilizza una versione MySQL prima della 8.0.23, utilizzare `SHOW SLAVE STATUS`.

Note per l'utilizzo

La procedura `mysql.rds_next_master_log` deve essere eseguita dall'utente master.

> **Warning**
>
> Chiama `mysql.rds_next_master_log` solo se la replica non riesce dopo un failover di un'istanza database Multi-AZ DB che è l'origine della replica e il campo `Last_IO_Errno` di `SHOW REPLICA STATUS` segnala l'errore I/O 1236.
> La chiamata di `mysql.rds_next_master_log` può comportare una perdita di dati nella replica di lettura se le transazioni nell'istanza di origine non sono state scritte nel log binario sul disco prima dell'evento di failover.
> Puoi ridurre le possibilità che si verifichi una situazione di questo tipo impostando i parametri dell'istanza di origine `sync_binlog` e `innodb_support_xa` su 1, anche se ciò può compromettere le prestazioni. Per ulteriori informazioni, consulta Risoluzione dei problemi relativi a una replica di lettura MySQL.

Esempi

Supponi che una replica di lettura RDS per MySQL non riesca. L'esecuzione di `SHOW REPLICA STATUS\G` nella replica di lettura restituisce il risultato seguente:

```
*************************** 1. row ***************************
             Replica_IO_State:
```

```
                  Source_Host: myhost.XXXXXXXXXXXXXXX.rr-rrrr-1.rds.amazonaws.com
                  Source_User: MasterUser
                  Source_Port: 3306
                Connect_Retry: 10
              Source_Log_File: mysql-bin-changelog.012345
          Read_Source_Log_Pos: 1219393
               Relay_Log_File: relaylog.012340
                Relay_Log_Pos: 30223388
        Relay_Source_Log_File: mysql-bin-changelog.012345
           Replica_IO_Running: No
          Replica_SQL_Running: Yes
              Replicate_Do_DB:
          Replicate_Ignore_DB:
           Replicate_Do_Table:
       Replicate_Ignore_Table:
      Replicate_Wild_Do_Table:
  Replicate_Wild_Ignore_Table:
                   Last_Errno: 0
                   Last_Error:
                 Skip_Counter: 0
          Exec_Source_Log_Pos: 30223232
              Relay_Log_Space: 5248928866
              Until_Condition: None
               Until_Log_File:
                Until_Log_Pos: 0
           Source_SSL_Allowed: No
           Source_SSL_CA_File:
           Source_SSL_CA_Path:
              Source_SSL_Cert:
            Source_SSL_Cipher:
               Source_SSL_Key:
        Seconds_Behind_Master: NULL
Source_SSL_Verify_Server_Cert: No
                Last_IO_Errno: 1236
                Last_IO_Error: Got fatal error 1236 from master when reading data from
 binary log: 'Client requested master to start replication from impossible position;
 the first event 'mysql-bin-changelog.013406' at 1219393, the last event read from
 '/rdsdbdata/log/binlog/mysql-bin-changelog.012345' at 4, the last byte read from '/
rdsdbdata/log/binlog/mysql-bin-changelog.012345' at 4.'
               Last_SQL_Errno: 0
               Last_SQL_Error:
  Replicate_Ignore_Server_Ids:
             Source_Server_Id: 67285976
```

Il campo `Last_IO_Errno` mostra che l'istanza riceve l'errore I/O 1236. Il campo `Master_Log_File` mostra che il nome di file è `mysql-bin-changelog.012345`, il che significa che l'indice del file di log è `12345`. Per risolvere il problema, puoi chiamare `mysql.rds_next_master_log` con il seguente parametro:

```
CALL mysql.rds_next_master_log(12345);
```

> **Note**
>
> Versioni precedenti di MySQL utilizzate `SHOW SLAVE STATUS` al posto di `SHOW REPLICA STATUS`. Se si utilizza una versione MySQL prima della 8.0.23, utilizzare `SHOW SLAVE STATUS`.

mysql.rds_reset_external_master

Riconfigura un'istanza database RDS per MySQL affinché non sia più una replica di lettura di un'istanza di MySQL in esecuzione all'esterno di Amazon RDS.

> **Important**
>
> Per eseguire questa procedura, è necessario abilitare `autocommit`. Per abilitarlo, impostare il parametro `autocommit` su `1`. Per ulteriori informazioni sulla modifica dei parametri, consulta Modifica di parametri in un gruppo di parametri del database.

Sintassi

```
CALL mysql.rds_reset_external_master;
```

Note per l'utilizzo

La procedura `mysql.rds_reset_external_master` deve essere eseguita dall'utente master. Questa procedura deve essere eseguita sull'istanza database MySQL da rimuovere come replica di lettura di un'istanza MySQL eseguita esternamente a Amazon RDS.

> **Note**
>
> Ti consigliamo di usare le repliche di lettura per gestire la replica tra due istanze database di Amazon RDS. In questo caso, si consiglia di usare solo questa e altre stored procedure correlate alla replica. Questo consente di usare topologie di replica più complesse tra le istanze database Amazon RDS. Queste stored procedure sono fornite principalmente per abilitare la replica con le istanze MySQL eseguite esternamente a Amazon RDS. Per ulteriori informazioni sulla gestione della replica tra istanze database Amazon RDS, consulta Uso delle repliche di lettura dell'istanza database.

Per ulteriori informazioni sull'uso della replica per importare dati da un'istanza di MySQL in esecuzione all'esterno di Amazon RDS, consulta Configurazione della replica della posizione del file di log binario con un'istanza di origine esterna..

mysql.rds_set_external_master

Configura un'istanza database RDS per MySQL come replica di lettura di un'istanza di MySQL in esecuzione all'esterno di Amazon RDS.

> **Important**
>
> Per eseguire questa procedura, è necessario abilitare `autocommit`. Per abilitarlo, impostare il parametro `autocommit` su 1. Per ulteriori informazioni sulla modifica dei parametri, consulta Modifica di parametri in un gruppo di parametri del database.

> **Note**
>
> Puoi usare la procedura archiviata mysql.rds_set_external_master_with_delay per configurare un'istanza database di origine esterna e una replica ritardata.

Sintassi

```
CALL mysql.rds_set_external_master (
  host_name
```

```
  , host_port
  , replication_user_name
  , replication_user_password
  , mysql_binary_log_file_name
  , mysql_binary_log_file_location
  , ssl_encryption
);
```

Parametri

host_name

Il nome host o l'indirizzo IP dell'istanza di MySQL eseguita esternamente a Amazon RDS per diventare l'istanza database di origine.

host_port

La porta utilizzata dall'istanza di MySQL eseguita esternamente a Amazon RDS e da configurare come istanza database di origine. Se la configurazione della rete include la replica della porta Secure Shell (SSH) che converte il numero di porta, specifica il numero di porta esposto da SSH.

replication_user_name

L'ID di un utente con autorizzazioni `REPLICATION CLIENT` e `REPLICATION SLAVE` nell'istanza di MySQL eseguita esternamente a Amazon RDS. Ti consigliamo di fornire un account utilizzato unicamente per la replica con l'istanza esterna.

replication_user_password

La password dell'ID utente specificata in `replication_user_name`.

mysql_binary_log_file_name

Il nome del log binario sull'istanza database di origine che contiene le informazioni relative alla replica.

mysql_binary_log_file_location

La posizione nel log binario `mysql_binary_log_file_name` a partire dalla quale la replica inizia a leggere le informazioni a essa relative.

È possibile determinare il nome e la posizione del file binlog in esecuzione `SHOW MASTER STATUS` sull'istanza del database di origine.

ssl_encryption

Un valore che specifica se la crittografia Secure Socket Layer (SSL) è utilizzata sulla connessione di replica. 1 indica che la crittografia SSL deve essere utilizzata; 0 specifica che la crittografia non deve essere utilizzata. Il valore predefinito è 0.

> **Note**
>
> L'opzione `MASTER_SSL_VERIFY_SERVER_CERT` non è supportata. Questa opzione è impostata su 0, il che significa che la connessione è crittografata, ma i certificati non sono verificati.

Note per l'utilizzo

La procedura `mysql.rds_set_external_master` deve essere eseguita dall'utente master. Questa procedura deve essere eseguita sull'istanza database MySQL da configurare come replica di lettura di un'istanza MySQL eseguita esternamente a Amazon RDS.

Prima di eseguire `mysql.rds_set_external_master`, devi configurare l'istanza di MySQL in esecuzione all'esterno di Amazon RDS come istanza database di origine. Per connetterti all'istanza MySQL in esecuzione all'esterno di Amazon RDS, devi specificare i valori di `replication_user_name` e `replication_user_password` che indicano un utente della replica dotato delle autorizzazioni `REPLICATION CLIENT` e `REPLICATION SLAVE` per l'istanza esterna di MySQL.

Per configurare un'istanza esterna di MySQL come istanza database di origine

1. Mediante il client MySQL scelto, eseguire la connessione all'istanza esterna di MySQL e creare un account utente da utilizzare per la replica. Di seguito è riportato un esempio.

   MySQL 5.7

   ```
   CREATE USER 'repl_user'@'mydomain.com' IDENTIFIED BY 'password';
   ```

   MySQL 8.0

   ```
   CREATE USER 'repl_user'@'mydomain.com' IDENTIFIED WITH mysql_native_password BY
    'password';
   ```

> **Note**
>
> Specifica una password diversa dal prompt mostrato qui come best practice per la sicurezza.

2. Nell'istanza esterna di MySQL, concedere i privilegi `REPLICATION CLIENT` e `REPLICATION SLAVE` all'utente della replica. L'esempio seguente concede i privilegi `REPLICATION CLIENT` e `REPLICATION SLAVE` su tutti i database per l'utente "repl_user" del dominio:

   MySQL 5.7

   ```
   GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user'@'mydomain.com'
    IDENTIFIED BY 'password';
   ```

   MySQL 8.0

   ```
   GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user'@'mydomain.com';
   ```

Per utilizzare la replica crittografata, configura l'istanza database di origine per utilizzare le connessioni SSL.

> **Note**
>
> Ti consigliamo di usare le repliche di lettura per gestire la replica tra due istanze database di Amazon RDS. In questo caso, si consiglia di usare solo questa e altre stored procedure correlate alla replica. Questo consente di usare topologie di replica più complesse tra le istanze database Amazon RDS. Queste stored procedure sono fornite principalmente per abilitare la replica con le istanze MySQL eseguite esternamente a Amazon RDS. Per ulteriori informazioni sulla gestione della replica tra istanze database Amazon RDS, consulta Uso delle repliche di lettura dell'istanza database.

Dopo aver chiamato `mysql.rds_set_external_master` per configurare un'istanza database di Amazon RDS come replica di lettura, puoi chiamare mysql.rds_start_replication nella replica di lettura per avviare il processo di replica. Puoi chiamare mysql.rds_reset_external_master per rimuovere la configurazione della replica di lettura.

Quando `mysql.rds_set_external_master` viene chiamato, Amazon RDS registra l'ora, l'utente e un'operazione di `set master` nelle tabelle `mysql.rds_history` e `mysql.rds_replication_status`.

Esempi

Nel caso di esecuzione su un'istanza database MySQL, l'esempio seguente configura l'istanza database come replica di lettura di un'istanza di MySQL eseguita esternamente a Amazon RDS.

```
call mysql.rds_set_external_master(
  'Externaldb.some.com',
  3306,
  'repl_user',
  'password',
  'mysql-bin-changelog.0777',
  120,
  0);
```

mysql.rds_set_external_master_with_auto_position

Configura un'istanza database RDS for MySQL come replica di lettura di un'istanza di MySQL eseguita esternamente a Amazon RDS. Questa procedura configura anche la replica ritardata e la replica basata sugli ID globali di transazione (GTID).

> **Important**
>
> Per eseguire questa procedura, è necessario abilitare `autocommit`. Per abilitarlo, impostare il parametro `autocommit` su 1. Per ulteriori informazioni sulla modifica dei parametri, consulta Modifica di parametri in un gruppo di parametri del database.

Sintassi

```
CALL mysql.rds_set_external_master_with_auto_position (
  host_name
  , host_port
  , replication_user_name
  , replication_user_password
  , ssl_encryption
  , delay
```

```
);
```

Parametri

host_name

Il nome host o l'indirizzo IP dell'istanza di MySQL eseguita esternamente a Amazon RDS per diventare l'istanza database di origine.

host_port

La porta utilizzata dall'istanza di MySQL eseguita esternamente a Amazon RDS e da configurare come istanza database di origine. Se la configurazione della rete include la replica della porta Secure Shell (SSH) che converte il numero di porta, specifica il numero di porta esposto da SSH.

replication_user_name

L'ID di un utente con autorizzazioni `REPLICATION CLIENT` e `REPLICATION SLAVE` nell'istanza di MySQL eseguita esternamente a Amazon RDS. Ti consigliamo di fornire un account utilizzato unicamente per la replica con l'istanza esterna.

replication_user_password

La password dell'ID utente specificata in `replication_user_name`.

ssl_encryption

Un valore che specifica se la crittografia Secure Socket Layer (SSL) è utilizzata sulla connessione di replica. 1 indica che la crittografia SSL deve essere utilizzata; 0 specifica che la crittografia non deve essere utilizzata. Il valore predefinito è 0.

> **Note**
>
> L'opzione `MASTER_SSL_VERIFY_SERVER_CERT` non è supportata. Questa opzione è impostata su 0, il che significa che la connessione è crittografata, ma i certificati non sono verificati.

delay

Numero minimo di secondi per ritardare la replica dall'istanza database di origine.

Il limite per questo parametro è un giorno (86400 secondi).

Note per l'utilizzo

La procedura `mysql.rds_set_external_master_with_auto_position` deve essere eseguita dall'utente master. Questa procedura deve essere eseguita sull'istanza database MySQL da configurare come replica di lettura di un'istanza MySQL eseguita esternamente a Amazon RDS.

Questa procedura è supportata per tutte le versioni di RDS per MySQL 5.7 e per RDS per MySQL 8.0.26 e versioni successive alla 8.0.

Prima di eseguire `mysql.rds_set_external_master_with_auto_position`, devi configurare l'istanza di MySQL in esecuzione all'esterno di Amazon RDS come istanza database di origine. Per connetterti all'istanza MySQL in esecuzione all'esterno di Amazon RDS, devi specificare i valori per `replication_user_name` e `replication_user_password`. Questi valori devono indicare un utente di replica che dispone delle autorizzazioni `REPLICATION CLIENT` e `REPLICATION SLAVE` sull'istanza esterna di MySQL.

Per configurare un'istanza esterna di MySQL come istanza database di origine

1. Mediante il client MySQL scelto, eseguire la connessione all'istanza esterna di MySQL e creare un account utente da utilizzare per la replica. Di seguito è riportato un esempio.

   ```
   CREATE USER 'repl_user'@'mydomain.com' IDENTIFIED BY 'SomePassW0rd'
   ```

2. Nell'istanza esterna di MySQL, concedere i privilegi `REPLICATION CLIENT` e `REPLICATION SLAVE` all'utente della replica. L'esempio seguente concede i privilegi `REPLICATION CLIENT` e `REPLICATION SLAVE` su tutti i database per l'utente `'repl_user'` per il dominio.

   ```
   GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user'@'mydomain.com'
   IDENTIFIED BY 'SomePassW0rd'
   ```

Per ulteriori informazioni, consulta Configurazione della replica della posizione del file di log binario con un'istanza di origine esterna..

> **Note**
>
> Ti consigliamo di usare le repliche di lettura per gestire la replica tra due istanze database di Amazon RDS. In questo caso, si consiglia di usare solo questa e altre stored procedure correlate alla replica. Questo consente di usare topologie di replica più complesse tra le istanze database Amazon RDS. Queste stored procedure sono fornite principalmente per

abilitare la replica con le istanze MySQL eseguite esternamente a Amazon RDS. Per ulteriori informazioni sulla gestione della replica tra istanze database Amazon RDS, consulta Uso delle repliche di lettura dell'istanza database.

Dopo aver chiamato `mysql.rds_set_external_master_with_auto_position` per configurare un'istanza database di Amazon RDS come replica di lettura, puoi chiamare mysql.rds_start_replication nella replica di lettura per avviare il processo di replica. Puoi chiamare mysql.rds_reset_external_master per rimuovere la configurazione della replica di lettura.

Quando viene chiamato `mysql.rds_set_external_master_with_auto_position`, Amazon RDS registra l'ora, l'utente e un'operazione `set master` nelle tabelle `mysql.rds_history` e `mysql.rds_replication_status`.

Per il disaster recovery puoi usare questa procedura con la stored procedure mysql.rds_start_replication_until o mysql.rds_start_replication_until_gtid. Per effettuare il roll forward delle modifiche a una replica di lettura ritardata al momento immediatamente precedente a un errore, puoi eseguire la procedura `mysql.rds_set_external_master_with_auto_position`. Dopo che la procedura `mysql.rds_start_replication_until_gtid` arresta la replica, puoi promuovere la replica di lettura come nuova istanza database primaria seguendo le istruzioni in Promozione di una replica di lettura a istanza database standalone.

Per usare la procedura `mysql.rds_rds_start_replication_until_gtid`, è necessario che sia abilitata la replica basata su GTID. Per passare a una specifica transazione basata su GTID che notoriamente causa un problema, puoi usare la stored procedure mysql.rds_skip_transaction_with_gtid. Per ulteriori informazioni sull'utilizzo della replica basata su GTID, consulta Utilizzo della replica basata su GTID.

Esempi

Nel caso di esecuzione su un'istanza database MySQL, l'esempio seguente configura l'istanza database come replica di lettura di un'istanza di MySQL eseguita esternamente a Amazon RDS. Imposta il ritardo di replica minimo su un'ora (3600 secondi) nell'istanza database MySQL. Una modifica dall'istanza database di origine MySQL in esecuzione all'esterno di Amazon RDS non viene applicata nella replica di lettura dell'istanza database MySQL per almeno un'ora.

```
call mysql.rds_set_external_master_with_auto_position(
  'Externaldb.some.com',
  3306,
  'repl_user',
```

```
  'SomePassW0rd',
  0,
  3600);
```

mysql.rds_set_external_master_with_delay

Configura un'istanza database RDS for MySQL come replica di lettura di un'istanza di MySQL in esecuzione all'esterno di Amazon RDS e configura la replica ritardata.

> **Important**
>
> Per eseguire questa procedura, è necessario abilitare `autocommit`. Per abilitarlo, impostare il parametro `autocommit` su 1. Per ulteriori informazioni sulla modifica dei parametri, consulta Modifica di parametri in un gruppo di parametri del database.

Sintassi

```
CALL mysql.rds_set_external_master_with_delay (
  host_name
  , host_port
  , replication_user_name
  , replication_user_password
  , mysql_binary_log_file_name
  , mysql_binary_log_file_location
  , ssl_encryption
  , delay
);
```

Parametri

host_name

Nome host o indirizzo IP dell'istanza MySQL in esecuzione all'esterno di Amazon RDS che diventerà l'istanza database di origine.

host_port

La porta utilizzata dall'istanza di MySQL eseguita esternamente a Amazon RDS e da configurare come istanza database di origine. Se la configurazione della rete include la replica della porta SSH che converte il numero di porta, specifica il numero di porta esposto da SSH.

replication_user_name

L'ID di un utente con autorizzazioni `REPLICATION CLIENT` e `REPLICATION SLAVE` nell'istanza di MySQL eseguita esternamente a Amazon RDS. Ti consigliamo di fornire un account utilizzato unicamente per la replica con l'istanza esterna.

replication_user_password

La password dell'ID utente specificata in `replication_user_name`.

mysql_binary_log_file_name

Il nome del log binario sull'istanza database di origine contiene le informazioni relative alla replica.

mysql_binary_log_file_location

Posizione nel log binario `mysql_binary_log_file_name` a partire dalla quale la replica inizierà a leggere le informazioni di replica.

È possibile determinare il nome e la posizione del file binlog in esecuzione `SHOW MASTER STATUS` sull'istanza del database di origine.

ssl_encryption

Un valore che specifica se la crittografia Secure Socket Layer (SSL) è utilizzata sulla connessione di replica. 1 indica che la crittografia SSL deve essere utilizzata; 0 specifica che la crittografia non deve essere utilizzata. Il valore predefinito è 0.

> **Note**
>
> L'opzione `MASTER_SSL_VERIFY_SERVER_CERT` non è supportata. Questa opzione è impostata su 0, il che significa che la connessione è crittografata, ma i certificati non sono verificati.

delay

Numero minimo di secondi per ritardare la replica dall'istanza database di origine.

Il limite per questo parametro è un giorno (86400 secondi).

Note per l'utilizzo

La procedura `mysql.rds_set_external_master_with_delay` deve essere eseguita dall'utente master. Questa procedura deve essere eseguita sull'istanza database MySQL da configurare come replica di lettura di un'istanza MySQL eseguita esternamente a Amazon RDS.

Prima di eseguire `mysql.rds_set_external_master_with_delay`, devi configurare l'istanza di MySQL in esecuzione all'esterno di Amazon RDS come istanza database di origine. Per connetterti all'istanza MySQL in esecuzione all'esterno di Amazon RDS, devi specificare i valori per `replication_user_name` e `replication_user_password`. Questi valori devono indicare un utente di replica che dispone delle autorizzazioni `REPLICATION CLIENT` e `REPLICATION SLAVE` sull'istanza esterna di MySQL.

Per configurare un'istanza esterna di MySQL come istanza database di origine

1. Mediante il client MySQL scelto, eseguire la connessione all'istanza esterna di MySQL e creare un account utente da utilizzare per la replica. Di seguito è riportato un esempio.

   ```
   CREATE USER 'repl_user'@'mydomain.com' IDENTIFIED BY 'SomePassW0rd'
   ```

2. Nell'istanza esterna di MySQL, concedere i privilegi `REPLICATION CLIENT` e `REPLICATION SLAVE` all'utente della replica. L'esempio seguente concede i privilegi `REPLICATION CLIENT` e `REPLICATION SLAVE` su tutti i database per l'utente `'repl_user'` per il dominio.

   ```
   GRANT REPLICATION CLIENT, REPLICATION SLAVE ON *.* TO 'repl_user'@'mydomain.com'
   IDENTIFIED BY 'SomePassW0rd'
   ```

Per ulteriori informazioni, consulta Configurazione della replica della posizione del file di log binario con un'istanza di origine esterna..

> **Note**
>
> Ti consigliamo di usare le repliche di lettura per gestire la replica tra due istanze database di Amazon RDS. In questo caso, si consiglia di usare solo questa e altre stored procedure correlate alla replica. Questo consente di usare topologie di replica più complesse tra le istanze database Amazon RDS. Queste stored procedure sono fornite principalmente per abilitare la replica con le istanze MySQL eseguite esternamente a Amazon RDS. Per ulteriori

informazioni sulla gestione della replica tra istanze database Amazon RDS, consulta Uso delle repliche di lettura dell'istanza database.

Dopo aver chiamato `mysql.rds_set_external_master_with_delay` per configurare un'istanza database di Amazon RDS come replica di lettura, puoi chiamare mysql.rds_start_replication nella replica di lettura per avviare il processo di replica. Puoi chiamare mysql.rds_reset_external_master per rimuovere la configurazione della replica di lettura.

Quando viene chiamato `mysql.rds_set_external_master_with_delay`, Amazon RDS registra l'ora, l'utente e un'operazione `set master` nelle tabelle `mysql.rds_history` e `mysql.rds_replication_status`.

Per il disaster recovery puoi usare questa procedura con la stored procedure mysql.rds_start_replication_until o mysql.rds_start_replication_until_gtid. Per effettuare il roll forward delle modifiche a una replica di lettura ritardata al momento immediatamente precedente a un errore, puoi eseguire la procedura `mysql.rds_set_external_master_with_delay`. Dopo che la procedura `mysql.rds_start_replication_until` arresta la replica, puoi promuovere la replica di lettura come nuova istanza database primaria seguendo le istruzioni in Promozione di una replica di lettura a istanza database standalone.

Per usare la procedura `mysql.rds_rds_start_replication_until_gtid`, è necessario che sia abilitata la replica basata su GTID. Per passare a una specifica transazione basata su GTID che notoriamente causa un problema, puoi usare la stored procedure mysql.rds_skip_transaction_with_gtid. Per ulteriori informazioni sull'utilizzo della replica basata su GTID, consulta Utilizzo della replica basata su GTID.

La procedura `mysql.rds_set_external_master_with_delay` è disponibile nelle seguenti versioni di RDS for MySQL:

- MySQL 8.0.26 e versioni successive alla 8.0
- Tutte le versioni 5.7

Esempi

Nel caso di esecuzione su un'istanza database MySQL, l'esempio seguente configura l'istanza database come replica di lettura di un'istanza di MySQL eseguita esternamente a Amazon RDS. Imposta il ritardo di replica minimo su un'ora (3600 secondi) nell'istanza database MySQL. Una

modifica dall'istanza database di origine MySQL in esecuzione all'esterno di Amazon RDS non viene applicata nella replica di lettura dell'istanza database MySQL per almeno un'ora.

```
call mysql.rds_set_external_master_with_delay(
  'Externaldb.some.com',
  3306,
  'repl_user',
  'SomePassW0rd',
  'mysql-bin-changelog.000777',
  120,
  0,
  3600);
```

mysql.rds_set_master_auto_position

Imposta la modalità di replica in modo che sia basata sulle posizioni dei file di log binario o sugli ID globali di transazione (GTID).

Sintassi

```
CALL mysql.rds_set_master_auto_position (
auto_position_mode
);
```

Parametri

auto_position_mode

Valore che indica se usare la replica basata sulla posizione del file di log o la replica basata su GTID:

- `0` – Usa il metodo di replica basato sulla posizione del file di log binario. Il valore di default è `0`.
- 1 – Usa il metodo di replica basato su GTID.

Note per l'utilizzo

La procedura `mysql.rds_set_master_auto_position` deve essere eseguita dall'utente master.

Questa procedura è supportata per tutte le versioni di RDS per MySQL 5.7 e per RDS per MySQL 8.0.26 e versioni successive alla 8.0.

mysql.rds_set_source_delay

Imposta il numero minimo di secondi per ritardare la replica dall'istanza database di origine alla replica di lettura corrente. Usa questa procedura in presenza di una connessione a una replica di lettura per ritardare la replica rispetto all'istanza database di origine.

Sintassi

```
CALL mysql.rds_set_source_delay(
delay
);
```

Parametri

delay

Numero minimo di secondi per ritardare la replica dall'istanza database di origine.

Il limite per questo parametro è un giorno (86400 secondi).

Note per l'utilizzo

La procedura `mysql.rds_set_source_delay` deve essere eseguita dall'utente master.

Per il disaster recovery puoi usare questa procedura con la stored procedure mysql.rds_start_replication_until o mysql.rds_start_replication_until_gtid. Per effettuare il roll forward delle modifiche a una replica di lettura ritardata al momento immediatamente precedente a un errore, puoi eseguire la procedura `mysql.rds_set_source_delay`. Dopo che la procedura `mysql.rds_start_replication_until` o `mysql.rds_start_replication_until_gtid` arresta la replica, puoi promuovere la replica di lettura come nuova istanza database master seguendo le istruzioni in Promozione di una replica di lettura a istanza database standalone.

Per usare la procedura `mysql.rds_rds_start_replication_until_gtid`, è necessario che sia abilitata la replica basata su GTID. Per passare a una specifica transazione basata su GTID che notoriamente causa un problema, puoi usare la stored procedure mysql.rds_skip_transaction_with_gtid. Per ulteriori informazioni sulla replica basata su GTID, consulta Utilizzo della replica basata su GTID.

La procedura `mysql.rds_set_source_delay` è disponibile nelle seguenti versioni di RDS for MySQL:

- MySQL 8.0.26 e versioni successive alla 8.0
- Tutte le versioni 5.7

Esempi

Per ritardare la replica rispetto all'istanza database di origine nella replica di lettura corrente per almeno un'ora (3600 secondi), puoi chiamare `mysql.rds_set_source_delay` con il parametro seguente:

```
CALL mysql.rds_set_source_delay(3600);
```

mysql.rds_skip_transaction_with_gtid

Ignora la replica di una transazione con l'ID globale di transazione (GTID) specificato in un'istanza database MySQL.

Puoi usare questa procedura per il ripristino di emergenza quando è noto che una specifica transazione GTID causa un problema. Usa questa stored procedure per saltare la transazione problematica. Esempi di transazioni problematiche includono le transazioni che disabilitano la replica, eliminano dati importanti o con le quali l'istanza database diventa non disponibile.

Sintassi

```
CALL mysql.rds_skip_transaction_with_gtid (
gtid_to_skip
);
```

Parametri

gtid_to_skip

GTID della transazione di replica da ignorare.

Note per l'utilizzo

La procedura `mysql.rds_skip_transaction_with_gtid` deve essere eseguita dall'utente master.

Questa procedura è supportata per tutte le versioni di RDS per MySQL 5.7 e per RDS per MySQL 8.0.26 e versioni successive alla 8.0.

Esempi

Nell'esempio seguente viene ignorata la replica della transazione con il GTID `3E11FA47-71CA-11E1-9E33-C80AA9429562:23`.

```
CALL mysql.rds_skip_transaction_with_gtid('3E11FA47-71CA-11E1-9E33-C80AA9429562:23');
```

mysql.rds_skip_repl_error

Ignora ed elimina un errore di replica su una replica di lettura database MySQL.

Sintassi

```
CALL mysql.rds_skip_repl_error;
```

Note per l'utilizzo

La procedura `mysql.rds_skip_repl_error` deve essere eseguita dall'utente master su una replica di lettura. Per ulteriori informazioni su questa procedura, consulta Chiamata della procedura mysql.rds_skip_repl_error.

Per determinare se ci sono errori, esegui il comando MySQL `SHOW REPLICA STATUS\G`. Se un errore di replica non è critico, puoi eseguire `mysql.rds_skip_repl_error` per ignorare l'errore. Se vi sono più errori, `mysql.rds_skip_repl_error` elimina il primo, quindi informa della presenza di altri errori. Puoi quindi utilizzare `SHOW REPLICA STATUS\G` per determinare l'operazione corretta per l'errore successivo. Per informazioni sui valori restituiti, consulta Istruzione SHOW REPLICA STATUS nella documentazione di MySQL.

> **Note**
>
> Versioni precedenti di MySQL utilizzate `SHOW SLAVE STATUS` al posto di `SHOW REPLICA STATUS`. Se si utilizza una versione MySQL prima della 8.0.23, utilizzare `SHOW SLAVE STATUS`.

Per ulteriori informazioni sulla risoluzione degli errori di replica con Amazon RDS, consulta Risoluzione dei problemi relativi a una replica di lettura MySQL.

Errore di replica interrotta

Quando si chiama la procedura `mysql.rds_skip_repl_error`, è possibile che venga visualizzato un messaggio di errore che indica che la replica è inattiva o disattivata.

Questo messaggio di errore viene visualizzato se si esegue la procedura sull'istanza primaria anziché sulla replica di lettura. È necessario eseguire questa procedura sulla replica di lettura affinché funzioni.

Questo messaggio di errore può essere visualizzato anche quando si esegue la procedura sulla replica di lettura, ma la replica non viene riavviata correttamente.

Se devi ignorare un numero elevato di errori, il ritardo della replica potrebbe superare il periodo di retention predefinito per i file di log binari (binlog). In questo caso può verificarsi un errore irreversibile causato dall'eliminazione dei file binlog prima della loro riproduzione nella replica di lettura. Questa eliminazione causa l'arresto della replica e non è più possibile chiamare il comando `mysql.rds_skip_repl_error` per ignorare errori di replica.

Puoi limitare questo problema aumentando il numero di ore di retention dei file binlog nell'istanza database di origine. Una volta aumentato il tempo di retention dei file binlog, puoi riavviare la replica e chiamare il comando `mysql.rds_skip_repl_error` secondo necessità.

Per impostare il periodo di retention dei file binlog, usa la procedura mysql.rds_set_configuration e specifica un parametro di configurazione di `'binlog retention hours'` insieme al numero di ore di retention dei file binlog nel cluster di database. Nell'esempio seguente il periodo di retention dei file binlog è impostato su 48 ore.

```
CALL mysql.rds_set_configuration('binlog retention hours', 48);
```

mysql.rds_start_replication

Avvia la replica da un'istanza database RDS per MySQL.

> **Note**
>
> Puoi usare la stored procedure mysql.rds_start_replication_until o mysql.rds_start_replication_until_gtid per avviare la replica da un'istanza database RDS per MySQL e arrestare la replica in corrispondenza della posizione del file di log binario specificato.

Sintassi

```
CALL mysql.rds_start_replication;
```

Note per l'utilizzo

La procedura `mysql.rds_start_replication` deve essere eseguita dall'utente master.

Per importare dati da un'istanza di MySQL in esecuzione all'esterno di Amazon RDS, devi chiamare `mysql.rds_start_replication` nella replica di lettura per avviare il processo di replica dopo aver chiamato `mysql.rds_set_external_master` per creare la configurazione della replica. Per ulteriori informazioni, consulta Ripristino di un backup in un'istanza database MySQL.

Per esportare dati in un'istanza di MySQL in esecuzione all'esterno di Amazon RDS, devi chiamare `mysql.rds_start_replication` e `mysql.rds_stop_replication` nella replica di lettura per controllare alcune operazioni di replica, come l'eliminazione di log binari. Per ulteriori informazioni, consulta Esportazione di dati da un'istanza database MySQL tramite la replica.

Puoi anche chiamare `mysql.rds_start_replication` nella replica di lettura per riavviare un processo di replica arrestato in precedenza chiamando `mysql.rds_stop_replication`. Per ulteriori informazioni, consulta Uso delle repliche di lettura dell'istanza database.

mysql.rds_start_replication_until

Avvia la replica da un'istanza database RDS per MySQL e la arresta in corrispondenza della posizione del file di log binario specificato.

Sintassi

```
CALL mysql.rds_start_replication_until (
replication_log_file
  , replication_stop_point
);
```

Parametri

`replication_log_file`

Il nome del log binario sull'istanza database di origine che contiene le informazioni relative alla replica.

replication_stop_point

Posizione nel log binario `replication_log_file` in corrispondenza di cui la replica verrà arrestata.

Note per l'utilizzo

La procedura `mysql.rds_start_replication_until` deve essere eseguita dall'utente master.

La procedura `mysql.rds_start_replication_until` è disponibile nelle seguenti versioni di RDS for MySQL:

- MySQL 8.0.26 e versioni successive alla 8.0
- Tutte le versioni 5.7

Puoi usare questa procedura archiviata con la replica ritardata per il disaster recovery. Se hai configurato la replica ritardata, puoi usare questa procedura archiviata per effettuare il roll forward delle modifiche a una replica di lettura ritardata al momento immediatamente precedente a un errore. Dopo che questa procedura archiviata arresta la replica, puoi promuovere la replica di lettura come nuova istanza database primaria seguendo le istruzioni in Promozione di una replica di lettura a istanza database standalone.

Puoi configurare la replica ritardata usando le procedure archiviate seguenti:

- mysql.rds_set_configuration
- mysql.rds_set_external_master_with_delay
- mysql.rds_set_source_delay

Il nome file specificato per il parametro `replication_log_file` deve corrispondere al nome file binlog dell'istanza database di origine.

Quando il parametro `replication_stop_point` specifica una posizione di arresto nel passato, la replica viene arrestata immediatamente.

Esempi

L'esempio seguente avvia la replica e replica le modifiche fino a raggiungere la posizione `120` nel file di log binario `mysql-bin-changelog.000777`.

```
call mysql.rds_start_replication_until(
  'mysql-bin-changelog.000777',
  120);
```

mysql.rds_start_replication_until_gtid

Avvia la replica da un'istanza database RDS per MySQL e la arresta immediatamente dopo l'ID globale di transazione (GTID) specificato.

Sintassi

```
CALL mysql.rds_start_replication_until_gtid(gtid);
```

Parametri

gtid

Il GTID dopo il quale deve essere arrestata la replica.

Note per l'utilizzo

La procedura `mysql.rds_start_replication_until_gtid` deve essere eseguita dall'utente master.

Questa procedura è supportata per tutte le versioni di RDS per MySQL 5.7 e per RDS per MySQL 8.0.26 e versioni successive alla 8.0.

Puoi usare questa procedura archiviata con la replica ritardata per il disaster recovery. Se hai configurato la replica ritardata, puoi usare questa procedura archiviata per effettuare il roll forward delle modifiche a una replica di lettura ritardata al momento immediatamente precedente a un errore. Dopo che questa procedura archiviata arresta la replica, puoi promuovere la replica di lettura come nuova istanza database primaria seguendo le istruzioni in Promozione di una replica di lettura a istanza database standalone.

Puoi configurare la replica ritardata usando le procedure archiviate seguenti:

- mysql.rds_set_configuration
- mysql.rds_set_external_master_with_delay

- mysql.rds_set_source_delay

Quando il parametro `gtid` specifica una transazione che è già stata eseguita dalla replica, la procedura viene arrestata immediatamente.

Esempi

L'esempio seguente avvia la replica e replica le modifiche finché non raggiunge il GTID `3E11FA47-71CA-11E1-9E33-C80AA9429562:23`.

```
call mysql.rds_start_replication_until_gtid('3E11FA47-71CA-11E1-9E33-C80AA9429562:23');
```

mysql.rds_stop_replication

Arresta la replica da un'istanza database MySQL.

Sintassi

```
CALL mysql.rds_stop_replication;
```

Note per l'utilizzo

La procedura `mysql.rds_stop_replication` deve essere eseguita dall'utente master.

Se configuri la replica per importare dati da un'istanza di MySQL in esecuzione all'esterno di Amazon RDS, puoi chiamare `mysql.rds_stop_replication` nella replica di lettura per arrestare il processo di replica al termine dell'importazione. Per ulteriori informazioni, consulta Ripristino di un backup in un'istanza database MySQL.

Se configuri la replica per esportare dati in un'istanza di MySQL esterna ad Amazon RDS, devi chiamare `mysql.rds_start_replication` e `mysql.rds_stop_replication` nella replica di lettura per controllare alcune operazioni di replica, come l'eliminazione di log binari. Per ulteriori informazioni, consulta Esportazione di dati da un'istanza database MySQL tramite la replica.

Puoi usare `mysql.rds_stop_replication` anche per arrestare la replica tra due istanze database Amazon RDS. In genere si arresta una replica per eseguire un'operazione di lunga durata nella replica di lettura, come la creazione di un indice di grandi dimensioni nella replica di lettura. Puoi riavviare qualsiasi processo di replica arrestato chiamando mysql.rds_start_replication nella replica di lettura. Per ulteriori informazioni, consulta Uso delle repliche di lettura dell'istanza database.

Precaricamento della cache di InnoDB

Le seguenti stored procedure salvano, caricano o annullano il caricamento del pool di buffer di InnoDB nelle istanze database RDS per MySQL. Per ulteriori informazioni, consulta Precaricamento della cache InnoDB per MySQL su Amazon RDS.

Argomenti

mysql.rds_innodb_buffer_pool_dump_now

Esegue il dump dello stato corrente del pool di buffer sul disco.

Sintassi

```
CALL mysql.rds_innodb_buffer_pool_dump_now();
```

Note per l'utilizzo

La procedura `mysql.rds_innodb_buffer_pool_dump_now` deve essere eseguita dall'utente master.

mysql.rds_innodb_buffer_pool_load_abort

Annulla un caricamento in corso dello stato del pool di buffer salvato.

Sintassi

```
CALL mysql.rds_innodb_buffer_pool_load_abort();
```

Note per l'utilizzo

La procedura `mysql.rds_innodb_buffer_pool_load_abort` deve essere eseguita dall'utente master.

mysql.rds_innodb_buffer_pool_load_now

Carica lo stato salvato del pool di buffer dal disco.

Sintassi

```
CALL mysql.rds_innodb_buffer_pool_load_now();
```

Note per l'utilizzo

La procedura `mysql.rds_innodb_buffer_pool_load_now` deve essere eseguita dall'utente master.

Amazon RDS for Oracle

Amazon RDS supporta le istanze database che eseguono le seguenti versioni ed edizioni di Oracle Database:

- Oracle Database 21c (21.0.0.0)
- Oracle Database 19c (19.0.0.0)

> **Note**
>
> Oracle Database 11g, Oracle Database 12c e Oracle Database 18c sono versioni legacy non più supportate in Amazon RDS.

Prima di creare un'istanza database, completa i passi indicati nella sezione Configurazione di Amazon RDS di questa guida. Quando si crea un'istanza database utilizzando l'account master, l'account ottiene privilegi DBA, con alcune limitazioni. Utilizzare questo account per attività amministrative, ad esempio la creazione di account di database aggiuntivi. Non è possibile utilizzare SYS, SYSTEM o altri account amministrativi forniti da Oracle.

Puoi creare:

- Istanze DB
- Snapshot DB
- Ripristini point-in-time
- Backup automatizzati
- Backup manuali

È possibile utilizzare istanze DB che eseguono Oracle all'interno di un VPC. Inoltre, è possibile abilitare varie opzioni per aggiungere altre funzionalità all'istanza database Oracle. Amazon RDS supporta le implementazioni Multi-AZ per Oracle come soluzione failover a elevata disponibilità.

> **Important**
>
> Per offrire un'esperienza di servizio gestito, Amazon RDS non fornisce accesso shell alle istanze database. Limita anche l'accesso ad alcune procedure di sistema e tabelle che

richiedono privilegi avanzati. Puoi accedere al database utilizzando i client SQL standard come Oracle SQL*Plus. Tuttavia, non è possibile accedere direttamente all'host utilizzando Telnet o Secure Shell (SSH).

Argomenti

Panoramica di Oracle su Amazon RDS

Le sezioni seguenti includono una panoramica di RDS per Oracle.

Argomenti

- Limitazioni di RDS for Oracle

Funzionalità di RDS for Oracle

Amazon RDS for Oracle supporta la maggior parte delle caratteristiche e funzionalità di Oracle Database. Alcune funzionalità potrebbero avere un supporto o privilegi limitati. Alcune funzionalità sono disponibili solo nella versione Enterprise Edition e alcune richiedono licenze aggiuntive. Per ulteriori informazioni sulle funzionalità di Oracle per versioni specifiche del database, consultare il Manuale dell'utente delle informazioni sulla licenza del database Oracle per la versione che si sta utilizzando.

Please change to "Puoi filtrare le nuove funzionalità Amazon RDS alla pagina Quali sono le novità del database?. Per Prodotti, scegli Amazon RDS. Quindi esegui la ricerca utilizzando parole chiave come **`Oracle 2022`**.

> **Note**
>
> I seguenti elenchi non sono esaustivi.

Argomenti

- Nuove funzionalità di RDS per Oracle
- Funzioni supportate per RDS per Oracle
- Funzioni non supportate per RDS per Oracle

Nuove funzionalità di RDS per Oracle

Per visualizzare le nuove funzionalità di RDS for Oracle, utilizza le seguenti tecniche:

- Ricerca in Cronologia dei documenti della parola chiave **`Oracle`**.
- Filtra le nuove funzionalità di Amazon RDS nella sezione What's New with Database? pagina. Per Prodotti, scegli Amazon RDS. Quindi cerca **`Oracle`** ***`YYYY`***, dove: *YYYY* è un anno come **`2024`**.

Funzioni supportate per RDS per Oracle

Amazon RDS per Oracle supporta le seguenti funzionalità di Oracle Database:

- Advanced Compression
- Application Express (APEX)

  Per ulteriori informazioni, consulta Oracle Application Express (APEX).
- Gestione automatica della memoria
- Gestione automatica di annulla operazione
- Automatic Workload Repository (AWR)

  Per ulteriori informazioni, consulta Generazione di report sulle prestazioni con AWR (Automatic Workload Repository).
- Active Data Guard con prestazioni massime nella stessa AWS regione o in più AWS regioni

  Per ulteriori informazioni, consulta Utilizzo di repliche di lettura per Amazon RDS per Oracle.
- Tabelle blockchain (Oracle Database 21c e versioni successive)

  Per ulteriori informazioni, consultare Gestione di tabelle Blockchain nella documentazione relativa a Oracle Database.
- Continuous Query Notification (versione 12.1.0.2.v7 e successive)

  Per ulteriori informazioni, consulta Using Continuous Query Notification (CQN) nella documentazione Oracle.
- Data Redaction
- Notifica della modifica del database

  Per ulteriori informazioni, consulta Database Change Notification nella documentazione Oracle.

> **Note**
>
> Questa funzionalità diventa Continuous Query Notification in Oracle Database 12c Release 1 (12.1) e versioni successive.

- Database in memoria (Oracle Database 12c e versioni successive)
- Query e transazioni distribuite
- Edition-Based Redefinition

  Per ulteriori informazioni, consulta Impostazione dell'edizione predefinita per un'istanza database.

- EM Express (12c e versioni successive)

  Per ulteriori informazioni, consulta Oracle Enterprise Manager.
- Auditing granulare
- Flashback Table, Flashback Query, Flashback Transaction Query
- Rollover graduale della password per applicazioni (Oracle Database 21c e versioni successive)

  Per ulteriori informazioni, consultare Gestione graduale del rollover delle password del database per le applicazioni nella documentazione relativa a Oracle Database.
- HugePages

  Per ulteriori informazioni, consulta Attivazione di HugePages per un'istanza RDS per Oracle.
- Import/export (legacy e Data Pump) e SQL*Loader

  Per ulteriori informazioni, consulta Importazione di dati in Oracle in Amazon RDS.
- Java Virtual Machine (JVM)

  Per ulteriori informazioni, consulta Oracle Java Virtual Machine.
- JavaScript (Oracle Database 21c e versioni successive)

  Per ulteriori informazioni, consultare DBMS_MLE nella documentazione di Oracle Database.
- Label Security (Oracle Database 12c e versioni successive)

  Per ulteriori informazioni, consulta Oracle Label Security.
- Locator

  Per ulteriori informazioni, consulta Oracle Locator.
- Viste materializzate
- Multimedia

  Per ulteriori informazioni, consulta Oracle Multimedia.
- Multitenant

  L'architettura multitenant Oracle è supportata in tutte le versioni di Oracle Database 19c e successive. Per ulteriori informazioni, consulta Utilizzo di database CDB per RDS per Oracle.
- Crittografia di rete

Per ulteriori informazioni, consulta Oracle native network encryption e Oracle Secure Sockets Layer.

- Partizionamento
- Test reale dell'applicazione

  Per utilizzare tutte le funzionalità di acquisizione e riproduzione, è necessario utilizzare Amazon Elastic File System (Amazon EFS) per accedere ai file generati da Oracle Real Application Testing. Per ulteriori informazioni, consulta Integrazione Amazon EFS il post sul blog Use Oracle Real Application Testing features with Amazon RDS for Oracle.
- Sharding a livello di applicazione (ma non la funzionalità Oracle Sharding)
- Modulo Spatial and Graph

  Per ulteriori informazioni, consulta Oracle Spatial.
- Ottimizzazione delle star query
- Stream e gestione avanzata delle code
- Summary Management – Materialized View Query Rewrite
- Text (i tipi di datastore file e URL non sono supportati)
- Total Recall
- Transparent Data Encryption (TDE)

  Per ulteriori informazioni, consulta Oracle Transparent Data Encryption.
- Unified Auditing, modalità mista

  Per ulteriori informazioni, consultare Mixed Mode Auditing nella documentazione Oracle.
- XML DB (senza XML DB Protocol Server)

  Per ulteriori informazioni, consulta Oracle XML DB.
- Database virtuale privato

Funzioni non supportate per RDS per Oracle

Amazon RDS per Oracle non supporta le seguenti funzionalità di Oracle Database:

- Automatic Storage Management (ASM)

- Vault del database
- Flashback Database

> **Note**
>
> Per soluzioni alternative, consulta la voce del blog AWS Database Alternative alla funzionalità del database Oracle flashback in Amazon RDS for Oracle.

- FTP e SFTP
- Tabelle partizionate ibride
- Gateway di messaggistica
- Oracle Enterprise Manager Cloud Control Management Repository
- Real Application Clusters (Oracle RAC)
- Real Application Security (RAS)
- Unified Auditing, Pure Mode
- Schema di Workspace Manager (WMSYS)

> **Note**
>
> L'elenco precedente non è esauriente.

> **Warning**
>
> In generale, Amazon RDS non impedisce la creazione di schemi per feature non supportate. Tuttavia, se si creano schemi per funzionalità e componenti Oracle che richiedono privilegi SYS, è possibile danneggiare il dizionario dati e pregiudicare la disponibilità dell'istanza. Utilizzare solo le funzioni e gli schemi supportati disponibili in Aggiunta di opzioni alle istanze database Oracle.

Release di RDS per Oracle

Amazon RDS per Oracle supporta versioni multiple di Oracle Database.

> **Note**
>
> Per informazioni sull'aggiornamento delle versioni, consulta Aggiornamento del motore di database RDS per Oracle.

Argomenti

- Oracle Database 21c con Amazon RDS
- Oracle Database 19c con Amazon RDS
- Oracle Database 12c con Amazon RDS

Oracle Database 21c con Amazon RDS

Amazon RDS supporta Oracle Database 21c, che include Oracle Enterprise Edition e Oracle Standard Edition 2. Oracle Database 21c (21.0.0.0) include molti aggiornamenti e nuove funzionalità rispetto alla versione precedente. Una modifica fondamentale è che Oracle Database 21c supporta solo l'architettura multitenant: non è più possibile creare un database come un tradizionale non CDB. Per ulteriori informazioni sulle differenze tra CDB e non CDB, consulta Limitazioni per i CDB RDS per Oracle.

In questa sezione troverai le funzionalità e le modifiche importanti per usare Oracle Database 21c (21.0.0.0) su Amazon RDS. Per un elenco completo delle modifiche, consultare la documentazione sul database Oracle 21c. Per un elenco completo delle funzionalità di ciascuna edizione di Oracle Database 21c, consulta Pacchetti di caratteristiche, opzioni e gestione consentiti dall'offerta di database Oracle nella documentazione di Oracle.

Modifiche ai parametri di Amazon RDS per Oracle Database 21c (21.0.0.0)

Oracle Database 21c (21.0.0.0) comprende diversi nuovi parametri, oltre a molti parametri con nuovi intervalli e valori predefiniti.

Argomenti

- Nuovi parametri
- Modifiche per il parametro compatibile
- Parametri rimossi

Nuovi parametri

La tabella seguente mostra i nuovi parametri di Amazon RDS per Oracle Database 21c (21.0.0.0).

Nome	Intervallo di valori	Valore predefinito	Modificabili	Descrizione
blockchain_table_max_no_drop	NONE | 0	NONE	Y	Consente di controllare la quantità massima di tempo di inattività che è possibile specificare durante la creazione di una tabella blockchain.
dbnest_enable	NONE | CDB_RESOURCE_PDB_ALL	NONE	N	Consente di abilitare o disabilitare DBNest. DbNest fornisce l'isolamento e la gestione delle risorse del sistema operativo, l'isolamento del file system e l'elaborazione sicura per i PDB.
dbnest_pdb_fs_conf	NONE | *pathname*	NONE	N	Specifica il file di configurazione del file system dbNest per un PDB.
diagnostics_control	ERROR | WARNING | IGNORE	IGNORE	Y	Consente di controllare e monitorare gli utenti che eseguono operazioni di diagnostica del database potenzialmente pericolose.
drcp_dedicated_opt	YES | NO	YES	Y	Abilita o disabilita l'utilizzo dell'ottimizzazione dedicata con Database Resident Connection Pooling (DRCP).
enable_per_pdb_drcp	true | false	true	N	Controlla se il Database Resident Connection

Nome	Intervallo di valori	Valore predefinito	Modificabili	Descrizione
				Pooling (DRCP) configura un connection pool per l'intero CDB o un connection pool isolato per ciascun PDB.
inmemory_deep_vectorization	`true | false`	`true`	Y	Abilita o disabilita il framework di vettorizzazione approfondita.
mandatory_user_profile	*`profile_name`*	N/D	N	Specifica il profilo utente obbligatorio per un CDB o PDB.
optimizer_capture_sql_quarantine	`true | false`	`false`	Y	Abilita o disabilita il framework di vettorizzazione approfondita.
optimizer_use_sql_quarantine	`true | false`	`false`	Y	Attiva o disabilita la creazione automatica di configurazioni di SQL Quarantine.
result_cache_execution_threshold	`0` Da a `68719476736`	2	Y	Specifica il numero massimo di volte in cui è possibile eseguire una funzione PL/SQL prima che il risultato venga memorizzato nella cache dei risultati.
result_cache_max_temp_result	`0` Da a `100`	5	Y	Specifica la percentuale di `RESULT_CACHE_MAX_TEMP_SIZE` che qualsiasi risultato di query memorizzato nella cache può consumare.

Nome	Intervallo di valori	Valore predefinito	Modificabili	Descrizione
result_cache_max_temp_size	0 Da a 2199023255552	RESULT_CACHE_SIZE * 10	Y	Specifica la quantità massima di tablespace temporaneo (in byte) che può essere utilizzata dalla cache dei risultati.
sga_min_size	Da 0 a 2199023255552 (il valore massimo è del 50% di sga_target)	0	Y	Indica un possibile valore minimo per l'utilizzo SGA di un database pluggable (PDB).
tablespace_encryption_default_algorithm	GOST256 | SEED128 | ARIA256 | ARIA192 | ARIA128 | 3DES168 | AES256 | AES192 | AES128	AES128	Y	Specifica l'algoritmo predefinito utilizzato dal database per la crittografia di un tablespace.

Modifiche per il parametro compatibile

Il parametro `compatible` su Amazon RDS ha un nuovo valore massimo per Oracle Database 21c (21.0.0.0). La tabella seguente indica il nuovo valore predefinito.

Nome del parametro	Valore massimo di Oracle Database 21c (21.0.0.0)
compatible	21.0.0

Parametri rimossi

In Oracle Database 21c (21.0.0.0) sono stati rimossi i seguenti parametri:

- `remote_os_authent`
- `sec_case_sensitive_logon`
- `unified_audit_sga_queue_size`

Oracle Database 19c con Amazon RDS

Amazon RDS supporta Oracle Database 19c che comprende Oracle Enterprise Edition e Oracle Standard Edition Two.

Oracle Database 19c (19.0.0.0) include molti aggiornamenti e nuove funzionalità rispetto alla versione precedente. In questa sezione troverai le funzionalità e le modifiche importanti per usare Oracle Database 19c (19.0.0.0) su Amazon RDS. Per un elenco completo delle modifiche, consultare la documentazione sul Database Oracle 19c. Per un elenco completo delle caratteristiche supportate da ciascuna edizione di Oracle Database 19c, consulta Permitted Features, Options, and Management Packs by Oracle Database Offering nella documentazione di Oracle.

Modifiche al parametro Amazon RDS for Oracle Database 19c (19.0.0.0)

Oracle Database 19c (19.0.0.0) comprende diversi nuovi parametri, oltre a molti parametri con nuovi intervalli e valori predefiniti.

Argomenti

Nuovi parametri

La tabella seguente mostra i nuovi parametri Amazon RDS for Oracle Database 19c (19.0.0.0).

Nome	Valori	Modifi ili	Descrizione
lob_signature_enable	TRUE, FALSE (predefinito)	Y	Attiva o disattiva la funzione di firma LOB locator.
max_datapump_parallel_per_job	Da 1 a 1024, o AUTO	Y	Specifica il numero massimo di processi paralleli consentiti per ciascuna attività Oracle Data Pump.

Modifiche al parametro compatibile

Il parametro `compatible` ha un nuovo valore massimo per Oracle Database 19c (19.0.0.0) su Amazon RDS. La tabella seguente indica il nuovo valore predefinito.

Nome del parametro	Valore massimo di Oracle Database 19c (19.0.0.0)
compatible	19.0.0

Parametri rimossi

In Oracle Database 19c (19.0.0.0) sono stati rimossi i parametri seguenti:

- `exafusion_enabled`
- `max_connections`
- `o7_dictionary_access`

Oracle Database 12c con Amazon RDS

Amazon RDS ha terminato il supporto per Oracle Database 12c sia per Oracle Enterprise Edition che per Oracle Standard Edition 2.

Argomenti

- Oracle Database 12c Release 2 (12.2.0.1) con Amazon RDS
- Oracle Database 12c Release 1 (12.1.0.2) con Amazon RDS

Oracle Database 12c Release 2 (12.2.0.1) con Amazon RDS

A partire dal 31 marzo 2022, Oracle Corporation ha terminato il supporto per Oracle Database 12c Release 2 (12.2.0.1) per BYOL e LI. In questa data, la release passa da Oracle Extended Support a Oracle Sustaining Support, indicando la fine del supporto per questa versione. Per ulteriori informazioni, consulta la pianificazione della fine del supporto in AWS re:Post.

Data	Azione
1 aprile 2022	Amazon RDS ha iniziato gli aggiornamenti automatici delle istanze Oracle Database 12c Release 2 (12.2.0.1) a Oracle Database 19c.
1 aprile 2022	Amazon RDS ha iniziato gli aggiornamenti automatici a Oracle Database 19c per tutte le istanze database Oracle Database 12c Release 2 (12.2.0.1) ripristinate da snapshot. L'aggiornamento automatico si verifica durante le finestre di manutenzione. Tuttavia, se non sono disponibili finestre di manutenzione quando è necessario eseguire l'aggiornamento, Amazon RDS per Oracle aggiorna immediatamente il motore.

Oracle Database 12c Release 1 (12.1.0.2) con Amazon RDS

A partire dal 31 luglio 2022 Amazon RDS ha terminato il supporto per Oracle Database 12c Release 1 (12.1.0.2) per BYOL e LI. La release passa da Oracle Extended Support a Oracle Sustaining Support, indicando che il Supporto Oracle non rilascerà più aggiornamenti critici per questa versione. Per ulteriori informazioni, consulta la pianificazione della fine del supporto in AWS re:Post.

Data	Azione
1 agosto 2022	Amazon RDS ha iniziato a rilasciare gli aggiornamenti automatici delle istanze di Oracle Database 12c Release 1 (12.1.0.2) alla release di aggiornamento (RU) più recente per Oracle Database 19c. L'aggiornamento automatico si verifica durante le finestre di manutenzione. Tuttavia, se non sono disponibili finestre di manutenzione quando è necessario eseguire l'aggiornamento, Amazon RDS per Oracle aggiorna immediatamente il motore.

Data	Azione
1 agosto 2022	Amazon RDS ha iniziato a rilasciare aggiornamenti automatici per Oracle Database 19c per tutte le istanze database Oracle Database 12c Release 1 (12.1.0.2) ripristinate da snapshot.

Opzioni di licenza per RDS per Oracle

Sono disponibili due opzioni di licenza per Amazon RDS for Oracle: licenza inclusa e Bring-Your-Own-License (BYOL, uso di licenze proprie). Dopo aver creato un'istanza database di Oracle su Amazon RDS, puoi modificare il modello di licenza modificando l'istanza database. Per ulteriori informazioni, consulta Modifica di un'istanza database Amazon RDS.

> **Important**
>
> Assicurati di disporre della licenza Oracle Database appropriata, con Software Update License and Support, per la tua classe di istanza DB e l'edizione di Oracle Database. Assicurati inoltre di disporre delle licenze per tutte le funzionalità di Oracle Database concesse in licenza separatamente.

Argomenti

- Modello con licenza inclusa per SE2
- Bring Your Own License (BYOL) per EE e SE2
- Opzioni di licenza delle implementazioni Multi-AZ per Oracle

Modello con licenza inclusa per SE2

Nel modello Licenza inclusa, non è necessario acquistare separatamente le licenze di Oracle Database. AWS detiene la licenza per il software di database Oracle. Il modello Licenza inclusa è supportato solo su Amazon RDS for Oracle Database Standard Edition 2 (SE2).

In questo modello, se disponi di un AWS Support account con assistenza clienti, contatta sia AWS Support per le richieste di assistenza Amazon RDS che Oracle Database. L'utilizzo di RDS for Oracle (l'opzione LI) è soggetto alla Sezione 10.3.1 dei AWS Termini di servizio.

Bring Your Own License (BYOL) per EE e SE2

Nel modello BYOL, è possibile utilizzare le licenze per Oracle Database esistenti per eseguire implementazioni di database su Amazon RDS. Amazon RDS supporta il modello BYOL solo per Oracle Database Enterprise Edition (EE) e Oracle Database Standard Edition 2 (SE2).

Assicurati di disporre della licenza per Oracle Database (con Licenza di aggiornamento software e supporto) adatta alla classe dell'istanza database e all'edizione di Oracle Database che desideri eseguire. È inoltre necessario rispettare le policy di licenza di Oracle per software database Oracle nell'ambiente di cloud computing. Per ulteriori informazioni sulla policy di concessione delle licenze Oracle per Amazon EC2, consulta la pagina relativa alle licenze dei software Oracle nell'ambiente di cloud computing.

Con questo modello di licenza, continui a usare l'account di supporto Oracle attivo e contatti direttamente Oracle per le richieste di servizio per Oracle Database. Se disponi di un AWS Support account con assistenza clienti, puoi contattarci AWS Support per problemi relativi ad Amazon RDS. Amazon Web Services e Oracle offrono una procedura di assistenza multi-fornitore per i casi che richiedono l'intervento di entrambe le organizzazioni.

Integrazione con AWS License Manager

Per semplificare il monitoraggio dell'utilizzo delle licenze Oracle nel modello BYOL, AWS License Manager si integra con Amazon RDS for Oracle. License Manager supporta il monitoraggio delle edizioni del motore RDS for Oracle e i pacchetti di licenze basati su vCPU (virtual core). È inoltre possibile utilizzare License Manager con AWS Organizations per gestire centralmente tutti gli account aziendali.

Nella tabella seguente vengono mostrati i filtri di informazioni sul prodotto di RDS per Oracle.

Filter	Nome	Descrizione
Edizione motore	`oracle-ee`	Oracle Database Enterprise Edition (EE)
	`oracle-se2`	Oracle Database Standard Edition 2 (SE2)
Pacchetto licenza	`data guard`	Vedere Utilizzo di repliche di lettura per Amazon RDS per Oracle (Oracle Active Data Guard)
	`olap`	Per informazioni, consulta Oracle OLAP

Filter	Nome	Descrizione
	`ols`	Per informazioni, consulta Oracle Label Security
	`diagnostic pack sqlt`	Per informazioni, consulta Oracle SQLT
	`tuning pack sqlt`	Consulta la sezione Oracle SQLT

Per tenere traccia dell'utilizzo della licenza delle istanze Oracle DB, puoi creare una licenza autogestita. In questo caso, le risorse RDS for Oracle che corrispondono al filtro delle informazioni sul prodotto vengono automaticamente associate alla licenza autogestita. L'individuazione delle istanze database di Oracle può richiedere fino a 24 ore.

Console

Per creare una licenza autogestita per tenere traccia dell'utilizzo della licenza delle istanze DB Oracle

1. Passare a https://console.aws.amazon.com/license-manager/.
2. Crea una licenza autogestita.

   Per istruzioni, consulta Creare una licenza autogestita nella Guida per l'AWS License Manager utente.

   Aggiungere una regola per un RDS Product Information Filter (Filtro di informazioni sui prodotti RDS) nel pannello Product Information (Informazioni sul prodotto) .

   Per ulteriori informazioni, consulta la ProductInformationsezione AWS License Manager API Reference.

AWS CLI

Per creare una licenza autogestita utilizzando AWS CLI, chiamate il create-license-configurationcomando. Utilizza i parametri `--cli-input-json` o `--cli-input-yaml` per passare i parametri al comando.

Example

L'esempio seguente crea una licenza autogestita per Oracle Enterprise Edition.

```
aws license-manager create-license-configuration --cli-input-json file://rds-oracle-
ee.json
```

Di seguito è riportato il file `rds-oracle-ee.json` di esempio utilizzato.

```
{
    "Name": "rds-oracle-ee",
    "Description": "RDS Oracle Enterprise Edition",
    "LicenseCountingType": "vCPU",
    "LicenseCountHardLimit": false,
    "ProductInformationList": [
        {
            "ResourceType": "RDS",
            "ProductInformationFilterList": [
                {
                    "ProductInformationFilterName": "Engine Edition",
                    "ProductInformationFilterValue": ["oracle-ee"],
                    "ProductInformationFilterComparator": "EQUALS"
                }
            ]
        }
    ]
}
```

Per ulteriori informazioni sul prodotto, consulta la pagina relativa all'individuazione automatica dell'inventario delle risorse nella Guida per l'utente di AWS License Manager .

Per ulteriori informazioni sul *`--cli-input`* parametro, vedere Generazione di AWS CLI scheletro e parametri di input da un file di input JSON o YAML nella Guida per l'utente.AWS CLI

Migrazione tra edizioni Oracle

Se si dispone di una licenza Oracle BYOL inutilizzata adatta all'edizione e alla classe di istanza database che si intende eseguire, è possibile eseguire la migrazione da Standard Edition 2 (SE2) a Enterprise Edition (EE). La migrazione da Enterprise Edition ad altre versioni non è consentita.

Per modificare l'edizione e conservare i dati

1. Creare una snapshot dell'istanza database.

   Per ulteriori informazioni, consulta Creazione di uno snapshot DB per un'istanza DB Single-AZ.

2. Ripristinare la snapshot in una nuova istanza database e selezionare l'edizione del database Oracle da utilizzare.

   Per ulteriori informazioni, consulta Ripristino da uno snapshot database.

3. (Facoltativo) Eliminare l'istanza database precedente, a meno che non la si voglia mantenere in esecuzione e sia disponibile la licenza Oracle Database idonea.

   Per ulteriori informazioni, consulta Eliminazione di un'istanza database.

Opzioni di licenza delle implementazioni Multi-AZ per Oracle

Amazon RDS supporta le implementazioni Multi-AZ per Oracle come soluzione failover a elevata disponibilità. Raccomandiamo Multi-AZ per carichi di lavoro di produzione. Per ulteriori informazioni, consulta Configurazione e gestione di un'implementazione multi-AZ.

Se utilizzi il modello Bring-Your-Own-License (uso di licenze proprie), è necessario disporre di una licenza sia per l'istanza database primaria sia per l'istanza database in standby, in un'implementazione Multi-AZ.

Utenti e privilegi di RDS per Oracle

Quando si crea un'istanza database Amazon RDS per Oracle, l'utente master predefinito dispone della maggior parte delle autorizzazioni utente massime sull'istanza database. Utilizza questo account utente master per qualsiasi attività amministrativa, come la creazione di altri account utente nel database. Poiché RDS è un servizio gestito, non è consentito l'accesso come `SYS` e `SYSTEM` e pertanto non dispone di privilegi `SYSDBA`.

Argomenti

- Limitazioni per i privilegi Oracle DBA
- Come gestire i privilegi su oggetti SYS

Limitazioni per i privilegi Oracle DBA

Nel database un ruolo è una raccolta di privilegi che è possibile concedere o revocare a un utente. Un database Oracle utilizza i ruoli per garantire la sicurezza. Per ulteriori informazioni, consulta Configuring Privilege and Role Authorization nella documentazione del database Oracle.

Il ruolo predefinito DBA normalmente consente tutti i privilegi di amministrazione per un database Oracle. Quando crei un'istanza database utilizzando l'account master, l'account ottiene privilegi DBA, con alcune limitazioni. Per offrire un'esperienza gestita, un database RDS for Oracle non fornisce i seguenti privilegi per il ruolo DBA:

- `ALTER DATABASE`
- `ALTER SYSTEM`
- `CREATE ANY DIRECTORY`
- `DROP ANY DIRECTORY`
- `GRANT ANY PRIVILEGE`
- `GRANT ANY ROLE`

Per ulteriori informazioni sui privilegi di sistema e sui ruoli di RDS per Oracle, consulta Privilegi dell'account utente master.

Come gestire i privilegi su oggetti SYS

È possibile gestire i privilegi su oggetti SYS utilizzando il pacchetto `rdsadmin.rdsadmin_util`. Ad esempio, se si crea l'utente del database `myuser`, è possibile usare la procedura `rdsadmin.rdsadmin_util.grant_sys_object` per concedere privilegi `SELECT` su `V_$SQLAREA` a `myuser`. Per ulteriori informazioni, consulta i seguenti argomenti:

- Concedere privilegi SELECT o EXECUTE agli oggetti SYS
- Revoca del privilegio SELECT o EXECUTE in oggetti SYS
- Concessione di privilegi a utenti non-master

Classi di istanza RDS for Oracle

La capacità di calcolo e memoria di un'istanza RDS for Oracle DB è determinata dalla relativa classe di istanza. La classe di istanza database di cui hai bisogno dipende dalla potenza di elaborazione e dai requisiti di memoria specifici.

Classi di istanza RDS per Oracle supportate

Le classi di istanza RDS per Oracle supportate sono un sottoinsieme delle classi di istanza database RDS. Per un elenco completo delle classi di istanza RDS, consulta Classi di istanze database.

Classi di istanze ottimizzate per la memoria RDS per Oracle

RDS for Oracle offre anche classi di istanze ottimizzate per carichi di lavoro che richiedono memoria, spazio di archiviazione e I/O aggiuntivi per vCPU. Queste classi di istanza utilizzano la seguente convenzione di denominazione:

```
db.r5b.instance_size.tpcthreads_per_core.memratio
db.r5.instance_size.tpcthreads_per_core.memratio
```

Di seguito è riportato un esempio di una classe di istanza supportata:

```
db.r5b.4xlarge.tpc2.mem2x
```

I componenti del nome della classe di istanza precedente sono i seguenti:

- `db.r5b.4xlarge`: il nome della classe di istanza.
- `tpc2`: i thread per core. Il valore 2 indica che il multithreading è attivato. Il valore 1 indica che il multithreading è disattivato.
- `mem2x`: il rapporto tra memoria aggiuntiva e memoria standard per la classe di istanza. In questo esempio, l'ottimizzazione fornisce il doppio della memoria di un'istanza standard db.r5.4xlarge.

Edizione, classe di istanza e combinazioni di licenza supportate in RDS for Oracle

Se utilizzi la console RDS, puoi scoprire se un'edizione, una classe di istanza e una combinazione di licenza specifiche sono supportate scegliendo Crea database e specificando un'opzione diversa. In AWS CLI, puoi eseguire il seguente comando:

```
aws rds describe-orderable-db-instance-options --engine engine-type --license-
model license-type
```

La tabella seguente elenca tutte le edizioni, le classi di istanze e i tipi di licenza supportati per RDS for Oracle. Oracle Database 12c Release 1 (12.1.0.2) e Oracle Database 12c Release 2 (12.2.0.2) sono elencati nella tabella, ma il supporto per queste release è obsoleto. Per informazioni sugli attributi di memoria di ogni tipo, consulta RDS for Oracle instance types (Tipi di istanza di Amazon RDS per Oracle). Per informazioni sui prezzi, consulta i modelli di prezzo di Amazon RDS for Oracle.

Edizione Oracle	Oracle Database 19c e versioni successive, Oracle Database 12c Release 2 (12.2.0.1) (obsoleto)	Oracle Database 12c Release 1 (12.1.0.2) (obsoleto)
Enterprise Edition (EE) Modello di licenza Bring Your Own License (BYOL)	Classi di istanza Standard	
	db.m6i.large–db.m6i.32xlarge (solo 19c) db.m5d.large–db.m5d.24xlarge db.m5.large–db.m5.24xlarge	db.m5.large–db.m5.24xlarge
	Classi di istanza ottimizzata per la memoria	
	db.r6i.large–db.r6i.32xlarge (solo 19c) db.r5d.large–db.r5d.24xlarge db.r5b.8xlarge.tpc2.mem3x db.r5b.6xlarge.tpc2.mem4x db.r5b.4xlarge.tpc2.mem4x db.r5b.4xlarge.tpc2.mem3x db.r5b.4xlarge.tpc2.mem2x db.r5b.2xlarge.tpc2.mem8x db.r5b.2xlarge.tpc2.mem4x db.r5b.2xlarge.tpc1.mem2x db.r5b.xlarge.tpc2.mem4x db.r5b.xlarge.tpc2.mem2x db.r5b.large.tpc1.mem2x db.r5b.large–db.r5b.24xlarge	db.r5.12xlarge.tpc2.mem2x db.r5b.large–db.r5b.24xlarge db.r5.8xlarge.tpc2.mem3x db.r5.6xlarge.tpc2.mem4x db.r5.4xlarge.tpc2.mem4x db.r5.4xlarge.tpc2.mem3x db.r5.4xlarge.tpc2.mem2x db.r5.2xlarge.tpc2.mem8x db.r5.2xlarge.tpc2.mem4x db.r5.2xlarge.tpc1.mem2x db.r5.xlarge.tpc2.mem4x db.r5.xlarge.tpc2.mem2x db.r5.large.tpc1.mem2x db.r5.large–db.r5.24xlarge

Edizione Oracle	Oracle Database 19c e versioni successive, Oracle Database 12c Release 2 (12.2.0.1) (obsoleto)	Oracle Database 12c Release 1 (12.1.0.2) (obsoleto)
	db.r5.12xlarge.tpc2.mem2x db.r5.8xlarge.tpc2.mem3x db.r5.6xlarge.tpc2.mem4x db.r5.4xlarge.tpc2.mem4x db.r5.4xlarge.tpc2.mem3x db.r5.4xlarge.tpc2.mem2x db.r5.2xlarge.tpc2.mem8x db.r5.2xlarge.tpc2.mem4x db.r5.2xlarge.tpc1.mem2x db.r5.xlarge.tpc2.mem4x db.r5.xlarge.tpc2.mem2x db.r5.large.tpc1.mem2x db.r5.large–db.r5.24xlarge db.x2iedn.xlarge - db.x2iedn.32xlarge db.x2iezn.2xlarge - db.x2iezn.12xlarge db.x2idn.16xlarge - db.x2idn.32xlarge db.x1e.xlarge–db.x1e.32xlarge db.x1.16xlarge–db.x1.32xlarge db.z1d.large–db.z1d.12xlarge	db.x1e.xlarge–db.x1e.32xlarge db.x1.16xlarge–db.x1.32xlarge db.z1d.large–db.z1d.12xlarge
	Classi di istanza espandibile	

Edizione Oracle	Oracle Database 19c e versioni successive, Oracle Database 12c Release 2 (12.2.0.1) (obsoleto)	Oracle Database 12c Release 1 (12.1.0.2) (obsoleto)
	db.t3.small–db.t3.2xlarge	db.t3.micro–db.t3.2xlarge
Standard Edition 2 (SE2) Modello di licenza Bring Your Own License (BYOL)	Classi di istanza Standard	
	db.m6i.large—db.m6i.4xlarge db.m5d.large–db.m5d.4xlarge db.m5.large–db.m5.4xlarge	db.m5.large–db.m5.4xlarge
	Classi di istanza ottimizzata per la memoria	

Edizione Oracle	Oracle Database 19c e versioni successive, Oracle Database 12c Release 2 (12.2.0.1) (obsoleto)	Oracle Database 12c Release 1 (12.1.0.2) (obsoleto)
	db.r6i.large–db.r6i.4xlarge (solo 19c)	db.r5.4xlarge.tpc2.mem4x
	db.r5d.large–db.r5d.4xlarge	db.r5.4xlarge.tpc2.mem3x
	db.r5.4xlarge.tpc2.mem4x	db.r5.4xlarge.tpc2.mem2x
	db.r5.4xlarge.tpc2.mem3x	db.r5.2xlarge.tpc2.mem8x
	db.r5.4xlarge.tpc2.mem2x	db.r5.2xlarge.tpc2.mem4x
	db.r5.2xlarge.tpc2.mem8x	db.r5.2xlarge.tpc1.mem2x
	db.r5.2xlarge.tpc2.mem4x	db.r5.xlarge.tpc2.mem4x
	db.r5.2xlarge.tpc1.mem2x	db.r5.xlarge.tpc2.mem2x
	db.r5.xlarge.tpc2.mem4x	db.r5.large.tpc1.mem2x
	db.r5.xlarge.tpc2.mem2x	db.r5.large–db.r5.4xlarge
	db.r5.large.tpc1.mem2x	db.r5b.large–db.r5b.4xlarge
	db.r5.large–db.r5.4xlarge	db.z1d.large–db.z1d.3xlarge
	db.r5b.large–db.r5b.4xlarge	
	db.x2iedn.xlarge - db.x2iedn.4xlarge	
	db.x2iezn.2xlarge - db.x2iezn.4xlarge	
	db.z1d.large–db.z1d.3xlarge	
	Classi di istanza espandibile	
	db.t3.small–db.t3.2xlarge	db.t3.micro–db.t3.2xlarge

Edizione Oracle	Oracle Database 19c e versioni successive, Oracle Database 12c Release 2 (12.2.0.1) (obsoleto)	Oracle Database 12c Release 1 (12.1.0.2) (obsoleto)
Standard Edition 2 (SE2) Licenza inclusa	Classi di istanze Standard	
	db.m5.large–db.m5.4xlarge	db.m5.large–db.m5.4xlarge
	Classi di istanza ottimizzata per la memoria	
	db.r6i.large–db.r6i.4xlarge (solo 19c) db.r5.large–db.r5.4xlarge	db.r5.large–db.r5.4xlarge
	Classi di istanza espandibile	
	db.t3.small–db.t3.2xlarge	db.t3.micro–db.t3.2xlarge

> **Note**
>
> Incoraggiamo tutti i clienti che utilizzano la propria licenza a consultare il proprio accordo di licenza per valutare l'impatto delle deprecazioni di Amazon RDS for Oracle. Per ulteriori informazioni sulla capacità di calcolo delle classi di istanze database supportate da RDS per Oracle, consulta Classi di istanze database e Configurazione del processore per una classe di istanza database in RDS per Oracle.

> **Note**
>
> Se hai snapshot DB di istanze database che usano classi di istanza database obsolete, puoi scegliere una classe di istanza database non obsoleta quando ripristini gli snapshot DB. Per ulteriori informazioni, consulta Ripristino da uno snapshot database.

Classi di istanze RDS per Oracle DB obsolete

Di seguito sono riportate le classi di istanza database obsolete per RDS per Oracle:

- db.m1, db.m2, db.m3, db.m4

- db.t3.micro (supportata solo su 12.1.0.2, che è obsoleta)
- db.t1, db.t2
- db.r1, db.r2, db.r3, db.r4

Queste classi di istanza database sono state sostituite da classi di istanza database con prestazioni migliori, di solito disponibili a un costo inferiore. Se le istanze database usano classi di istanza database deprecate, hai le seguenti opzioni:

- Consenti ad Amazon RDS di modificare automaticamente ogni istanza database per utilizzare una classe di istanza database equivalente non deprecata. Per le tempistiche di deprecazione, consulta Tipi di classi di istanza database.
- Cambia autonomamente la classe di istanza database modificando l'istanza database. Per ulteriori informazioni, consulta Modifica di un'istanza database Amazon RDS.

Se hai snapshot DB di istanze database che usano classi di istanza database obsolete, puoi scegliere una classe di istanza database non obsoleta quando ripristini gli snapshot DB. Per ulteriori informazioni, consulta Ripristino da uno snapshot database.

Architettura del database RDS per Oracle

L'architettura multitenant Oracle, nota anche come architettura CDB, consente a un database Oracle di funzionare come database container multitenant (CDB). Un CDB può includere database integrabili creati dal cliente (PDB). Un database non CDB è un database Oracle che utilizza l'architettura tradizionale, che non può contenere PDB. Per ulteriori informazioni sull'architettura multitenant, consulta Guida per l'amministratore multitenant Oracle.

Per Oracle Database 19c e versioni successive, è possibile creare un'istanza database RDS per Oracle che utilizza l'architettura CDB. Le applicazioni client si connettono a livello di PDB anziché a livello di CDB. RDS per Oracle supporta le seguenti configurazioni dell'architettura CDB:

Configurazione multi-tenant

Questa funzionalità della piattaforma RDS consente a un'istanza RDS for Oracle CDB di contenere da 1 a 30 database tenant, a seconda dell'edizione del database e di qualsiasi opzione richiesta per le licenze dei database tenant (PDB). La configurazione multi-tenant non supporta i PDB di applicazioni o i PDB di proxy. È possibile utilizzare le API RDS per aggiungere, modificare e rimuovere i database del tenant.

> **Note**
>
> La funzionalità Amazon RDS è chiamata "multi-tenant" anziché "multitenant" perché è una funzionalità della piattaforma RDS, non solo del motore di database Oracle. Il termine "Oracle multitenant" si riferisce esclusivamente all'architettura del database Oracle, che è compatibile sia con le implementazioni on-premise che con quelle RDS.

Configurazione a tenant singolo

Questa funzionalità della piattaforma RDS limita un'istanza RDS for Oracle CDB a 1 database tenant (PDB). Non è possibile aggiungere altri PDB utilizzando le API RDS. La configurazione a tenant singolo utilizza le stesse API RDS dell'architettura non CDB. Pertanto, l'esperienza di utilizzo di un CDB nella configurazione a tenant singolo è per lo più la stessa di un non CDB.

È possibile convertire un CDB che utilizza la configurazione single-tenant in una configurazione multi-tenant, in modo da aggiungere PDB al CDB. Questa modifica dell'architettura è permanente e irreversibile. Per ulteriori informazioni, consulta Conversione della configurazione a tenant singolo in multi-tenant.

> **Note**
>
> Non è possibile accedere al CDB stesso.

In Oracle Database 21c e versioni successive, tutti i database sono CDB. È invece possibile creare un'istanza database Oracle Database 19c come CDB o non CDB. Non è possibile aggiornare un database non CDB a database CDB, ma è possibile convertire un database Oracle Database 19c non CDB in database CDB e quindi aggiornarlo. Non è possibile convertire un database CDB in un database non CDB.

Per ulteriori informazioni, consulta le seguenti risorse:

- Utilizzo di database CDB per RDS per Oracle
- Limitazioni per i CDB RDS per Oracle
- Creazione di un'istanza database Amazon RDS

Parametri di RDS for Oracle

Gruppi di parametri database

In Amazon RDS, gestisci i parametri utilizzando gruppi di parametri DB. Per ulteriori informazioni, consulta Utilizzo di gruppi di parametri. Per visualizzare i parametri di inizializzazione supportati per un'edizione e una versione specifiche di Oracle Database, esegui il AWS CLI comando. `describe-engine-default-parameters`

Ad esempio, per visualizzare i parametri di inizializzazione supportati per l'Enterprise Edition di Oracle Database 19c, eseguire il comando seguente.

```
aws rds describe-engine-default-parameters \
    --db-parameter-group-family oracle-ee-19
```

Parametri di inizializzazione del database Oracle

Per trovare la documentazione per i parametri di inizializzazione, vedere Parametri di inizializzazione nella documentazione del database Oracle. I seguenti parametri di inizializzazione hanno considerazioni speciali:

- `ARCHIVE_LAG_TARGET`

  Questo parametro forza un redo log switch allo scadere del tempo specificato. In RDS for Oracle, `ARCHIVE_LAG_TARGET` è impostato su `300` perché il Recovery Point Objective (RPO) è di 5 minuti. Per raggiungere questo obiettivo, RDS per Oracle cambia il redo log online ogni 5 minuti e lo archivia in un bucket Amazon S3. Se la frequenza del log switch causa un problema di prestazioni per il database RDS for Oracle, puoi scalare l'istanza DB e lo storage su uno con IOPS e throughput più elevati. In alternativa, se utilizzi RDS Custom for Oracle o distribuisci un database Oracle su Amazon EC2, puoi modificare l'impostazione del parametro di `ARCHIVE_LAG_TARGET` inizializzazione.

Set di caratteri RDS for Oracle

RDS per Oracle supporta due tipi di set di caratteri: il set di caratteri DB e il set di caratteri nazionale.

Set di caratteri DB

Il set di caratteri del database Oracle viene utilizzato nei tipi di dati `CHAR`, `VARCHAR2` e `CLOB`. Il database utilizza anche questo set di caratteri per metadati quali nomi di tabelle, nomi di colonne e istruzioni SQL. Il set di caratteri del database Oracle viene in genere indicato come set di caratteri DB.

L'utente imposta il set di caratteri al momento della creazione di un'istanza database. Non è possibile modificare il set di caratteri DB dopo aver creato il database.

Set di caratteri DB supportati

Nella tabella seguente sono elencati i set di caratteri di database Oracle supportati in Amazon RDS. Puoi utilizzare un valore di questa tabella con il parametro `--character-set-name` del comando della AWS CLI create-db-instance o con il parametro `CharacterSetName` dell'operazione API Amazon RDS CreateDBInstance.

> **Note**
>
> Il set di caratteri per un CDB è sempre AL32UTF8. È possibile impostare un set di caratteri diverso solo per il PDB.

Valore	Descrizione
AL32UTF8	Set di caratteri Unicode 5.0 UTF-8 Universal (predefinito)
AR8ISO8859P6	ISO 8859-6 latino/arabo
AR8MSWIN1256	Tabella codici di Microsoft Windows 1256 8-bit latino/arabo
BLT8ISO8859P13	ISO 8859-13 lingue baltiche
BLT8MSWIN1257	Tabella codici di Microsoft Windows 1257 8-bit lingue baltiche
CL8ISO8859P5	ISO 88559-5 latino/cirillico

Valore	Descrizione
CL8MSWIN1251	Tabella codici di Microsoft Windows 1251 8-bit latino/cirillico
EE8ISO8859P2	ISO 8859-2 Europa orientale
EL8ISO8859P7	ISO 8859-7 latino/greco
EE8MSWIN1250	Tabella codici di Microsoft Windows 1250 8-bit Europa orientale
EL8MSWIN1253	Tabella codici di Microsoft Windows 1253 8-bit latino/greco
IW8ISO8859P8	ISO 8859-8 latino/ebraico
IW8MSWIN1255	Tabella codici di Microsoft Windows 1255 8-bit latino/ebraico
JA16EUC	EUC 24-bit giapponese
JA16EUCTILDE	Uguale a JA16EUC ad eccezione della mappatura di trattino ondulato e tilde da e verso Unicode
JA16SJIS	Shift-JIS 16-bit giapponese
JA16SJISTILDE	Uguale a JA16SJIS ad eccezione della mappatura di trattino ondulato e tilde da e verso Unicode
KO16MSWIN949	Tabella codici di Microsoft Windows 949 coreano
NE8ISO8859P10	ISO 8859-10 Europa settentrionale
NEE8ISO8859P4	ISO 8859-4 Europa settentrionale e nord-orie ntale

Valore	Descrizione
TH8TISASCII	Thai Industrial Standard 620-2533-ASCII 8-bit
TR8MSWIN1254	Tabella codici di Microsoft Windows 1254 8-bit turco
US7ASCII	ASCII 7-bit americano
UTF8	Set di caratteri Unicode 3.0 UTF-8 Universal, compatibile con CESU-8
VN8MSWIN1258	Tabella codici di Microsoft Windows 1258 8-bit vietnamita
WE8ISO8859P1	Europa occidentale 8-bit ISO 8859 Parte 1
WE8ISO8859P15	ISO 8859-15 Europa occidentale
WE8ISO8859P9	ISO 8859-9 Europa occidentale e turco
WE8MSWIN1252	Tabella codici di Microsoft Windows 1252 8-bit Europa occidentale
ZHS16GBK	GBK 16-bit cinese semplificato
ZHT16HKSCS	Tabella codici di Microsoft Windows 950 con set di caratteri supplementare di Hong Kong HKSCS-2001. La conversione del set di caratteri è basata su Unicode 3.0.
ZHT16MSWIN950	Tabella codici di Microsoft Windows 950 cinese tradizionale
ZHT32EUC	EUC 32-bit cinese tradizionale

Variabile di ambiente NLS_LANG

Un locale è un insieme di informazioni che riguardano i requisiti linguistici e culturali che corrispondono a una determinata lingua e paese. L'impostazione del parametro di ambiente

NLS_LANG nell'ambiente del client è il modo più semplice per specificare il comportamento delle impostazioni locali per Oracle. Questo parametro imposta la lingua e il paese utilizzati dall'applicazione client e dal server di database. Indica inoltre il set di caratteri del client, che corrisponde al set di caratteri per i dati immessi o visualizzati da un'applicazione client. Per ulteriori informazioni su NLS_LANG e sui set di caratteri, consulta la descrizione di un set di caratteri o di una tabella codici nella documentazione di Oracle.

Parametri di inizializzazione del sistema

Puoi anche impostare i parametri di inizializzazione NLS (National Language Support) seguenti a livello di istanza per un'istanza database Oracle in Amazon RDS:

- NLS_DATE_FORMAT
- NLS_LENGTH_SEMANTICS
- NLS_NCHAR_CONV_EXCP
- NLS_TIME_FORMAT
- NLS_TIME_TZ_FORMAT
- NLS_TIMESTAMP_FORMAT
- NLS_TIMESTAMP_TZ_FORMAT

Per ulteriori informazioni sulla modifica dei parametri di un'istanza, consulta Utilizzo di gruppi di parametri.

Puoi impostare altri parametri di inizializzazione NLS nel client SQL. L'istruzione seguente imposta ad esempio il parametro di inizializzazione NLS_LANGUAGE su GERMAN in un client SQL connesso a un'istanza database Oracle:

```
ALTER SESSION SET NLS_LANGUAGE=GERMAN;
```

Per informazioni sulla connessione a un'istanza database Oracle con un client SQL, consulta Connessione all'istanza database RDS per Oracle.

Set di caratteri nazionali

Il set di caratteri nazionali viene utilizzato nei tipi di dati `NCHAR`, `NVARCHAR2` e `NCLOB`. Il set di caratteri nazionale è in genere indicato come set di caratteri NCHAR. A differenza del set di caratteri DB, il set di caratteri NCHAR non influisce sui metadati del database.

Il set di caratteri NCHAR supporta i seguenti set di caratteri:

- AL16UTF16 (impostazione predefinita)
- UTF8

Puoi specificare entrambi i valori con il parametro `--nchar-character-set-name` del comando create-db-instance (solo AWS CLI versione 2). Se utilizzi l'API Amazon RDS, specifica il parametro `NcharCharacterSetName` dell'operazione CreateDBInstance. Non è possibile modificare il set di caratteri nazionali dopo aver creato il database.

Per ulteriori informazioni su Unicode nei database Oracle, vedere Supporto di database multilingue con unicode nella documentazione di Oracle.

Limitazioni di RDS for Oracle

Nelle sezioni seguenti, sono disponibili importanti limitazioni all'utilizzo di RDS per Oracle. Per le limitazioni specifiche dei CDB, consulta Limitazioni per i CDB RDS per Oracle.

> **Note**
>
> L'elenco non è completo.

Argomenti

- Limiti delle dimensioni dei file Oracle in Amazon RDS
- Sinonimi pubblici per gli schemi forniti da Oracle
- Schemi per funzionalità non supportate
- Limitazioni per i privilegi Oracle DBA
- Obsolescenza di TLS 1.0 e 1.1 Transport Layer Security

Limiti delle dimensioni dei file Oracle in Amazon RDS

La dimensione massima di un file nelle istanze database RDS per Oracle è pari a 16 TiB (tebibyte). Questo limite è imposto dal file system ext4 utilizzato dall'istanza. Pertanto, i file di dati Oracle con tablespace a file unico sono limitati a 16 TiB. Se si tenta di ridimensionare un file di dati in uno spazio tabella bigfile a un valore superiore al limite, viene visualizzato un errore analogo al seguente:

```
ORA-01237: cannot extend datafile 6
ORA-01110: data file 6: '/rdsdbdata/db/mydir/datafile/myfile.dbf'
ORA-27059: could not reduce file size
Linux-x86_64 Error: 27: File too large
Additional information: 2
```

Sinonimi pubblici per gli schemi forniti da Oracle

Non creare o modificare sinonimi pubblici per schemi gestiti da Oracle, tra cui `SYS`, `SYSTEM` e `RDSADMIN`. Queste operazioni potrebbero comportare l'invalidazione di componenti essenziali del database e influire sulla disponibilità dell'istanza database.

È possibile creare sinonimi pubblici che fanno riferimento a oggetti nei propri schemi.

Schemi per funzionalità non supportate

In generale, Amazon RDS non impedisce la creazione di schemi per feature non supportate. Tuttavia, se si creano schemi per feature e componenti Oracle che richiedono privilegi SYS, è possibile danneggiare il dizionario dati e influire sulla disponibilità dell'istanza. Utilizzare solo le funzioni e gli schemi supportati disponibili in Aggiunta di opzioni alle istanze database Oracle.

Limitazioni per i privilegi Oracle DBA

Nel database un ruolo è una raccolta di privilegi che è possibile concedere o revocare a un utente. Un database Oracle utilizza i ruoli per garantire la sicurezza.

Il ruolo predefinito `DBA` normalmente consente tutti i privilegi di amministrazione per un database Oracle. Quando crei un'istanza database utilizzando l'account master, l'account ottiene privilegi DBA, con alcune limitazioni. Per offrire un'esperienza gestita, un database RDS for Oracle non fornisce i seguenti privilegi per il ruolo `DBA`:

- `ALTER DATABASE`
- `ALTER SYSTEM`
- `CREATE ANY DIRECTORY`
- `DROP ANY DIRECTORY`
- `GRANT ANY PRIVILEGE`
- `GRANT ANY ROLE`

Utilizza questo account principale per qualsiasi attività amministrativa come la creazione di altri account utente nel database. Non è possibile utilizzare l'account `SYS`, `SYSTEM` o altri account amministrativi forniti.

Obsolescenza di TLS 1.0 e 1.1 Transport Layer Security

Le versioni 1.0 e 1.1 del protocollo Transport Layer Security (TLS 1.0 e TLS 1.1) sono obsolete. In conformità alle best practice di sicurezza, Oracle ha reso obsoleto l'uso di TLS 1.0 e TLS 1.1. Per soddisfare i requisiti di sicurezza, RDS for Oracle consiglia caldamente di utilizzare invece TLS 1.2.

Connessione all'istanza database RDS per Oracle

Dopo che Amazon RDS effettua il provisioning dell'istanza database Oracle, è possibile usare qualsiasi applicazione client SQL standard per accedere all'istanza database. Poiché RDS è un servizio gestito, non è possibile accedere come SYS o SYSTEM. Per ulteriori informazioni, consulta Utenti e privilegi di RDS per Oracle.

In questo argomento, viene illustrato come utilizzare Oracle SQL Developer o SQL*Plus per connettersi a un'istanza database RDS per Oracle. Per un esempio che illustra il processo di creazione e di connessione a un'istanza database di esempio, consulta Creazione e connessione a un'istanza database Oracle.

Argomenti

Esito dell'endpoint dell'istanza database RDS per Oracle

Ogni istanza database Amazon RDS dispone di un endpoint e ciascun endpoint è associato a un nome DNS e a un numero di porta per l'istanza database. Per connetterti all'istanza database tramite un'applicazione client SQL, devi conoscere il nome DNS e il numero di porta dell'istanza database.

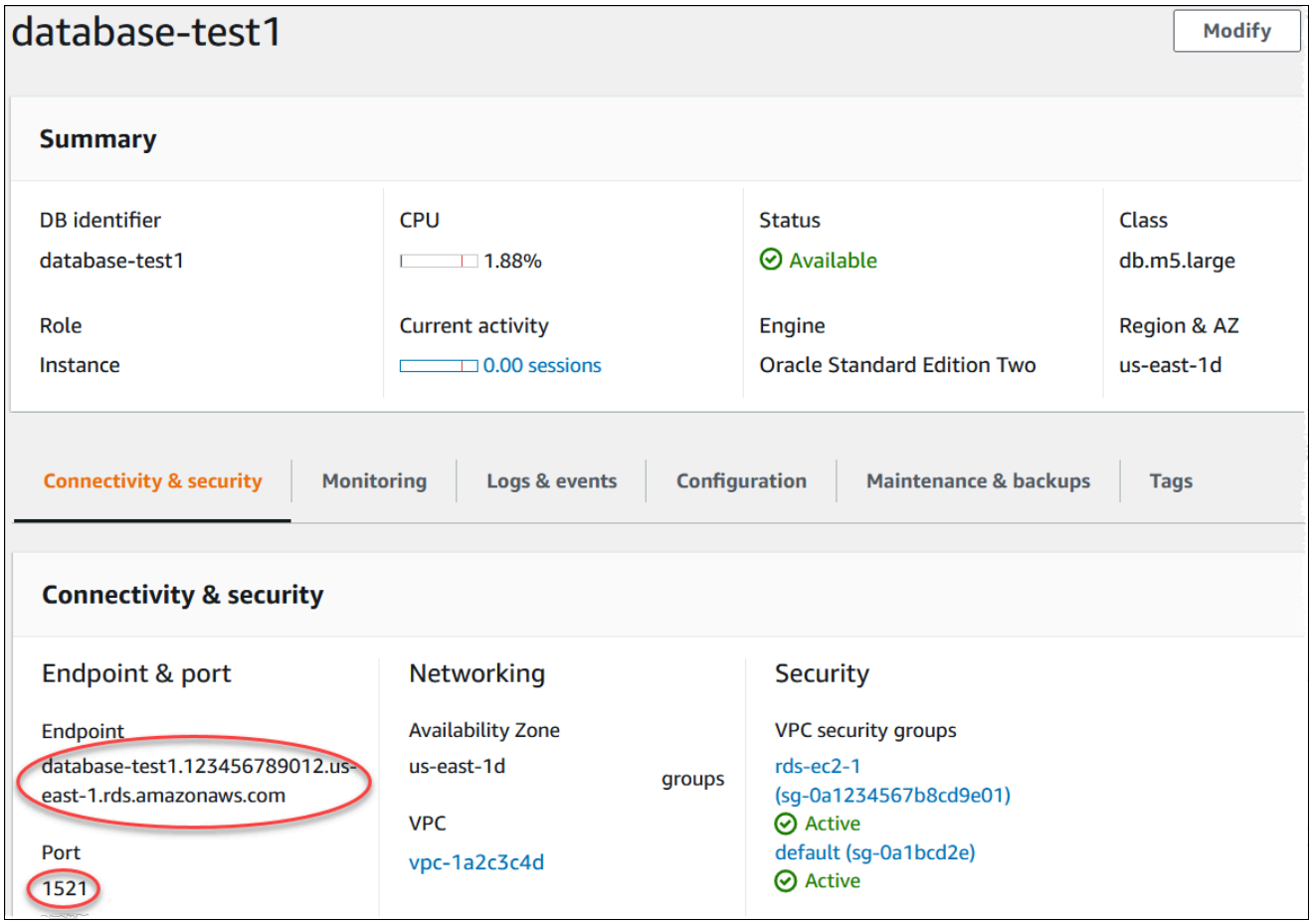
Per trovare l'endpoint di un'istanza database, puoi usare la console Amazon RDS o la AWS CLI.

> **Note**
>
> Se stai utilizzando Autenticazione Kerberos, consulta Connessione a Oracle con Autenticazione Kerberos.

Console

Per trovare l'endpoint tramite la console

1. Accedi alla AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nell'angolo superiore destro della console, scegliere la regione AWS dell'istanza database.
3. Trovare il nome DNS e il numero della porta per l'istanza database.
   a. Scegliere Databases (Database) per visualizzare un elenco di istanze database.
   b. Selezionare l'istanza database Oracle per visualizzare i dettagli dell'istanza.
   c. Nella scheda Connectivity & security (Connettività e sicurezza), copiare l'endpoint. Annotare anche il numero di porta. L'endpoint e il numero di porta sono necessari per la connessione all'istanza database.



AWS CLI

Per trovare l'endpoint di un'istanza database Oracle tramite AWS CLI, chiamare il comando describe-db-instances.

Example Per trovare l'endpoint tramite AWS CLI

```
aws rds describe-db-instances
```

Cercare `Endpoint` nell'output per trovare il nome DNS e il numero di porta dell'istanza database. La riga `Address` nell'output contiene il nome DNS. Di seguito è riportato un esempio di output dell'endpoint JSON.

```
"Endpoint": {
    "HostedZoneId": "Z1PVIF0B656C1W",
    "Port": 3306,
    "Address": "myinstance.123456789012.us-west-2.rds.amazonaws.com"
},
```

> **Note**
>
> L'output potrebbe contenere informazioni per più istanze database.

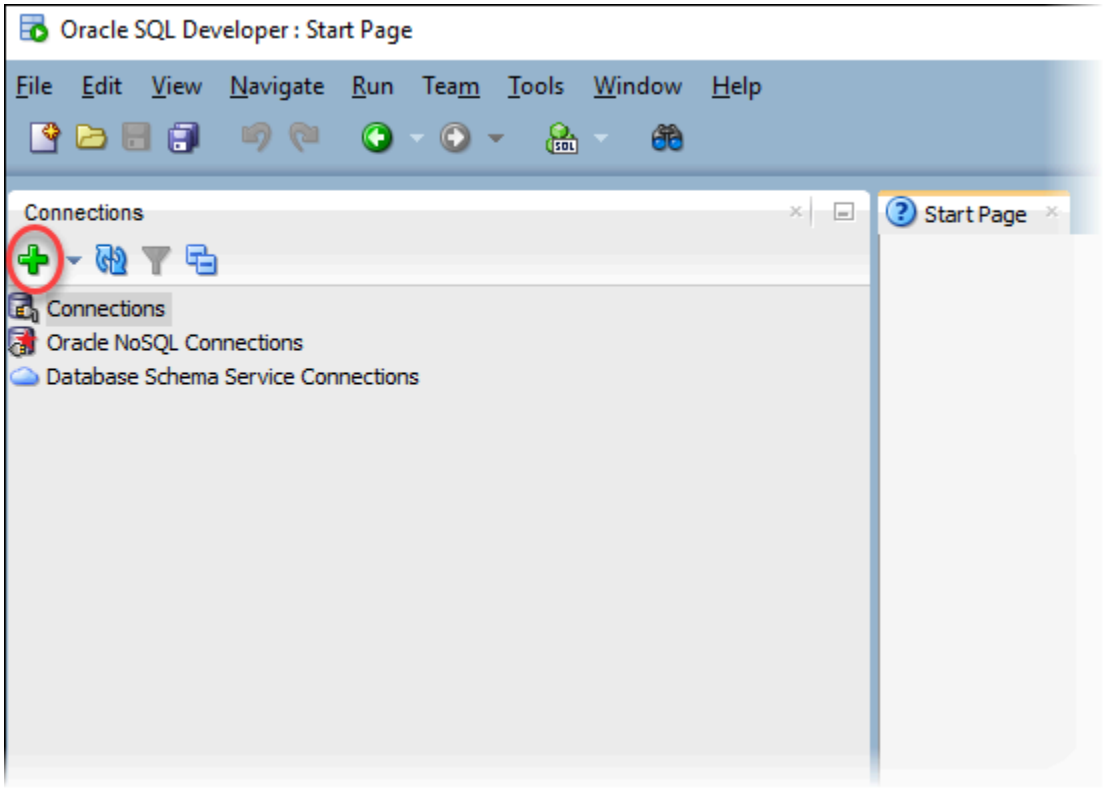
Connessione all'istanza database tramite Oracle SQL Developer

In questa procedura eseguirai la connessione all'istanza database tramite Oracle SQL Developer. Per scaricare una versione autonoma di questa utility, consulta la pagina per i download di Oracle SQL Developer.

Per connetterti alla tua istanza database, dovrai disporre del relativo DNS e del numero di porta. Per informazioni su come trovare il nome DNS e il numero di porta di un'istanza database, consulta Esito dell'endpoint dell'istanza database RDS per Oracle.

Per eseguire la connessione a un'istanza database tramite SQL Developer

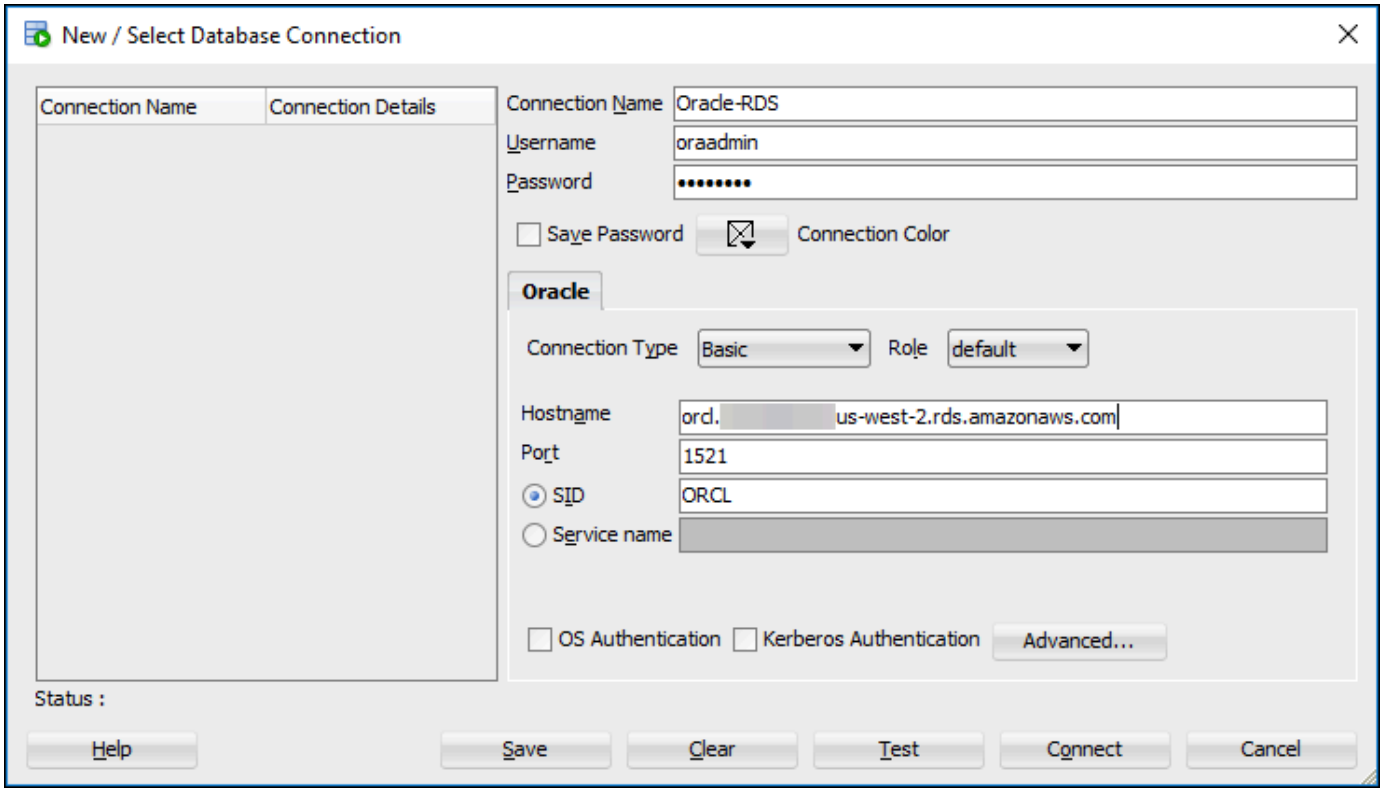
1. Avviare Oracle SQL Developer.
2. Nella scheda Connections (Connessioni) scegliere l'icona add (+) (aggiungi +).



3. Nella finestra di dialogo New/Select Database Connection (Nuova/Seleziona connessione database) fornire le informazioni per l'istanza database:

   - Per Connection Name (Nome connessione) inserire un nome che descriva la connessione, ad esempio `Oracle-RDS`.
   - Per Username (Nome utente) inserire il nome dell'amministratore di database per l'istanza database.
   - Per Password inserire la password per l'amministratore di database.
   - Per Hostname (Nome host) inserire il nome DNS dell'istanza database.
   - Per Port (Porta) inserire il numero di porta.
   - Per SID, immetti il nome del database. Puoi trovare il nome del database nella scheda Configuration (Configurazione) della pagina dei dettagli del database.

   La finestra di dialogo completata si presenta in maniera analoga a quanto segue.

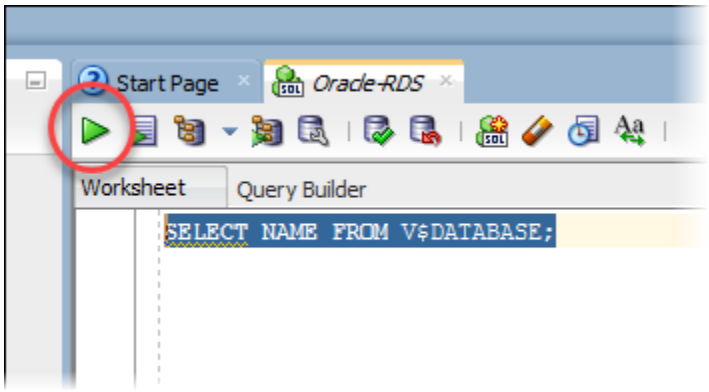


4. Scegliere Connetti.
5. A questo punto, puoi iniziare a creare database personali ed eseguire normalmente query sull'istanza e sui database. Per eseguire una query di test dell'istanza database, utilizzare la seguente procedura:

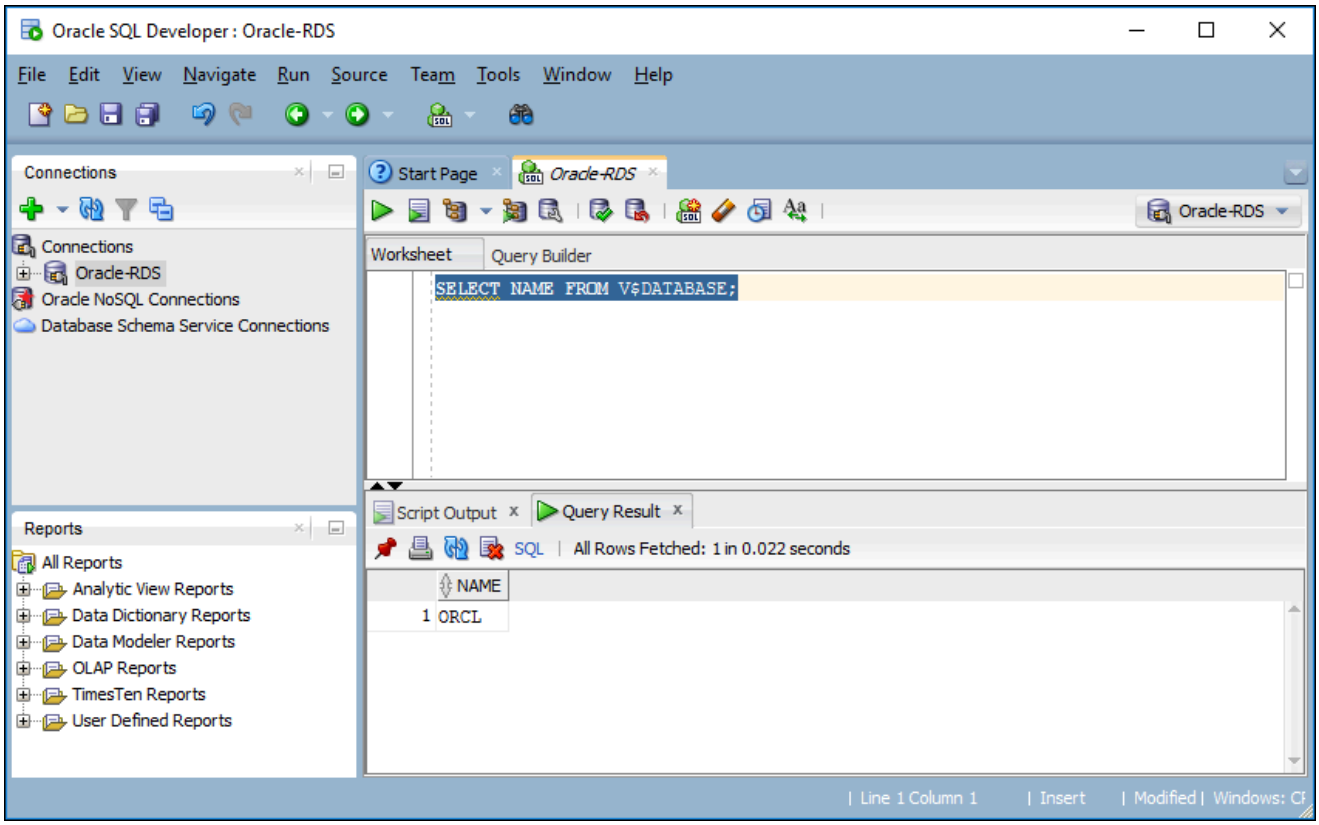
   a. Nella scheda Worksheet (Foglio di lavoro) della connessione inserire la query SQL seguente.

   ```
   SELECT NAME FROM V$DATABASE;
   ```

   b. Selezionare l'icona execute (esegui) per eseguire la query.



SQL Developer restituisce il nome del database.



Connessione all'istanza database tramite SQL*Plus

È possibile utilizzare un'utility come SQL*Plus per connettersi a un'istanza database Amazon RDS che esegue Oracle. Per scaricare Oracle Instant Client, che include una versione autonoma di SQL*Plus, consulta Download di Oracle Instant Client.

Per connetterti alla tua istanza database, dovrai disporre del relativo DNS e del numero di porta. Per informazioni su come trovare il nome DNS e il numero di porta di un'istanza database, consulta Esito dell'endpoint dell'istanza database RDS per Oracle.

Example Per eseguire la connessione a un'istanza database Oracle tramite SQL*Plus

Negli esempi seguenti sostituisci il nome utente dell'amministratore dell'istanza database e il nome DNS dell'istanza database, quindi includi il numero di porta e il SID Oracle. Il valore SID è il nome del database dell'istanza database specificato quando si crea l'istanza database, non il nome dell'istanza database.

Per LinuxmacOS, oUnix:

```
sqlplus 'user_name@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=dns_name)(PORT=port))
(CONNECT_DATA=(SID=database_name)))'
```

Per Windows:

```
sqlplus user_name@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=dns_name)(PORT=port))
(CONNECT_DATA=(SID=database_name)))
```

Verrà visualizzato un output simile al seguente.

```
SQL*Plus: Release 12.1.0.2.0 Production on Mon Aug 21 09:42:20 2017
```

Dopo l'immissione della password dell'utente, verrà visualizzato il prompt SQL.

```
SQL>
```

> **Note**
>
> Il formato più breve della stringa di connessione (EZ Connect), ad esempio `sqlplus USER/PASSWORD@longer-than-63-chars-rds-endpoint-here:1521/database-identifier`, può incorrere nel limite relativo al numero massimo di caratteri, pertanto si consiglia di non utilizzarlo per la connessione.

Considerazioni per i gruppi di sicurezza

Affinché possa eseguire la connessione all'istanza database, è necessaria l'associazione a un gruppo di sicurezza che contiene gli indirizzi IP e la configurazione di rete richiesti. L'istanza database potrebbe utilizzare il gruppo di sicurezza predefinito. Se hai assegnato un gruppo di sicurezza predefinito non configurato quando hai creato l'istanza database, il firewall dell'istanza database impedisce le connessioni. Per ulteriori informazioni sulla creazione di un nuovo gruppo di sicurezza, consulta Controllo dell'accesso con i gruppi di sicurezza.

Dopo aver creato il nuovo gruppo di sicurezza, modifica l'istanza database per associarla al gruppo di sicurezza. Per ulteriori informazioni, consulta Modifica di un'istanza database Amazon RDS.

Puoi aumentare la sicurezza utilizzando la crittografia SSL per proteggere le connessioni alla tua istanza database. Per ulteriori informazioni, consultare Oracle Secure Sockets Layer.

Considerazioni sull'architettura del processo

I processi server gestiscono le connessioni utente a un'istanza database Oracle. Per impostazione predefinita, l'istanza database Oracle utilizza processi server dedicati. In questo caso, ciascun processo server gestisce un solo processo utente. Puoi anche configurare processi server condivisi. In tal caso, ciascun processo server può gestire più processi utente.

Potresti considerare di utilizzare i processi server condivisi quando un elevato numero di sessioni utente utilizza un'eccessiva quantità di memoria sul server oppure in caso di connessioni e disconnessioni molto frequenti delle sessioni, con conseguente riduzione delle prestazioni. L'uso dei processi server condivisi presenta anche alcuni svantaggi, in quanto possono ad esempio implicare un eccessivo utilizzo delle risorse CPU e sono più complessi da configurare e gestire.

Per ulteriori informazioni, consulta About Dedicated and Shared Server Processes nella documentazione Oracle. Per ulteriori informazioni sulla configurazione dei processi server condivisi in un'istanza database RDS for Oracle, consulta Configurazione di Amazon RDS for Oracle Database per l'uso con server condivisi nel Knowledge Center.

Risoluzione dei problemi relativi alle connessioni all'istanza database Oracle

Quando tenti di connetterti all'istanza database Oracle, è possibile che si verifichino i seguenti errori.

Problema	Suggerimenti sulla risoluzione dei problemi
Unable to connect to your DB instance (Impossib ile connettersi all'istanza database)	Per una nuova istanza database creata, l'istanza database avrà lo stato creating (in creazione) fino a quando sarà pronta per essere impiegata. Quando lo stato cambia in available (disponibile) è possibile connettersi all'istanza database. A seconda della classe di istanza database e della quantità di storage, prima che la nuova istanza database sia disponibile possono trascorrere fino a 20 minuti.
Unable to connect to your DB instance (Impossib ile connettersi all'istanza database)	Se non riesci a inviare o ricevere comunicazioni sulla porta specifica ta quando hai creato l'istanza database, non potrai connetterti all'istanza database. Verifica con il tuo amministratore di rete che la porta che hai specificato per la tua istanza database consenta la comunicazione in entrata e in uscita.

Problema	Suggerimenti sulla risoluzione dei problemi
Unable to connect to your DB instance (Impossib ile connettersi all'istanza database)	Le regole di accesso applicate dal firewall locale e gli indirizzi IP a cui hai fornito l'autorizzazione per accedere alla tua istanza database nel gruppo di sicurezza per l'istanza database potrebber o non corrispondere. Il problema è molto probabilmente legato alle regole in entrata e in uscita sul firewall. Puoi aggiungere o modificare una regola in entrata nel gruppo di sicurezza: per Source (Origine), scegli My IP (Il mio IP). Questo consente l'accesso all'istanza database dall'indirizzo IP rilevato nel browser. Per ulteriori informazioni, consulta VPC di Amazon VPC e Amazon RDS. Per ulteriori informazioni sui gruppi di sicurezza, consulta Controllo dell'accesso con i gruppi di sicurezza. Per informazioni sul processo di impostazione delle regole del gruppo di sicurezza, consulta Tutorial: Creazione di un Amazon VPC da utilizzare con un'istanza database (solo IPv4).
Connect failed because target host or object does not exist – Oracle, Error: ORA-12545 (Connessi one non riuscita perché l'oggetto o l'host di destinazione non esiste – Oracle, errore: ORA-12545)	Verificare di avere specificato correttamente il nome del server e il numero di porta. Per Server name (Nome server) inserire o incollare il nome DNS dalla console. Per informazioni su come trovare il nome DNS e il numero di porta di un'istanza database, consulta Esito dell'endpoint dell'istanza database RDS per Oracle.
Invalid username/ password; logon denied – Oracle, Error: ORA-01017 (Nome utente/password non valido; accesso negato – Oracle, errore: ORA-01017)	È stato possibile raggiungere l'istanza database, ma la connessio ne è stata rifiutata. Ciò avviene in genere quando si fornisce una password o un nome utente non corretto. Verificare il nome utente e la password e riprovare.

Problema	Suggerimenti sulla risoluzione dei problemi
TNS:listener attualmente non conosce il SID fornito nel descrittore di connessione - Oracle, ERRORE: ORA-12505	Assicurati che sia inserito il SID corretto. Il SID è lo stesso del nome DB. Trova il nome del database nella scheda Configuration (Configurazione) della pagina Databases (Database) dell'istanza. Puoi individuare il nome del database con AWS CLI: `aws rds describe-db-instances --query 'DBInstances[*].[DBInstanceIdentifier,DBName]' --output text`

Per ulteriori informazioni sui problemi di connessione, consulta Impossibile connettersi all'istanza database di Amazon RDS.

Modifica delle proprietà di connessione tramite i parametri sqlnet.ora

Il file sqlnet.ora include parametri che configurano le caratteristiche della rete di Oracle su server e client database di Oracle. Utilizzando i parametri nel file sqlnet.ora, è possibile modificare le proprietà per le connessioni in entrata e in uscita del database.

Per ulteriori informazioni sul perché impostare i parametri sqlnet.ora, consulta Configuring Profile Parameters (Configurazione dei parametri del profilo) nella documentazione di Oracle.

Impostazione dei parametri sqlnet.ora

I gruppi di parametri di Amazon RDS for Oracle includono una serie di parametri sqlnet.ora. Si impostano nello stesso modo in cui si impostano gli altri parametri Oracle. Il prefisso `sqlnetora.` identifica quali parametri sono i parametri sqlnet.ora. Ad esempio, in un gruppo di parametri Oracle in Amazon RDS il parametro `default_sdu_size` sqlnet.ora è `sqlnetora.default_sdu_size`.

Per informazioni sulla gestione dei gruppi di parametri e sull'impostazione dei valori del parametro, consulta Utilizzo di gruppi di parametri.

Parametri sqlnet.ora supportati

Amazon RDS supporta i seguenti parametri sqlnet.ora. Le modifiche apportate ai parametri sqlnet.ora hanno effetto immediato.

Parametro	Valori validi	Statico/Dinamico	Descrizione
sqlnetora.default_sdu_size	Oracle 12c – Da 512 a 20971	Dinamico	La dimensione, in byte, della session data unit (SDU)/Unità dei dati della sessione. L'SDU è la quantità di dati che viene inserita in un buffer e inviata attraverso la rete in una volta.
sqlnetora.diag_adr_enabled	ON, OFF	Dinamico	Un valore che abilita o disabilita la traccia di Automatic Diagnostic Repository (ADR) ON specifica che viene utilizzata la traccia del file ADR. OFF specifica che non viene utilizzata la traccia del file ADR.
sqlnetora.recv_buf_size	8192 Da a 26843	Dinamico	Il limite di spazio del buffer per le operazioni di ricezione delle sessioni, supportato dai protocolli TCP/IP, TCP/IP con SSL e SDP.
sqlnetora.send_buf_size	8192 Da a 26843	Dinamico	Il limite di spazio del buffer per le operazioni di invio delle sessioni, supportato dai protocolli TCP/IP, TCP/IP con SSL e SDP.

Parametro	Valori validi	Statico/Dinamico	Descrizione
sqlnetora.sqlnet.allowed_logon_version_client	8, 10, 11, 12	Dinamico	Versione del protocollo di autenticazione minima consentita per client e server operanti come client per stabilire una connessione alle istanze database Oracle.
sqlnetora.sqlnet.allowed_logon_version_server	8, 9, 10, 11, 12, 12a	Dinamico	Versione del protocollo di autenticazione minima consentita per stabilire una connessione alle istanze database Oracle.
sqlnetora.sqlnet.expire_time	0 Da a 1440	Dinamico	Intervallo di tempo, in minuti, per inviare un controllo per verificare che le connessioni client-server siano attive.
sqlnetora.sqlnet.inbound_connect_timeout	0 oppure 10 a 7200	Dinamico	Tempo, in secondi, affinché un client si connetta al server del database e fornisca le informazioni di autenticazione necessarie.
sqlnetora.sqlnet.outbound_connect_timeout	0 oppure 10 a 7200	Dinamico	Tempo, in secondi, affinché un client stabilisca una connessione Oracle Net con l'istanza database.
sqlnetora.sqlnet.recv_timeout	0 oppure 10 a 7200	Dinamico	Tempo, in secondi, affinché un server di database attenda i dati del client dopo aver stabilito una connessione.

Parametro	Valori validi	Statico/Dinamico	Descrizione
`sqlnetora.sqlnet.send_timeout`	`0` oppure `10` a `7200`	Dinamico	Tempo, in secondi, affinché un server di database completi un'operazione di invio ai client dopo aver stabilito una connessione.
`sqlnetora.tcp.connect_timeout`	`0` oppure `10` a `7200`	Dinamico	Tempo, in secondi, affinché un client stabilisca una connessione TCP con il server database.
`sqlnetora.trace_level_server`	`0`, `4`, `10`, `16`, `OFF`, `USER`, `ADMIN`, `SUPPOF`	Dinamico	Per la traccia non ADR, accendere la traccia del server a un livello specificato o spegnerla.

Il valore predefinito per ciascun parametro sqlnet.ora supportato è l'impostazione predefinita di Oracle per la versione. Per ulteriori informazioni sui valori predefiniti per Oracle Database 12c, consulta Parametri per il file sqlnet.ora nella documentazione di Oracle Database 12c.

Visualizzazione dei parametri sqlnet.ora

Puoi visualizzare i parametri sqlnet.ora e le rispettive impostazioni usando la AWS Management Console, AWS CLI o un client SQL.

Visualizzazione dei parametri sqlnet.ora usando la console

Per informazioni sulla visualizzazione dei parametri in un gruppo di parametri, consulta Utilizzo di gruppi di parametri.

Nel gruppo di parametri di Oracle, il prefisso `sqlnetora.` identifica quali parametri sono i parametri sqlnet.ora.

Visualizzazione dei parametri sqlnet.ora usando AWS CLI

Per visualizzare i parametri sqlnet.ora configurati in un gruppo di parametri Oracle, utilizzare il comando. AWS CLI describe-db-parameters

Per visualizzare tutti i parametri sqlnet.ora per un'istanza di Oracle DB, chiamate il comando -portion. AWS CLI download-db-log-file Specifica l'identificatore istanze DB, il nome del file di registro e il tipo di output.

Example

Il codice seguente elenca tutte i parametri sqlnet.ora per `mydbinstance`.

Per, o: Linux macOS Unix

```
aws rds download-db-log-file-portion \
    --db-instance-identifier mydbinstance \
    --log-file-name trace/sqlnet-parameters \
    --output text
```

Per Windows:

```
aws rds download-db-log-file-portion ^
    --db-instance-identifier mydbinstance ^
    --log-file-name trace/sqlnet-parameters ^
    --output text
```

Visualizzazione dei parametri sqlnet.ora usando un client SQL

Dopo essersi connessi all'istanza database di Oracle in un client SQL, la seguente query elenca i parametri sqlnet.ora.

```
SELECT * FROM TABLE
   (rdsadmin.rds_file_util.read_text_file(
        p_directory => 'BDUMP',
        p_filename  => 'sqlnet-parameters'));
```

Per informazioni sulla connessione a un'istanza database Oracle in un client SQL, consulta Connessione all'istanza database RDS per Oracle.

Protezione delle connessioni di istanze database di Oracle

Amazon RDS for Oracle supporta le connessioni crittografate SSL/TLS così come l'opzione Native Network Encryption (NNE) di Oracle per crittografare le connessioni tra l'applicazione e l'istanza database Oracle. Per ulteriori informazioni sull'opzione Native Network Encryption di Oracle, consulta Oracle native network encryption.

Argomenti

Utilizzo di SSL con un'istanza database RDS per Oracle

Secure Sockets Layer (SSL) è un protocollo standard del settore utilizzato per proteggere connessioni di rete tra client e server. Dopo SSL versione 3.0, il nome è stato modificato in Transport Layer Security (TLS) ma spesso viene ancora indicato come protocollo SSL. Amazon RDS supporta la crittografia SSL per le istanze database Oracle. Mediante SSL, è possibile crittografare una connessione tra l'applicazione cliente e l'istanza database di Oracle. Il supporto per SSL è disponibile in tutte le regioni AWS per Oracle.

Puoi abilitare la crittografia SSL per un'istanza database Oracle aggiungendo l'opzione Oracle SSL al gruppo di opzioni associato all'istanza database. Amazon RDS utilizza una seconda porta, come richiesto da Oracle, per le connessioni SSL. In questo modo è consentito allo stesso tempo sia testo in chiaro sia comunicazioni con crittografia SSL tra un'istanza database e un cliente Oracle. Ad esempio, è possibile utilizzare la porta con testo in chiaro per comunicare con altre risorse all'interno di un VPC mentre utilizzi la porta con crittografia SSL per comunicare con risorse all'esterno del VPC.

Per ulteriori informazioni, consulta Oracle Secure Sockets Layer.

> **Note**
>
> È possibile utilizzare sia SSL sia NNE di Oracle sulla stessa istanza database. Prima di iniziare a utilizzare la crittografia SSL, è necessario disabilitare le altre crittografie di connessione.

Aggiornamento delle applicazioni per la connessione a istanze database Oracle mediante nuovi certificati SSL/TLS

A partire dal 13 gennaio 2023, Amazon RDS ha pubblicato nuovi certificati dell'autorità di certificazione (CA) per la connessione alle istanze database RDS utilizzando Secure Socket Layer o Transport Layer Security (SSL/TLS). Di seguito sono disponibili le informazioni sull'aggiornamento delle applicazioni per utilizzare i nuovi certificati.

Questo argomento aiuta a determinare se le applicazioni client utilizzano SSL/TLS per connettersi alle istanze database.

> **Important**
>
> Quando si modifica il certificato per un'istanza database Amazon RDS for Oracle, viene riavviato solo il listener di database. L'istanza database non viene riavviata. Le connessioni al database esistenti non vengono influenzate, ma le nuove connessioni presenteranno errori per un breve periodo durante il riavvio del listener.

> **Note**
>
> Per le applicazioni client che utilizzano SSL/TLS per la connessione alle istanze database, è necessario aggiornare gli archivi di trust delle applicazioni client per includere i nuovi certificati CA.

Dopo aver aggiornato i certificati CA negli archivi di trust delle applicazioni client, puoi ruotare i certificati nelle istanze database. Consigliamo vivamente di testare queste procedure in un ambiente di sviluppo o di gestione temporanea prima di implementarle negli ambienti di produzione.

Per ulteriori informazioni sulla rotazione dei certificati, consulta Rotazione del certificato SSL/TLS. Per ulteriori informazioni sul download, consulta . Per informazioni sull'utilizzo di SSL/TLS con le istanze database Oracle, consulta Oracle Secure Sockets Layer.

Argomenti

- Verifica se le applicazioni si connettono utilizzando SSL
- Aggiornare l'archivio di trust delle applicazioni
- Codice Java di esempio per stabilire connessioni SSL

Verifica se le applicazioni si connettono utilizzando SSL

Se la tua istanza database Oracle utilizza un gruppo di opzioni con l'opzione SSL aggiunta, potresti dover utilizzare SSL. Controlla seguendo le istruzioni seguenti in Generazione di un elenco delle opzioni e delle impostazioni delle opzioni per un gruppo di opzioni. Per ulteriori informazioni sull'opzione SSL, consulta Oracle Secure Sockets Layer.

Controlla il log listener per determinare se ci siano connessioni SSL. Di seguito è riportato un output di esempio in un log listener.

```
date time * (CONNECT_DATA=(CID=(PROGRAM=program)
(HOST=host)(USER=user))(SID=sid)) *
(ADDRESS=(PROTOCOL=tcps)(HOST=host)(PORT=port)) * establish * ORCL * 0
```

Quando PROTOCOL ha il valore `tcps` per una entry, mostra una connessione SSL. Tuttavia, quando HOST è `127.0.0.1`, puoi ignorare la entry. Le connessioni da `127.0.0.1` sono un management agent locale sull'istanza database. Tali connessioni non sono esterne a SSL. Pertanto, hai applicazioni di connessione mediante SSL se vedi voci dei listener log in cui PROTOCOL è `tcps` e HOST non è `127.0.0.1`.

Per verificare il listener log, puoi pubblicare il log su Amazon CloudWatch Logs. Per ulteriori informazioni, consulta Pubblicazione dei log Oracle su Amazon CloudWatch Logs.

Aggiornare l'archivio di trust delle applicazioni

Puoi aggiornare l'archivio di trust delle applicazioni che utilizzano SQL*Plus o JDBC per le connessioni SSL/TLS.

Aggiornare l'archivio di trust delle applicazioni per SQL*Plus

Puoi aggiornare l'archivio di trust delle applicazioni che utilizzano SQL*Plus per le connessioni SSL/TLS.

> **Note**
>
> Quando aggiorni l'archivio di trust puoi conservare i certificati meno recenti oltre ad aggiungere i nuovi certificati.

Per aggiornare l'archivio di trust delle applicazioni SQL*Plus

1. Scarica il nuovo certificato root idoneo per tutte le regioni AWS e posiziona il file nella directory `ssl_wallet`.

   Per ulteriori informazioni sul download del certificato root, consulta .
2. Esegui i comandi seguenti per aggiornare il wallet Oracle.

   ```
   prompt>orapki wallet add -wallet $ORACLE_HOME/ssl_wallet -trusted_cert -cert
         $ORACLE_HOME/ssl_wallet/ssl-cert.pem -auto_login_only
   ```

   Sostituire il nome del file con il nome del file scaricato.
3. Eseguire il comando seguente per confermare che il wallet è stato correttamente installato.

   ```
   prompt>orapki wallet display -wallet $ORACLE_HOME/ssl_wallet
   ```

   L'output dovrebbe contenere le seguenti informazioni.

   ```
   Trusted Certificates:
   Subject: CN=Amazon RDS Root 2019 CA,OU=Amazon RDS,O=Amazon Web Services\,
    Inc.,L=Seattle,ST=Washington,C=US
   ```

Aggiornare l'archivio di trust delle applicazioni per JDBC

Puoi aggiornare l'archivio di trust delle applicazioni che utilizzano JDBC per le connessioni SSL/TLS.

Per ulteriori informazioni sul download del certificato root, consulta .

Per gli script di esempio che importano i certificati, consulta Script di esempio per l'importazione di certificati nel tuo archivio di trust.

Codice Java di esempio per stabilire connessioni SSL

L'esempio di codice seguente mostra come impostare la connessione SSL utilizzando JDBC.

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.util.Properties;

public class OracleSslConnectionTest {
    private static final String DB_SERVER_NAME = "<dns-name-provided-by-amazon-rds>";
    private static final Integer SSL_PORT = "<ssl-option-port-configured-in-option-
group>";
    private static final String DB_SID = "<oracle-sid>";
    private static final String DB_USER = "<user name>";
    private static final String DB_PASSWORD = "<password>";
    // This key store has only the prod root ca.
    private static final String KEY_STORE_FILE_PATH = "<file-path-to-keystore>";
    private static final String KEY_STORE_PASS = "<keystore-password>";

    public static void main(String[] args) throws SQLException {
        final Properties properties = new Properties();
        final String connectionString = String.format(
                "jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCPS)(HOST=%s)(PORT=
%d))(CONNECT_DATA=(SID=%s)))",
                DB_SERVER_NAME, SSL_PORT, DB_SID);
        properties.put("user", DB_USER);
        properties.put("password", DB_PASSWORD);
        properties.put("oracle.jdbc.J2EE13Compliant", "true");
        properties.put("javax.net.ssl.trustStore", KEY_STORE_FILE_PATH);
        properties.put("javax.net.ssl.trustStoreType", "JKS");
        properties.put("javax.net.ssl.trustStorePassword", KEY_STORE_PASS);
        final Connection connection = DriverManager.getConnection(connectionString,
 properties);
       // If no exception, that means handshake has passed, and an SSL connection can
 be opened
    }
}
```

> **Important**
>
> Dopo aver determinato che le connessioni al database utilizzano SSL/TLS e aver aggiornato l'archivio di trust delle applicazioni, puoi aggiornare il database per utilizzare i certificati rds-ca-rsa2048-g1. Per istruzioni, consulta la fase 3 in Aggiornamento del certificato CA modificando l'istanza o il cluster di database.

Utilizzo di native network encryption con un'istanza database RDS per Oracle

Oracle Database offre due modi per crittografare i dati sulla rete: native network encryption (NNE) e Transport Layer Security (TLS). NNE è una funzionalità di sicurezza di proprietà di Oracle, mentre TLS è uno standard di settore. RDS per Oracle supporta NNE per tutte le edizioni di Oracle Database.

NNE ha i vantaggi seguenti rispetto a TLS:

- È possibile controllare NNE sul client e sul server utilizzando le impostazioni dell'opzione NNE:
  - `SQLNET.ALLOW_WEAK_CRYPTO_CLIENTS` e `SQLNET.ALLOW_WEAK_CRYPTO`
  - `SQLNET.CRYPTO_CHECKSUM_CLIENT` e `SQLNET.CRYPTO_CHECKSUM_SERVER`
  - `SQLNET.CRYPTO_CHECKSUM_TYPES_CLIENT` e `SQLNET.CRYPTO_CHECKSUM_TYPES_SERVER`
  - `SQLNET.ENCRYPTION_CLIENT` e `SQLNET.ENCRYPTION_SERVER`
  - `SQLNET.ENCRYPTION_TYPES_CLIENT` e `SQLNET.ENCRYPTION_TYPES_SERVER`
- Nella maggior parte dei casi, non devi configurare il client o il server. Al contrario, TLS richiede la configurazione sia del client che del server.
- Non sono richiesti certificati. In TLS, il server richiede un certificato (che alla fine scade) e il client richiede un certificato root attendibile per l'autorità di certificazione che ha emesso il certificato del server.

Puoi abilitare la crittografia NNE per un'istanza database Oracle aggiungendo l'opzione Oracle NNE al gruppo di opzioni associato all'istanza database. Per ulteriori informazioni, consulta Oracle native network encryption.

> **Note**
>
> È possibile utilizzare NNE e TLS sulla stessa istanza database.

Configurazione dell'autenticazione Kerberos per Amazon RDS for Oracle

Puoi utilizzare l'autenticazione Kerberos per autenticare gli utenti quando si connettono all'istanza database di Amazon RDS per Oracle. In questa configurazione, l'istanza database funziona con AWS Directory Service for Microsoft Active Directory, chiamata anche AWS Managed Microsoft AD. Quando gli utenti eseguono l'autenticazione con un'istanza database di RDS per Oracle unita al dominio trusting, le richieste di autenticazione vengono inoltrate alla directory creata con AWS Directory Service.

Mantenere tutte le credenziali nella stessa directory consente di ridurre il tempo e l'impegno. È disponibile una posizione centralizzata per archiviare e gestire le credenziali per più istanze database. L'uso di una directory può inoltre migliorare il profilo di sicurezza complessivo.

Disponibilità di regioni e versioni

Il supporto varia a seconda delle versioni specifiche di ciascun motore di database e a seconda delle Regioni AWS. Per ulteriori informazioni sulla disponibilità delle versioni e regioni di RDS per Oracle con autenticazione Kerberos, consulta Regioni e motori DB supportati per l'autenticazione Kerberos in Amazon RDS.

> **Note**
>
> L'autenticazione Kerberos non è supportata per classi di istanze database dichiarate obsolete per istanze database di RDS per Oracle. Per ulteriori informazioni, consulta Classi di istanza RDS for Oracle.

Argomenti

- Configurazione dell'autenticazione Kerberos per istanze database Oracle
- Gestione di un'istanza database in un dominio
- Connessione a Oracle con Autenticazione Kerberos

Configurazione dell'autenticazione Kerberos per istanze database Oracle

Utilizzare AWS Directory Service for Microsoft Active Directory, chiamato anche AWS Managed Microsoft AD, per configurare l'autenticazione Kerberos per un'istanza DB Oracle. Per configurare Autenticazione Kerberos, completa le fasi seguenti:

- Fase 1: Creare una directory utilizzando AWS Managed Microsoft AD
- Fase 2: creazione di un trust
- Fase 3: configurazione delle autorizzazioni IAM per Amazon RDS
- Fase 4: creazione e configurazione di utenti
- Fase 5: abilitazione del traffico tra VPC tra la directory e l'istanza database
- Fase 6: creazione o modifica di un'istanza database Oracle
- Fase 7: creazione di login Oracle di autenticazione Kerberos
- Fase 8: configurazione di un client Oracle

> **Note**
>
> Durante l'installazione, RDS crea un utente del database Oracle denominato *`managed_service_user`*@*`example.com`* con il privilegio `CREATE SESSION`, dove *`example.com`* è il nome di dominio. Questo utente corrisponde all'utente creato dal servizio directory all'interno di Active Directory gestito. Periodicamente, RDS utilizza le credenziali fornite dal servizio directory per accedere al database Oracle. Successivamente, RDS distrugge immediatamente la cache dei ticket.

Fase 1: Creare una directory utilizzando AWS Managed Microsoft AD

AWS Directory Service crea una Active Directory completamente gestita nel AWS cloud. Quando crei una AWS Managed Microsoft AD directory, AWS Directory Service crea due controller di dominio e server DNS (Domain Name System) per tuo conto. I server di directory vengono creati in sottoreti diverse in un VPC. Questa ridondanza assicura che la directory rimanga accessibile anche se si verifica un errore.

Quando crei una AWS Managed Microsoft AD directory, AWS Directory Service esegue le seguenti attività per tuo conto:

- Configura una Active Directory all'interno del VPC.

- Crea un account amministratore della directory con nome utente Admin e la password specificata. Puoi utilizzare questo account per gestire le directory.

> **Note**
>
> Assicurati di salvare questa password. AWS Directory Service non la memorizza. È possibile reimpostarla ma non recuperarla.

- Crea un gruppo di sicurezza per i controller della directory.

Quando si avvia un AWS Managed Microsoft AD, AWS crea un'unità organizzativa (OU) che contiene tutti gli oggetti della directory. Questa unità organizzativa ha lo stesso nome NetBIOS che hai digitato al momento della creazione della directory e si trova nella radice del dominio. La radice del dominio è di proprietà e gestita da AWS.

L'account amministratore creato con la AWS Managed Microsoft AD directory dispone delle autorizzazioni per le attività amministrative più comuni dell'unità organizzativa:

- Creazione, aggiornamento o eliminazione di utenti
- Aggiungi risorse al dominio, come file server o server di stampa, e assegna le autorizzazioni per tali risorse a utenti dell'unità organizzativa
- Creazione di unità organizzative e container aggiuntivi
- Delega dell'autorità
- Ripristino degli oggetti eliminati dal cestino di Active Directory
- Esegui i PowerShell moduli Windows AD e DNS sul servizio Web Active Directory

L'account Admin dispone inoltre dei diritti per eseguire le seguenti attività estese a tutto il dominio:

- gestione delle configurazioni DNS (aggiunta, eliminazione o aggiornamento di record, zone e server d'inoltro);
- visualizzazione di log di eventi DNS;
- visualizzazione di log di eventi di sicurezza.

Per creare la directory, usa l' AWS Management Console AWS CLI, o l' AWS Directory Service API. Assicurati di aprire le porte in uscita pertinenti nel gruppo di sicurezza delle directory in modo che la directory possa comunicare con l'istanza database Oracle.

Per creare una directory con AWS Managed Microsoft AD

1. Accedere AWS Management Console e aprire la AWS Directory Service console all'indirizzo https://console.aws.amazon.com/directoryservicev2/.
2. Nel riquadro di navigazione, seleziona Directories (Directory) e quindi Set up directory (Configura la directory).
3. Scegliete AWS Managed Microsoft AD. AWS Managed Microsoft AD è l'unica opzione attualmente utilizzabile con Amazon RDS.
4. Immettere le seguenti informazioni:

   Nome DNS directory

   Il nome completo della directory, ad esempio **`corp.example.com`**.

   Nome NetBIOS della directory

   Nome breve per la directory, ad esempio **`CORP`**.

   Descrizione della directory

   (Opzionale) Una descrizione della directory.

   Password amministratore

   La password dell'amministratore della directory. Con il processo di creazione della directory viene generato un account amministratore con il nome utente Admin e questa password.

   La password dell'amministratore della directory e non può includere il termine "admin". La password distingue tra maiuscole e minuscole e la lunghezza deve essere compresa tra 8 e 64 caratteri. Deve anche contenere un carattere di almeno tre delle seguenti quattro categorie:

   - Lettere minuscole (a–z)
   - Lettere maiuscole (A–Z)
   - Numeri (0–9)
   - Caratteri non alfanumerici (~!@#$%^&*_-+=`|\(){}[]:;"'<>,.?/)

   Confirm password (Conferma password)

   La password dell'amministratore digitata nuovamente.

5. Seleziona Successivo.

6. Immettere le seguenti informazioni nella sezione Networking (Rete) e quindi scegliere Next (Avanti):

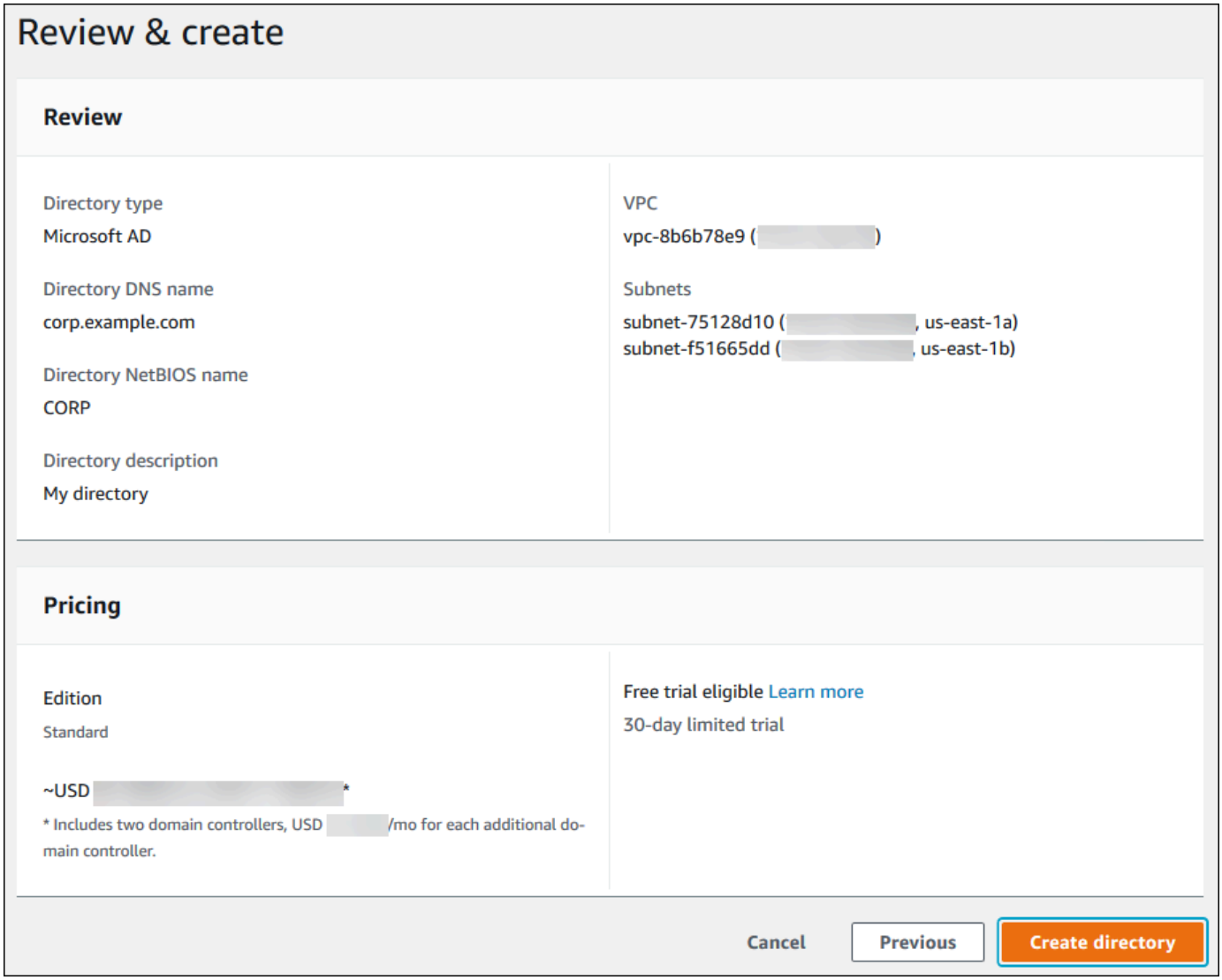
   VPC

   VPC per la directory. Creare l'istanza database Oracle in questo stesso VPC.

   Sottoreti

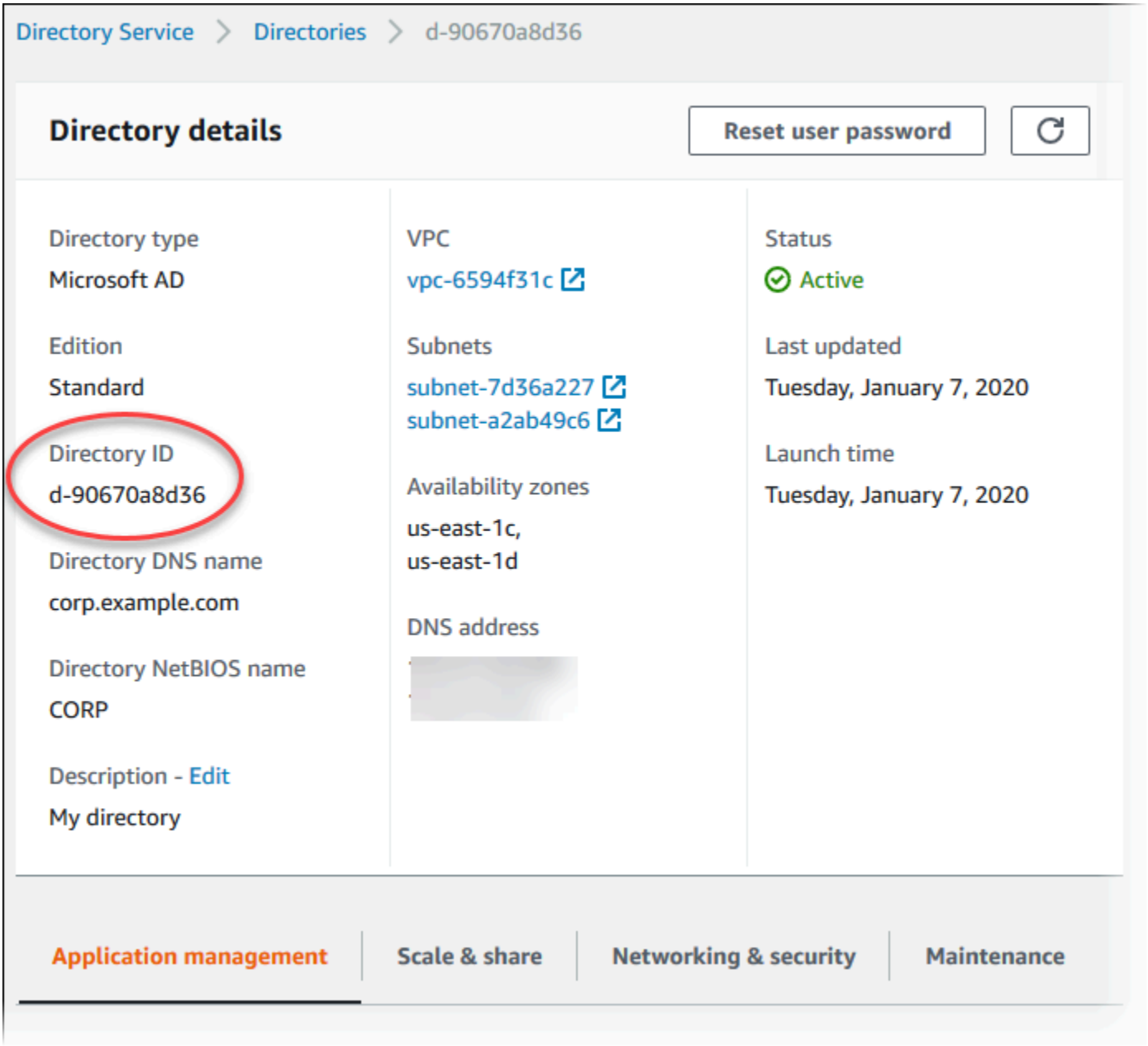
   Sottoreti per i server di directory. Le due sottoreti devono trovarsi in diverse zone di disponibilità.

7. Esaminare le informazioni relative alla directory e apportare eventuali modifiche. Quando le informazioni sono corrette, scegli Create Directory (Crea directory).



Per creare la directory sono necessari alcuni minuti. Una volta creata correttamente la directory, il valore Status (Stato) viene modificato in Active (Attivo).

Per consultare le informazioni sulla directory, selezionare il nome della directory nell'elenco di directory. Prendere nota del valore di Directory ID (ID directory) perché sarà necessario quando si crea o si modifica l'istanza database Oracle.



Fase 2: creazione di un trust

Se prevedi di utilizzarla AWS Managed Microsoft AD solo, passa a[Fase 3: configurazione delle autorizzazioni IAM per Amazon RDS](#).

Per ottenere Autenticazione Kerberos utilizzando un account Microsoft Active Directory locale o autogestito, crea un trust tra foreste o un trust esterno. La fiducia può essere a senso unico o bidirezionale. Per ulteriori informazioni sulla configurazione dei trust forestali utilizzando AWS Directory Service, vedere Quando creare una relazione di fiducia nella Guida all'AWS Directory Service amministrazione.

Fase 3: configurazione delle autorizzazioni IAM per Amazon RDS

AWS Directory Service Per chiamarti, Amazon RDS richiede un ruolo IAM che utilizzi la policy `AmazonRDSDirectoryServiceAccess` IAM gestita. Questo ruolo permette ad Amazon RDS di effettuare chiamate alla AWS Directory Service.

> **Note**
>
> Affinché il ruolo consenta l'accesso, l'endpoint AWS Security Token Service (AWS STS) deve essere attivato nel modo corretto Regione AWS per te. Account AWS AWS STS Gli endpoint sono attivi per impostazione predefinita in tutti Regioni AWS gli ambienti e puoi utilizzarli senza ulteriori azioni. Per ulteriori informazioni, consulta Attivazione e disattivazione AWS STSRegione AWS in un capitolo della IAM User Guide.

Creazione di un ruolo IAM

Quando crei un'istanza DB utilizzando e l' AWS Management Console utente della console dispone dell'`iam:CreateRole`autorizzazione, la console viene creata automaticamente. `rds-directoryservice-kerberos-access-role` In caso contrario, è necessario creare manualmente il ruolo IAM. Quando crei manualmente un ruolo IAM`Directory Service`, scegli e allega la policy AWS gestita `AmazonRDSDirectoryServiceAccess` ad esso.

Per ulteriori informazioni sulla creazione di ruoli IAM per un servizio, consulta Creating a role to delegate permissions to an AWS service nella IAM User Guide.

> **Note**
>
> Il ruolo IAM utilizzato per l'autenticazione Windows per RDS per Microsoft SQL Server non può essere utilizzato per RDS per Oracle.

Creazione manuale di una policy di attendibilità IAM

Facoltativamente, puoi creare policy delle risorse con le autorizzazioni richieste anziché utilizzare la policy IAM gestita `AmazonRDSDirectoryServiceAccess`. Specifica sia `directoryservice.rds.amazonaws.com` che `rds.amazonaws.com` come principali.

Per limitare le autorizzazioni alle risorse che Amazon RDS fornisce a un altro servizio per una risorsa specifica, si consiglia di utilizzare le chiavi di contesto delle condizioni globali `aws:SourceArn` e `aws:SourceAccount` nelle policy delle risorse. Il modo più efficace per proteggersi dal problema "confused deputy" è usare la chiave di contesto della condizione globale `aws:SourceArn` con l'ARN completo di una risorsa di Amazon RDS. Per ulteriori informazioni, consulta Prevenzione del problema "confused deputy" tra servizi.

L'esempio seguente mostra il modo in cui puoi utilizzare le chiavi di contesto `aws:SourceArn` e `aws:SourceAccount` delle condizioni globali in Amazon RDS per prevenire il problema “confused deputy”.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "directoryservice.rds.amazonaws.com",
          "rds.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "ArnLike": {
          "aws:SourceArn": "arn:aws:rds:us-east-1:123456789012:db:mydbinstance"
        },
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        }
      }
    }
  ]
}
```

Per le regioni che hanno aderito all'iniziativa, devi anche includere un responsabile del servizio per quella regione sotto forma di. `directoryservice.rds.region_name.amazonaws.com` Ad esempio, nella regione africana (Città del Capo), utilizza la seguente politica di fiducia:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "directoryservice.rds.amazonaws.com",
          "directoryservice.rds.af-south-1.amazonaws.com",
          "rds.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "ArnLike": {
          "aws:SourceArn": "arn:aws:rds:af-south-1:123456789012:db:mydbinstance"
        },
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        }
      }
    }
  ]
}
```

Il ruolo deve anche disporre della seguente policy IAM.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "ds:DescribeDirectories",
        "ds:AuthorizeApplication",
        "ds:UnauthorizeApplication",
        "ds:GetAuthorizedApplicationDetails"
      ],
    "Effect": "Allow",
```

```
    "Resource": "*"
    }
  ]
}
```

Fase 4: creazione e configurazione di utenti

Puoi creare utenti utilizzando lo strumento Users and Computers (Utenti e computer) di Active Directory, che è uno degli strumenti Domain Services (Servizi di dominio) e Lightweight Directory Services (Servizi di Lightweight Directory) di Active Directory. In questo caso, gli utenti sono individui singoli o entità che hanno accesso alla tua directory.

Per creare utenti in una AWS Directory Service directory, devi essere connesso a un'istanza Amazon EC2 basata su Windows che fa parte della directory. AWS Directory Service Allo stesso tempo, devi essere connesso come un utente che dispone di privilegi per creare utenti. Per ulteriori informazioni sulla creazione di utenti Microsoft Active Directory, consulta Gestione di utenti e gruppi in AWS Managed Microsoft AD nella Guida all'amministrazione di AWS Directory Service .

Fase 5: abilitazione del traffico tra VPC tra la directory e l'istanza database

Se prevedi di individuare la directory e l'istanza database nello stesso VPC, ignora questa fase e passa a Fase 6: creazione o modifica di un'istanza database Oracle.

Se prevedi di localizzare la directory e l'istanza DB in AWS account o VPC diversi, configura il traffico cross-VPC utilizzando il peering VPC o il Transit Gateway.AWS La procedura seguente abilita il traffico tra VPC utilizzando il peering di VPC. Segui le istruzioni in Che cos'è il peering di VPC? nella Amazon Virtual Private Cloud Peering Guide.

Per abilitare il traffico tra VPC utilizzando il peering di VPC

1. Configurare le regole di routing VPC appropriate per garantire che il traffico di rete possa scorrere in entrambe le direzioni.
2. Assicurarsi che il gruppo di protezione dell'istanza database possa ricevere traffico in entrata dal gruppo di sicurezza della directory. Per ulteriori informazioni, consulta Best practice per AWS Managed Microsoft AD nella Guida all'amministrazione di AWS Directory Service .
3. Assicurati che non sia presente una regola della lista di controllo accessi (ACL) di rete per bloccare il traffico.

Se la directory è di proprietà di un altro AWS account, è necessario condividerla.

Per condividere la cartella tra AWS account

1. Inizia a condividere la directory con l' AWS account in cui verrà creata l'istanza DB seguendo le istruzioni riportate nel Tutorial: Sharing your AWS Managed Microsoft AD directory for seamless EC2 Domain-join nella Administration Guide.AWS Directory Service
2. Accedi alla AWS Directory Service console utilizzando l'account per l'istanza DB e assicurati che il dominio abbia lo stato prima di procedere. `SHARED`
3. Dopo aver effettuato l'accesso alla AWS Directory Service console utilizzando l'account per l'istanza DB, annota il valore Directory ID. Utilizzare questo ID directory per aggiungere l'istanza database al dominio.

Fase 6: creazione o modifica di un'istanza database Oracle

Crea o modifica un'istanza database Oracle per l'utilizzo con la directory. Puoi utilizzare la console, CLI o l'API RDS per associare un'istanza database a una directory. Questa operazione può essere eseguita in uno dei seguenti modi:

- Crea una nuova istanza Oracle DB utilizzando la console, il comando create-db-instanceCLI o l'operazione API CreateDBInstance RDS.

  Per istruzioni, consulta Creazione di un'istanza database Amazon RDS.
- Modifica un'istanza Oracle DB esistente utilizzando la console, il comando modify-db-instanceCLI o l'operazione ModifyDBInstance RDS API.

  Per istruzioni, consulta Modifica di un'istanza database Amazon RDS.
- Ripristina un'istanza Oracle DB da uno snapshot DB utilizzando la console, il comando CLI restore-db-instance-from-db-snapshot o l'operazione API RestoreDB DBSnapshot RDS. InstanceFrom

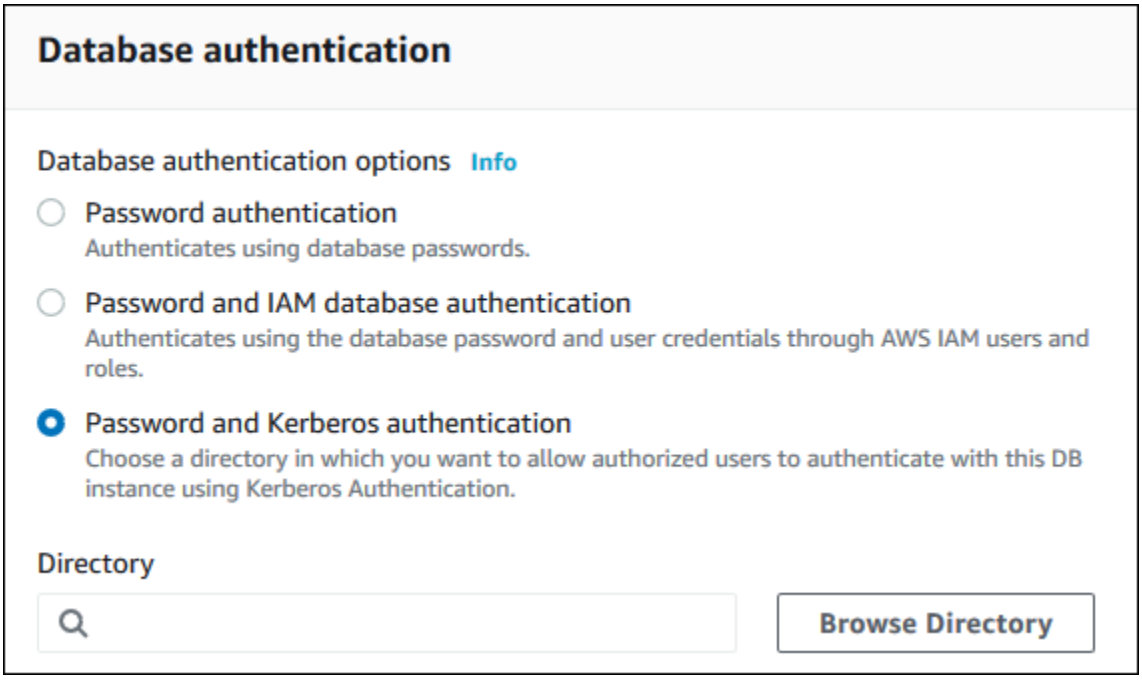
  Per istruzioni, consulta Ripristino da uno snapshot database.
- Ripristina un'istanza Oracle DB point-in-time utilizzando la console, il comando restore-db-instance-to- point-in-time CLI o l'operazione RestoreDB RDS API InstanceToPointInTime.

  Per istruzioni, consulta Ripristino a un'ora specifica per un'istanza database.

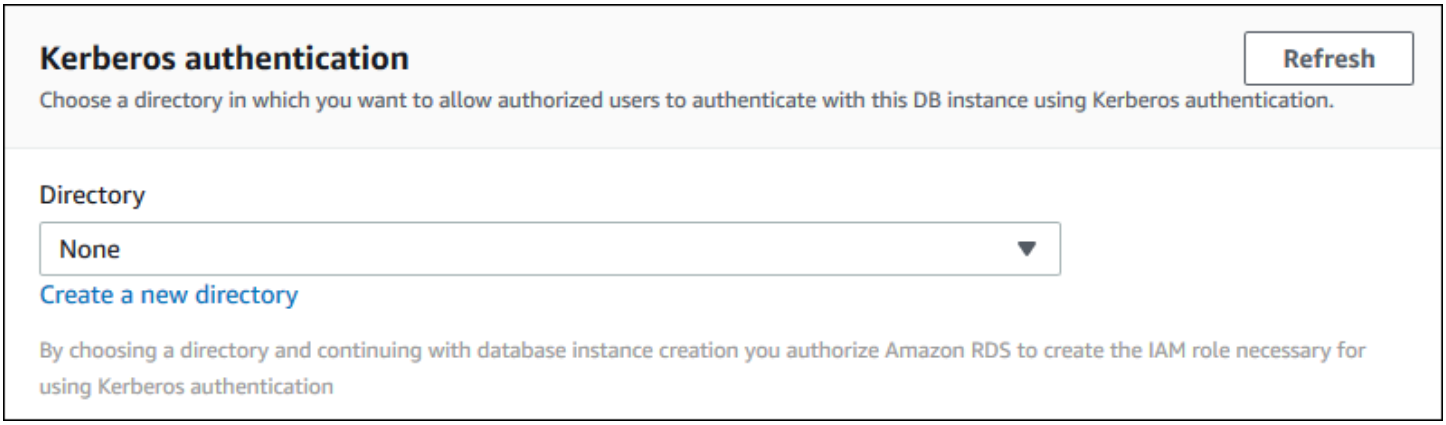
L'autenticazione Kerberos è supportata solo per istanze database Oracle in un VPC. L'istanza database Oracle può trovarsi nello stesso VPC della directory o in un VPC diverso. Quando crei o modifichi l'istanza database, completa le seguenti operazioni:

- Specifica l'identificativo del dominio (identificativo `d-*`) generato al momento della creazione della directory.
- Specifica anche il nome del ruolo IAM creato.
- Assicurati che il gruppo di sicurezza dell'istanza database possa ricevere traffico in ingresso dal gruppo di sicurezza della directory e inviare traffico in uscita alla directory.

Quando utilizzi la console per creare un'istanza database, scegli Password and Kerberos authentication (Password e autenticazione Kerberos) nella sezione Database authentication (Autenticazione database). Scegli Browse Directory (Sfoglia directory) quindi seleziona la directory oppure scegli Create a new directory (Crea una nuova directory).



Quando utilizzi la console per modificare o ripristinare un'istanza database, scegli la directory nella sezione Kerberos authentication (Autenticazione Kerberos) oppure scegli Create a new directory (Crea una nuova directory).



Quando si utilizza il AWS CLI, sono necessari i seguenti parametri affinché l'istanza DB possa utilizzare la directory creata:

- Per il parametro `--domain`, utilizza l'identificatore di dominio (identificatore "d-*") generato durante la creazione della directory.
- Per il parametro `--domain-iam-role-name`, utilizza il ruolo creato che utilizza la policy IAM gestita `AmazonRDSDirectoryServiceAccess`.

Ad esempio, il comando CLI seguente modifica un'istanza database per utilizzare una directory.

Per LinuxmacOS, oUnix:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --domain d-ID \
    --domain-iam-role-name role-name
```

Per Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --domain d-ID ^
    --domain-iam-role-name role-name
```

> **Important**
>
> Se modifichi un'istanza database per abilitare l'autenticazione Kerberos, riavvia l'istanza database dopo aver apportato la modifica.

> **Note**
>
> *MANAGED_SERVICE_USER* è un account di servizio il cui nome viene generato in modo casuale dal servizio directory per RDS. Durante l'impostazione dell'autenticazione Kerberos, RDS per Oracle crea un utente con lo stesso nome e gli assegna il privilegio `CREATE SESSION`. L'utente database Oracle viene identificato esternamente come *MANAGED_SERVICE_USER@EXAMPLE.COM*, dove *EXAMPLE.COM* è il nome del dominio. Periodicamente, RDS utilizza le credenziali fornite dal servizio directory per accedere al database Oracle. Successivamente, RDS distrugge immediatamente la cache dei ticket.

Fase 7: creazione di login Oracle di autenticazione Kerberos

Usa le credenziali dell'utente master Amazon RDS per eseguire la connessione all'istanza database Oracle come con qualunque altra istanza database. L'istanza DB viene aggiunta al AWS Managed Microsoft AD dominio. Pertanto, puoi eseguire il provisioning di login e utenti Oracle da utenti e gruppi Microsoft Active Directory nel dominio. Per gestire autorizzazioni del database, concedi e revoca autorizzazioni Oracle standard a questi login.

Per consentire a un utente di Microsoft Active Directory di eseguire l'autenticazione con Oracle

1. Per connetterti all'istanza database Oracle utilizza invece le credenziali dell'utente master Amazon RDS.
2. Crea un utente autenticato esternamente nel database Oracle.

   Nell'esempio seguente, sostituisci *KRBUSER@CORP.EXAMPLE.COM* con il nome utente e il nome di dominio.

   ```
   CREATE USER "KRBUSER@CORP.EXAMPLE.COM" IDENTIFIED EXTERNALLY;
   GRANT CREATE SESSION TO "KRBUSER@CORP.EXAMPLE.COM";
   ```

   Gli utenti (persone e applicazioni) del dominio possono ora connettersi all'istanza database Oracle da un computer client associato al dominio utilizzando l'autenticazione Kerberos.

Fase 8: configurazione di un client Oracle

Per configurare un client Oracle, devi rispettare i requisiti seguenti:

- Crea un file di configurazione denominato krb5.conf (Linux) o krb5.ini (Windows) che faccia riferimento al dominio. Per utilizzare questo file di configurazione, configura il client Oracle.
- Verifica che il traffico possa fluire tra l'host client e la porta DNS 53 AWS Directory Service su TCP/UDP, le porte Kerberos (88 e 464 per quelle gestite AWS Directory Service) su TCP e la porta LDAP 389 su TCP.
- Verifica che il traffico scorra senza problemi tra l'host client e l'istanza database sulla porta del database.

Di seguito è riportato un esempio di contenuto per. AWS Managed Microsoft AD

```
[libdefaults]
 default_realm = EXAMPLE.COM
[realms]
 EXAMPLE.COM = {
  kdc = example.com
  admin_server = example.com
 }
[domain_realm]
 .example.com = CORP.EXAMPLE.COM
 example.com = CORP.EXAMPLE.COM
```

Di seguito è riportato un esempio di contenuto per Microsoft AD on-premise. Nel file krb5.conf o krb5.ini, sostituisci *on-prem-ad-server-name con il nome* del server AD locale.

```
[libdefaults]
 default_realm = ONPREM.COM
[realms]
 AWSAD.COM = {
  kdc = awsad.com
  admin_server = awsad.com
 }
 ONPREM.COM = {
  kdc = on-prem-ad-server-name
  admin_server = on-prem-ad-server-name
 }
[domain_realm]
 .awsad.com = AWSAD.COM
 awsad.com= AWSAD.COM
 .onprem.com = ONPREM.COM
 onprem.com= ONPREM.COM
```

> **Note**
>
> Dopo aver configurato il file krb5.ini o krb5.conf, riavvia il server.

Di seguito viene fornito il contenuto di esempio del file sqlnet.ora per una configurazione SQL*Plus:

```
SQLNET.AUTHENTICATION_SERVICES=(KERBEROS5PRE,KERBEROS5)
SQLNET.KERBEROS5_CONF=path_to_krb5.conf_file
```

Per un esempio di una configurazione di SQL Developer, consulta Documento 1609359.1 del supporto di Oracle.

Gestione di un'istanza database in un dominio

Puoi utilizzare la console, la CLI o l'API RDS per gestire l'istanza database e la sua relazione con Microsoft Active Directory. Ad esempio, puoi associare una Microsoft Active Directory per abilitare l'autenticazione Kerberos. Puoi anche annullare l'associazione di una Microsoft Active Directory per disabilitare l'autenticazione Kerberos. Puoi anche spostare un'istanza database affinché venga autenticata esternamente da una Microsoft Active Directory a un'altra.

Ad esempio, utilizzando la CLI, puoi effettuare quanto segue:

- Per tentare nuovamente di abilitare Autenticazione Kerberos per un'appartenenza non riuscita, utilizza il comando CLI modify-db-instance e specifica l'ID directory dell'appartenenza corrente per l'opzione `--domain`.
- Per disabilitare Autenticazione Kerberos su un'istanza database, utilizza il comando CLI modify-db-instance e specifica `none` per l'opzione `--domain`.
- Per spostare un'istanza database da un dominio a un altro, utilizza il comando CLI modify-db-instance e specifica l'identificatore di dominio del nuovo dominio per l'opzione `--domain`.

Visualizzazione dello stato dell'appartenenza al dominio

Quando l'istanza database viene creata o modificata, questa diventa membro del dominio. Puoi visualizzare lo stato dell'appartenenza al dominio per l'istanza database nella console eseguendo il comando CLI describe-db-instances. Lo stato dell'istanza di database può essere uno dei seguenti:

- `kerberos-enabled`: l'autenticazione Kerberos è abilitata nell'istanza database.

- `enabling-kerberos`: AWS si trova nella fase di abilitazione dell'autenticazione Kerberos su questa istanza database.
- `pending-enable-kerberos`: l'abilitazione dell'autenticazione Kerberos è in corso su questa istanza database.
- `pending-maintenance-enable-kerberos`: AWS proverà ad abilitare l'autenticazione Kerberos sull'istanza database durante la prossima finestra di manutenzione pianificata.
- `pending-disable-kerberos`: la disabilitazione dell'autenticazione Kerberos è in corso su questa istanza database.
- `pending-maintenance-disable-kerberos`: AWS proverà a disabilitare l'autenticazione Kerberos sull'istanza database durante la prossima finestra di manutenzione pianificata.
- `enable-kerberos-failed`: un problema di configurazione ha impedito a AWS di abilitare l'autenticazione Kerberos sull'istanza database. Correggi il problema di configurazione prima di inviare nuovamente il comando per modificare l'istanza database.
- `disabling-kerberos`: AWS si trova nella fase di disabilitazione dell'autenticazione Kerberos su questa istanza database.

Una richiesta per abilitare l'autenticazione Kerberos potrebbe non andare a buon fine a causa di un problema di connettività di rete o un ruolo IAM non corretto. Se il tentativo di abilitare l'autenticazione Kerberos non va a buon fine quando crei o modifichi un'istanza database, assicurati innanzitutto di utilizzare il ruolo IAM corretto. Quindi, modifica l'istanza database per l'aggiunta al dominio

> **Note**
>
> Solo l'autenticazione Kerberos con Amazon RDS for Oracle invia traffico ai server DNS del dominio. Tutte le altre richieste DNS vengono gestite come accesso di rete in uscita sulle istanze database che eseguono Oracle. Per ulteriori informazioni sull'accesso di rete in uscita con Amazon RDS for Oracle, consulta Impostazione di un server DNS personalizzato.

Chiavi Kerberos con rotazione forzata

Una chiave segreta viene condivisa tra AWS Managed Microsoft AD e l'istanza database Amazon RDS for Oracle. Questa chiave viene ruotata automaticamente ogni 45 giorni. Puoi utilizzare la procedura Amazon RDS seguente per forzare la rotazione di questa chiave.

```
SELECT rdsadmin.rdsadmin_kerberos_auth_tasks.rotate_kerberos_keytab AS TASK_ID FROM
 DUAL;
```

> **Note**
>
> In una configurazione replica di lettura, questa procedura è disponibile solo nell'istanza database di origine e non nella replica di lettura.

L'istruzione `SELECT` restituisce l'ID dell'attività in un tipo di dati `VARCHAR2`. È possibile visualizzare lo stato di un'attività in corso in un file bdump. I file bdump si trovano nella directory `/rdsdbdata/log/trace`. Il nome del file bdump ha il formato che segue.

```
dbtask-task-id.log
```

È possibile visualizzare il risultato visualizzando il file di output dell'attività.

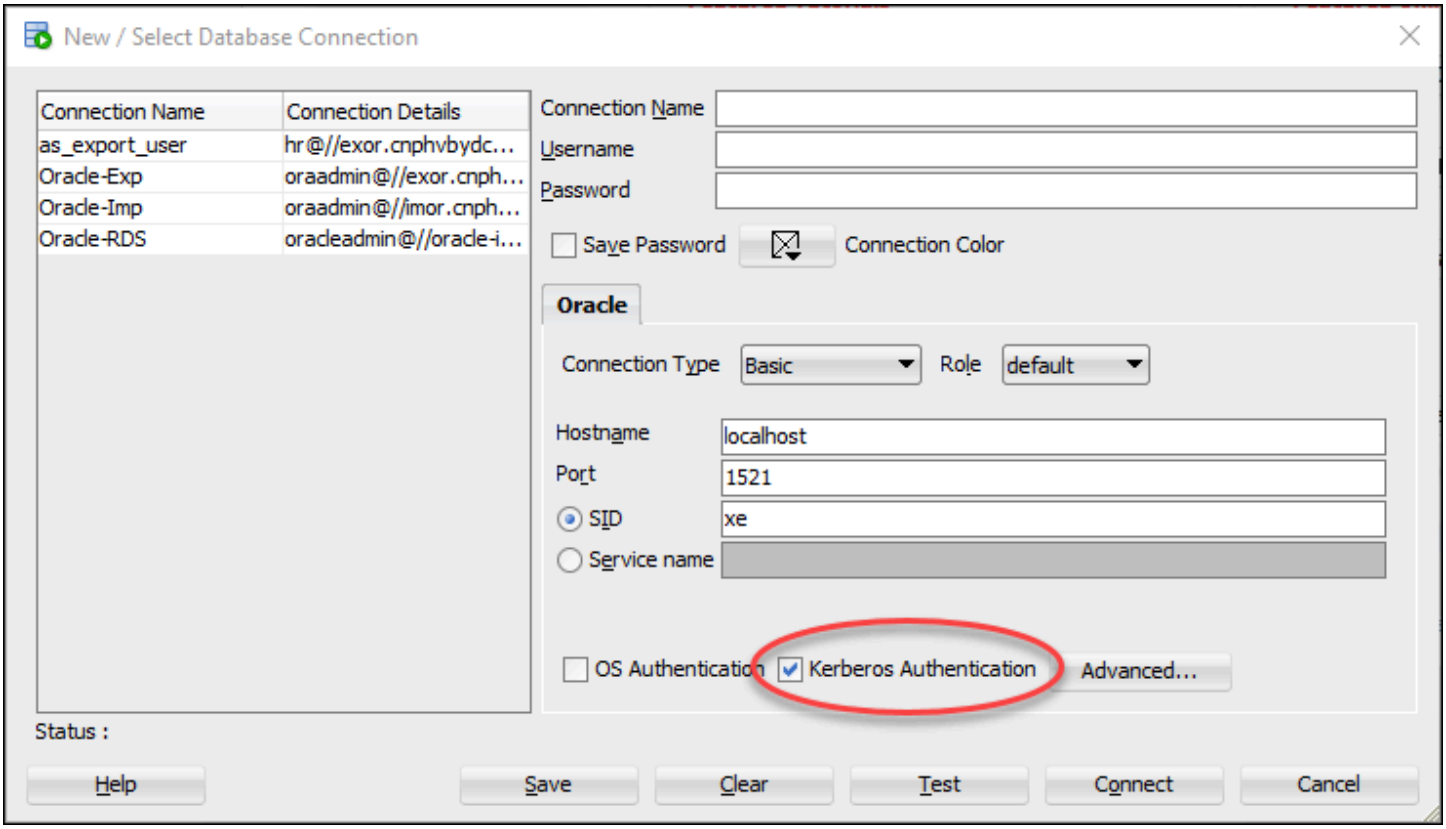
```
SELECT text FROM table(rdsadmin.rds_file_util.read_text_file('BDUMP','dbtask-task-
id.log'));
```

Sostituire *`task-id`* con l'ID attività restituito dalla procedura.

> **Note**
>
> Le attività vengono eseguite in modo asincrono.

Connessione a Oracle con Autenticazione Kerberos

In questa sezione si assume che il client Oracle sia stato configurato come descritto in Fase 8: configurazione di un client Oracle. Per connetterti a Oracle DB con Autenticazione Kerberos, accedi utilizzando il tipo di autenticazione Kerberos. Ad esempio, dopo aver avviato Oracle SQL Server, scegli Autenticazione Kerberos come tipo di autenticazione, come mostrato di seguito.



Per connettersi a Oracle con l'autenticazione Kerberos con SQL*Plus:

1. Al prompt dei comandi, esegui il comando seguente:

   ```
   kinit username
   ```

   Sostituisci *username* con il nome utente e, quando richiesto, immetti la password memorizzata nella Microsoft Active Directory per l'utente.

2. Apri SQL*Plus ed esegui la connessione utilizzando il nome DNS e il numero di porta per l'istanza database Oracle.

   Per ulteriori informazioni sulla connessione a un'istanza database Oracle in SQL*Plus, consulta Connessione all'istanza database tramite SQL*Plus.

Configurazione dell'accesso UTL_HTTP utilizzando certificati e un portafoglio Oracle

Amazon RDS supporta l'accesso alla rete in uscita sulle tue istanze DB RDS per Oracle. Per connettere l'istanza database alla rete, puoi utilizzare i seguenti pacchetti PL/SQL:

`UTL_HTTP`

Questo pacchetto effettua chiamate HTTP da SQL e PL/SQL. Puoi utilizzarlo per accedere ai dati su Internet tramite HTTP. Per ulteriori informazioni, consulta UTL_HTTP nella documentazione di Oracle.

`UTL_TCP`

Questo pacchetto fornisce funzionalità di accesso lato client TCP/IP in PL/SQL. Questo pacchetto è utile per le applicazioni PL/SQL che usano protocolli Internet ed e-mail. Per ulteriori informazioni, consulta UTL_TCP nella documentazione di Oracle.

`UTL_SMTP`

Questo pacchetto fornisce interfacce ai comandi SMTP che consentono a un client di inviare messaggi di posta elettronica a un server SMTP. Per ulteriori informazioni, consulta UTL_SMTP nella documentazione di Oracle.

Completando le seguenti attività, puoi configurare `UTL_HTTP.REQUEST` affinché funzioni con siti Web che richiedono certificati di autenticazione client durante l'handshake SSL. Puoi anche configurare l'autenticazione con password affinché `UTL_HTTP` acceda ai siti Web modificando i comandi di generazione del portafoglio Oracle wallet e la procedura `DBMS_NETWORK_ACL_ADMIN.APPEND_WALLET_ACE`. Per ulteriori informazioni, consulta DBMS_NETWORK_ACL_ADMIN nella documentazione di Oracle Database.

> **Note**
>
> Puoi adattare le seguenti attività per `UTL_SMTP`, il che consente di inviare e-mail tramite SSL/TLS (incluso Amazon Simple Email Service).

Argomenti

- Considerazioni sulla configurazione dell'accesso UTL_HTTP

- Passaggio 1: ottieni il certificato root per un sito Web
- Passaggio 2: crea un portafoglio Oracle
- Passaggio 3: scarica il tuo portafoglio Oracle nella tua istanza RDS for Oracle
- Passaggio 4: concedi le autorizzazioni utente per il portafoglio Oracle
- Passaggio 5: configura l'accesso a un sito Web dall'istanza database
- Passaggio 6: testa le connessioni dall'istanza database a un sito Web

Considerazioni sulla configurazione dell'accesso UTL_HTTP

Prima di configurare l'accesso, considerate quanto segue:

- È possibile utilizzare SMTP con l'opzione UTL_MAIL. Per ulteriori informazioni, consulta UTL_MAIL di Oracle.
- Il nome del DNS (Domain Name Server) dell'host remoto può essere uno qualsiasi dei seguenti:
  - Risolvibile pubblicamente.
  - L'endpoint di un'istanza database Amazon RDS.
  - Risolvibile attraverso un server DNS personalizzato. Per ulteriori informazioni, consulta Impostazione di un server DNS personalizzato.
  - Il nome DNS privato di un'istanza Amazon EC2 nello stesso VPC o in un VPC in peering. In questo caso, assicurarsi che il nome sia risolvibile attraverso un server DNS personalizzato. In alternativa, per utilizzare il DNS fornito da Amazon, è possibile abilitare l'attributo `enableDnsSupport` nelle impostazioni VPC e abilitare il supporto alla risoluzione DNS per la connessione VPC in peering. Per ulteriori informazioni, consulta la sezione sul supporto DNS nel tuo VPC e quella sulla modifica della tua connessione VPC in peering.
  - Per connetterti in modo sicuro a risorse SSL/TLS remote, ti consigliamo creare e caricare portafogli Oracle personalizzati. Utilizzando la funzionalità di integrazione di Amazon S3 con Amazon RDS for Oracle, è possibile scaricare un portafoglio da Amazon S3 sulle istanze database Oracle. Per informazioni sull'integrazione di Amazon S3 per Oracle, consultare Integrazione Amazon S3.
- Se l'opzione SSL Oracle è configurata per ogni istanza, è possibile stabilire collegamenti di database tra istanze database di Oracle tramite un endpoint SSL/TLS. Non è richiesta alcuna configurazione aggiuntiva. Per ulteriori informazioni, consulta Oracle Secure Sockets Layer.

Passaggio 1: ottieni il certificato root per un sito Web

Per consentire all'istanza DB RDS for Oracle di stabilire connessioni sicure a un sito Web, aggiungi il certificato CA principale. Amazon RDS utilizza il certificato root per firmare il certificato del sito Web al portafoglio Oracle.

È possibile ottenere il certificato root in vari modi. Ad esempio, puoi eseguire le operazioni seguenti:

1. Utilizza un server Web per visitare il sito Web protetto dal certificato.
2. Scarica il certificato root utilizzato per la firma.

Per i servizi AWS, i certificati root sono in genere disponibili nell'Amazon Trust Services Repository.

Passaggio 2: crea un portafoglio Oracle

Crea un portafoglio Oracle contenente sia i certificati del server Web che i certificati di autenticazione client. L'istanza Oracle RDS utilizza il certificato del server Web per stabilire una connessione sicura al sito Web. Il sito Web ha bisogno del certificato client per autenticare l'utente del database Oracle.

È possibile configurare connessioni sicure senza utilizzare certificati client per l'autenticazione. In questo caso, puoi ignorare i passaggi del keystore Java nella seguente procedura.

Per creare un portafoglio Oracle

1. Posiziona i certificati root e client in un'unica directory, quindi passa a questa directory.
2. Converti il certificato client .p12 nel keystore Java.

> **Note**
>
> Se non utilizzi certificati client per l'autenticazione, puoi ignorare questo passaggio.

Nell'esempio seguente viene convertito il certificato client denominato *client_certificate.p12* nel keystore Java denominato *client_keystore.jks*. Il keystore viene quindi incluso nel portafoglio Oracle. La password del keystore è *P12PASSWORD*.

```
orapki wallet pkcs12_to_jks -wallet ./client_certificate.p12 -
jksKeyStoreLoc ./client_keystore.jks -jksKeyStorepwd P12PASSWORD
```

3. Crea una directory per il tuo portafoglio Oracle diversa dalla directory dei certificati.

   Nell'esempio seguente viene creata la directory `/tmp/wallet`.

   ```
   mkdir -p /tmp/wallet
   ```

4. Crea un portafoglio Oracle nella directory del portafoglio.

   Nell'esempio seguente viene impostata la password del portafoglio Oracle su *P12PASSWORD*, che è la stessa password utilizzata dal keystore Java in un passaggio precedente. L'utilizzo della stessa password è comodo, ma non necessario. Il parametro `-auto_login` attiva la caratteristica di accesso automatico, in modo che non sia necessario specificare una password ogni volta che si desidera accedervi.

   > **Note**
   >
   > Specifica una password diversa dal prompt mostrato qui come best practice per la sicurezza.

   ```
   orapki wallet create -wallet /tmp/wallet -pwd P12PASSWORD -auto_login
   ```

5. Aggiungi il keystore Java al tuo portafoglio Oracle.

   > **Note**
   >
   > Se non utilizzi certificati client per l'autenticazione, puoi ignorare questo passaggio.

   Nell'esempio seguente viene aggiunto il keystore *client_keystore.jks* al portafoglio Oracle denominato */tmp/wallet*. In questo esempio, si specifica la stessa password del keystore Java e del portafoglio Oracle.

   ```
   orapki wallet jks_to_pkcs12 -wallet /tmp/wallet -pwd P12PASSWORD -
   keystore ./client_keystore.jks -jkspwd P12PASSWORD
   ```

6. Aggiungi il certificato root per il tuo sito Web di destinazione al portafoglio Oracle.

   Nell'esempio seguente viene aggiunto un certificato denominato *Root_CA.cer*.

```
orapki wallet add -wallet /tmp/wallet -trusted_cert -cert ./Root_CA.cer -
pwd P12PASSWORD
```

7. Aggiungi eventuali certificati intermedi.

   Nell'esempio seguente viene aggiunto un certificato denominato *Intermediate.cer*. Ripeti questo passaggio tutte le volte necessarie per caricare tutti i certificati intermedi.

   ```
   orapki wallet add -wallet /tmp/wallet -trusted_cert -cert ./Intermediate.cer -
   pwd P12PASSWORD
   ```

8. Verifica che il tuo portafoglio Oracle appena creato disponga dei certificati richiesti.

   ```
   orapki wallet display -wallet /tmp/wallet -pwd P12PASSWORD
   ```

Passaggio 3: scarica il tuo portafoglio Oracle nella tua istanza RDS for Oracle

In questo passaggio, carichi il tuo portafoglio Oracle su Amazon S3 e poi scarichi il portafoglio da Amazon S3 nella tua istanza RDS for Oracle.

Per scaricare il tuo portafoglio Oracle nella tua istanza database RDS for Oracle

1. Completare i prerequisiti per l'integrazione di Amazon S3 con Oracle e aggiungere l'opzione `S3_INTEGRATION` all'istanza database Oracle. Assicurarsi che il ruolo IAM per l'opzione abbia accesso al bucket Amazon S3 che si sta utilizzando.

   Per ulteriori informazioni, consulta Integrazione Amazon S3.

2. Accedei all'istanza database come utente principale e quindi crea una directory Oracle per contenere il portafoglio Oracle.

   Nell'esempio seguente viene creata una directory Oracle denominata *WALLET_DIR*.

   ```
   EXEC rdsadmin.rdsadmin_util.create_directory('WALLET_DIR');
   ```

   Per ulteriori informazioni, consulta Creazione ed eliminazione di directory nello spazio di archiviazione dati principale.

3. Carica il portafoglio Oracle sul tuo bucket Amazon S3.

Puoi utilizzare qualsiasi tecnica di caricamento supportata.

4. Se stai ricaricando un portafoglio Oracle, elimina il portafoglio esistente. Altrimenti, passare alla fase successiva.

   Nell'esempio seguente viene rimosso il portafoglio esistente denominato *cwallet.sso*.

   ```
   EXEC UTL_FILE.FREMOVE ('WALLET_DIR','cwallet.sso');
   ```

5. Scarica il portafoglio Oracle dal bucket Amazon S3 sull'istanza database Oracle.

   Nell'esempio seguente viene scaricato il portafoglio denominato *cwallet.sso* dal bucket Amazon S3 denominato *my_s3_bucket* nella directory dell'istanza database denominata *WALLET_DIR*.

   ```
   SELECT rdsadmin.rdsadmin_s3_tasks.download_from_s3(
         p_bucket_name    =>  'my_s3_bucket',
         p_s3_prefix      =>  'cwallet.sso',
         p_directory_name =>  'WALLET_DIR')
      AS TASK_ID FROM DUAL;
   ```

6. (Facoltativo) Scarica un portafoglio Oracle protetto da password.

   Scarica questo portafoglio solo se vuoi richiedere una password per ogni utilizzo del portafoglio. Nell'esempio seguente viene scaricato il portafoglio protetto da password *ewallet.p12*.

   ```
   SELECT rdsadmin.rdsadmin_s3_tasks.download_from_s3(
         p_bucket_name    =>  'my_s3_bucket',
         p_s3_prefix      =>  'ewallet.p12',
         p_directory_name =>  'WALLET_DIR')
      AS TASK_ID FROM DUAL;
   ```

7. Verifica lo stato dell'attività del database.

   Sostituisci l'ID attività restituito dai passaggi precedenti per *dbtask-1234567890123-4567.log* nell'esempio seguente.

   ```
   SELECT TEXT FROM
    TABLE(rdsadmin.rds_file_util.read_text_file('BDUMP','dbtask-1234567890123-4567.log'));
   ```

8. Controlla il contenuto della directory che stai utilizzando per memorizzare il portafoglio Oracle.

```
SELECT * FROM TABLE(rdsadmin.rds_file_util.listdir(p_directory => 'WALLET_DIR'));
```

Per ulteriori informazioni, consulta Generazione di un elenco dei file in una directory di istanze database.

Passaggio 4: concedi le autorizzazioni utente per il portafoglio Oracle

Puoi creare un nuovo utente database o configurare un utente esistente. In entrambi i casi, è necessario configurare l'utente per accedere al portafoglio Oracle per le connessioni sicure e l'autenticazione client tramite certificati.

Per concedere autorizzazioni utente per il portafoglio Oracle

1. Accedi all'istanza database RDS for Oracle come utente principale.
2. Se non desideri configurare un utente del database esistente, creare un nuovo utente. Altrimenti, passare alla fase successiva.

   Nell'esempio seguente viene creato un utente del database denominato *my-user*.

   ```
   CREATE USER my-user IDENTIFIED BY my-user-pwd;
   GRANT CONNECT TO my-user;
   ```
3. Concedi l'autorizzazione all'utente del database nella directory contenente il portafoglio Oracle.

   L'esempio seguente consente l'accesso in lettura all'utente *my-user* nella directory *WALLET_DIR*.

   ```
   GRANT READ ON DIRECTORY WALLET_DIR TO my-user;
   ```
4. Concedi l'autorizzazione all'utente del database per utilizzare il pacchetto UTL_HTTP.

   Il seguente programma PL/SQL garantisce a UTL_HTTP l'accesso all'utente *my-user*.

   ```
   BEGIN
     rdsadmin.rdsadmin_util.grant_sys_object('UTL_HTTP', UPPER('my-user'));
     END;
   /
   ```
5. Concedi l'autorizzazione all'utente del database per utilizzare il pacchetto UTL_FILE.

Il seguente programma PL/SQL garantisce a UTL_FILE l'accesso all'utente *my-user*.

```
BEGIN
  rdsadmin.rdsadmin_util.grant_sys_object('UTL_FILE', UPPER('my-user'));
  END;
/
```

Passaggio 5: configura l'accesso a un sito Web dall'istanza database

In questa fase, configuri l'utente del database Oracle in modo che possa connettersi al sito Web di destinazione utilizzando UTL_HTTP, il portafoglio Oracle caricato e il certificato client. Per ulteriori informazioni, consulta Configuring Access Control to an Oracle Wallet (Configurazione del controllo dell'accesso a un portafoglio Oracle) nella documentazione di Oracle Database.

Per configurare l'accesso a un sito Web dall'istanza database RDS for Oracle

1. Accedi all'istanza database RDS for Oracle come utente principale.
2. Crea una voce di controllo dell'accesso (ACE) host per il tuo utente e il sito Web di destinazione su una porta sicura.

   L'esempio seguente configura *my-user* per accedere a *secret.encrypted-website.com* sulla porta 443 sicura.

```
BEGIN
  DBMS_NETWORK_ACL_ADMIN.APPEND_HOST_ACE(
    host       => 'secret.encrypted-website.com',
    lower_port => 443,
    upper_port => 443,
    ace        => xs$ace_type(privilege_list => xs$name_list('http'),
                              principal_name => 'my-user',
                              principal_type => xs_acl.ptype_db));
                           -- If the program unit results in PLS-00201, set
                           -- the principal_type parameter to 2 as follows:
                           -- principal_type => 2));
END;
/
```

> **Important**
>
> L'unità di programma precedente può causare il seguente errore:. `PLS-00201: identifier 'XS_ACL' must be declared` Se viene restituito questo errore, sostituisci la riga che assegna un valore `principal_type` con la riga seguente, quindi riesegui l'unità di programma:
>
> ```
> principal_type => 2));
> ```
>
> Per ulteriori informazioni sulle costanti nel pacchetto PL/SQLXS_ACL, vedere Real Application Security Administrator's and Developer's Guide nella documentazione di Oracle Database.

Per ulteriori informazioni, consulta Configuring Access Control for External Network Services (Configurazione del controllo dell'accesso per servizi di rete esterni) nella documentazione di Oracle Database.

3. (Facoltativo) Crea un ACE per il tuo sito Web utente e di destinazione sulla porta standard.

   Potrebbe essere necessario utilizzare la porta standard se alcune pagine Web sono servite dalla porta standard del server Web (80) anziché dalla porta sicura (443).

   ```
   BEGIN
     DBMS_NETWORK_ACL_ADMIN.APPEND_HOST_ACE(
       host       => 'secret.encrypted-website.com',
       lower_port => 80,
       upper_port => 80,
       ace        => xs$ace_type(privilege_list => xs$name_list('http'),
                                 principal_name => 'my-user',
                                 principal_type => xs_acl.ptype_db));
                               -- If the program unit results in PLS-00201, set
                               -- the principal_type parameter to 2 as follows:
                               -- principal_type => 2));
   END;
   /
   ```

4. Verifica che le voci di controllo dell'accesso siano presenti.

   ```
   SET LINESIZE 150
   ```

```
COLUMN HOST FORMAT A40
COLUMN ACL FORMAT A50

SELECT HOST, LOWER_PORT, UPPER_PORT, ACL
  FROM DBA_NETWORK_ACLS
ORDER BY HOST;
```

5. Concedi l'autorizzazione all'utente del database per utilizzare il pacchetto UTL_HTTP.

   Il seguente programma PL/SQL garantisce a UTL_HTTP l'accesso all'utente *my-user*.

```
BEGIN
  rdsadmin.rdsadmin_util.grant_sys_object('UTL_HTTP', UPPER('my-user'));
  END;
/
```

6. Conferma l'esistenza di liste di controllo dell'accesso correlate.

```
SET LINESIZE 150
COLUMN ACL FORMAT A50
COLUMN PRINCIPAL FORMAT A20
COLUMN PRIVILEGE FORMAT A10

SELECT ACL, PRINCIPAL, PRIVILEGE, IS_GRANT,
       TO_CHAR(START_DATE, 'DD-MON-YYYY') AS START_DATE,
       TO_CHAR(END_DATE, 'DD-MON-YYYY') AS END_DATE
  FROM DBA_NETWORK_ACL_PRIVILEGES
ORDER BY ACL, PRINCIPAL, PRIVILEGE;
```

7. Concedi l'autorizzazione all'utente del database di utilizzare i certificati per l'autenticazione client e il portafoglio Oracle per le connessioni.

   > **Note**
   >
   > Se non utilizzi certificati client per l'autenticazione, puoi ignorare questo passaggio.

```
DECLARE
  l_wallet_path all_directories.directory_path%type;
BEGIN
  SELECT DIRECTORY_PATH
    INTO l_wallet_path
```

```
    FROM ALL_DIRECTORIES
   WHERE UPPER(DIRECTORY_NAME)='WALLET_DIR';
  DBMS_NETWORK_ACL_ADMIN.APPEND_WALLET_ACE(
    wallet_path => 'file:/' || l_wallet_path,
    ace         =>  xs$ace_type(privilege_list => xs
$name_list('use_client_certificates'),
                                principal_name => 'my-user',
                                principal_type => xs_acl.ptype_db));
END;
/
```

Passaggio 6: testa le connessioni dall'istanza database a un sito Web

In questa fase, configuri l'utente del database in modo che possa connettersi al sito Web utilizzando UTL_HTTP, il portafoglio Oracle caricato e il certificato client.

Per configurare l'accesso a un sito Web dall'istanza database RDS for Oracle

1. Accedi all'istanza database RDS for Oracle come utente del database con autorizzazioni UTL_HTTP.
2. Conferma che una connessione al sito Web di destinazione può risolvere l'indirizzo host.

   Nell'esempio seguente si ottiene l'indirizzo host da *secret.encrypted-website.com*.

   ```
   SELECT UTL_INADDR.GET_HOST_ADDRESS(host => 'secret.encrypted-website.com')
     FROM DUAL;
   ```
3. Testa una connessione non riuscita.

   La seguente query ha esito negativo perché UTL_HTTP richiede la posizione del portafoglio Oracle con i certificati.

   ```
   SELECT UTL_HTTP.REQUEST('secret.encrypted-website.com') FROM DUAL;
   ```
4. Verifica l'accesso al sito web utilizzando UTL_HTTP.SET_WALLET e selezionando da DUAL.

   ```
   DECLARE
     l_wallet_path all_directories.directory_path%type;
   BEGIN
     SELECT DIRECTORY_PATH
   ```

```
    INTO l_wallet_path
    FROM ALL_DIRECTORIES
   WHERE UPPER(DIRECTORY_NAME)='WALLET_DIR';
  UTL_HTTP.SET_WALLET('file:/' || l_wallet_path);
END;
/

SELECT UTL_HTTP.REQUEST('secret.encrypted-website.com') FROM DUAL;
```

5. (Facoltativo) Testa l'accesso al sito Web memorizzando la query in una variabile e utilizzando `EXECUTE IMMEDIATE`.

```
DECLARE
  l_wallet_path all_directories.directory_path%type;
  v_webpage_sql VARCHAR2(1000);
  v_results     VARCHAR2(32767);
BEGIN
  SELECT DIRECTORY_PATH
    INTO l_wallet_path
    FROM ALL_DIRECTORIES
   WHERE UPPER(DIRECTORY_NAME)='WALLET_DIR';
  v_webpage_sql := 'SELECT UTL_HTTP.REQUEST(''secret.encrypted-website.com'', '''',
 ''file:/' ||l_wallet_path||''') FROM DUAL';
  DBMS_OUTPUT.PUT_LINE(v_webpage_sql);
  EXECUTE IMMEDIATE v_webpage_sql INTO v_results;
  DBMS_OUTPUT.PUT_LINE(v_results);
END;
/
```

6. (Facoltativo) Individua la posizione del file system della directory del portafoglio Oracle.

```
SELECT * FROM TABLE(rdsadmin.rds_file_util.listdir(p_directory => 'WALLET_DIR'));
```

Utilizza l'output del comando precedente per effettuare una richiesta HTTP. Ad esempio, se la directory è *rdsdbdata/userdirs/01*, esegui la seguente query.

```
SELECT UTL_HTTP.REQUEST('https://secret.encrypted-website.com/', '',
 'file://rdsdbdata/userdirs/01')
FROM   DUAL;
```

Utilizzo di database CDB per RDS per Oracle

Nell'architettura multi-tenant Oracle, un database container (CDB) può includere database pluggable (PDB) creati dal cliente. Per ulteriori informazioni sui database CDB, consulta l'argomento relativo all'introduzione all'architettura multi-tenant nella documentazione di Oracle Database.

Argomenti

Panoramica dei database CDB RDS per Oracle

È possibile creare un'istanza database RDS per Oracle come database container (CDB) quando si esegue Oracle Database 19c o versione successiva. A partire da Oracle Database 21c, tutti i database sono CDB. Un CDB si differenzia da un non CDB perché può contenere database collegabili (PDB), denominati database tenant in RDS for Oracle. Un PDB è una raccolta portatile di schemi e oggetti visualizzato in un'applicazione come database distinto.

Il database dei tenant iniziali (PDB) viene creato quando si crea l'istanza CDB. In RDS for Oracle, l'applicazione client interagisce con un PDB anziché con il CDB. L'esperienza con un database PDB è per lo più identica all'esperienza con un database non CDB.

Argomenti

- Account utente e privilegi in un CDB
- Famiglie di gruppi di parametri in un CDB
- Limitazioni per i CDB RDS per Oracle

Configurazione multi-tenant dell'architettura CDB

RDS per Oracle supporta la configurazione multi-tenant dell'architettura multitenant Oracle, chiamata anche architettura CDB. In questa configurazione, l'istanza CDB di RDS for Oracle può contenere da 1 a 30 database tenant, a seconda dell'edizione del database e delle eventuali licenze opzionali richieste. Nel database Oracle, un database tenant è un PDB. L'istanza database deve utilizzare la versione del database Oracle 19.0.0.0.ru-2022-01.rur-2022.r1 o successiva.

> **Note**
>
> La funzionalità Amazon RDS è chiamata "multi-tenant" anziché "multitenant" perché è una funzionalità della piattaforma RDS, non solo del motore di database Oracle. Il termine "Oracle multitenant" si riferisce esclusivamente all'architettura del database Oracle, che è compatibile sia con le implementazioni on-premise che con quelle RDS.

È possibile configurare le seguenti impostazioni:

- Nome del database tenant
- Nome utente principale del database tenant
- Password principale del database tenant
- Set di caratteri del database tenant
- Set di caratteri nazionali del database tenant

Il set di caratteri del database del tenant può essere diverso dal set di caratteri del CDB. Lo stesso vale per il set di caratteri nazionali. Dopo aver creato il database del tenant iniziale, è possibile creare, modificare o eliminare il database del tenant utilizzando le API RDS. Per impostazione predefinita, il nome CDB viene impostato su `RDSCDB` e non può essere modificato. Per ulteriori informazioni, consulta Impostazioni per istanze database e Modifica di un database del tenant RDS per Oracle.

Configurazione a tenant singolo dell'architettura CDB

RDS per Oracle supporta una configurazione legacy dell'architettura multitenant Oracle chiamata configurazione a tenant singolo. In questa configurazione, un'istanza CDB RDS per Oracle può contenere un solo tenant (PDB). Non è possibile creare altri PDB in un secondo momento.

Opzioni di creazione e conversione per CDB

Oracle Database 21c supporta solo CDB mentre Oracle Database 19c supporta sia CDB che non CDB. Tutte le istanze CDB RDS per Oracle supportano sia la configurazione multi-tenant che quella a tenant singolo.

Opzioni di creazione, conversione e aggiornamento per l'architettura del database Oracle

La tabella seguente mostra le diverse opzioni di architettura per la creazione e l'aggiornamento di database RDS per Oracle.

Versione	Opzioni di creazione del database	Opzioni di conversione dell'architettura	Destinazioni di aggiornamento alla versione principale
Oracle Database 21c	Solo architettura CDB	N/D	N/D
Oracle Database 19c	Architettura CDB o non CDB	Architettura da non CDB a CDB (aggiornamento della versione di aprile 2021 o versione successiva)	CDB 21c
Oracle Database 12c (obsoleto)	Solo architettura non CDB	N/D	Non CDB 19c

Come mostrato nella tabella precedente, non è possibile aggiornare direttamente un database non CDB a database CDB in una nuova versione principale del database. Tuttavia, è possibile convertire un database non CDB Oracle Database 19c in database CDB Oracle Database 19c CDB e quindi aggiornarlo a database CDB Oracle Database 21c. Per ulteriori informazioni, consulta Conversione di un database non CDB RDS per Oracle in un database CDB.

Opzioni di conversione per le configurazioni dell'architettura CDB

La tabella seguente mostra le diverse opzioni per la conversione della configurazione dell'architettura di un'istanza database RDS per Oracle.

Architettura e configurazione correnti	Conversione dell'architettura CDB alla configurazione a tenant singolo	Conversione dell'architettura CDB alla configurazione multi-tenant	Conversione all'architettura non CDB
Non CDB	Supportato	Supportato*	N/D
CDB che utilizza la configurazione a tenant singolo	N/D	Supportato	Non supportato
CDB che utilizza la configurazione multi-tenant	Non supportato	N/D	Non supportato

* Non è possibile convertire un database non CDB in una configurazione multi-tenant in un'unica operazione. Quando si converte un database non CDB in un database CDB, il CDB si trova nella configurazione a tenant singolo. È possibile convertire la configurazione a tenant singolo in quella multi-tenant in un'operazione separata.

Account utente e privilegi in un CDB

Nell'architettura multi-tenant Oracle, tutti gli account utente sono utenti comuni o utenti locali. Un utente comune CDB è un utente del database la cui identità e password singole sono note nel root CDB e in ogni PDB esistente e futuro. Al contrario, un utente locale esiste solo in un unico PDB.

L'utente master RDS è un account utente locale nel PDB, a cui viene assegnato un nome quando si crea l'istanza database. Se crei nuovi account utente, anche questi utenti saranno utenti locali che risiedono nel PDB. Non è possibile utilizzare alcun account utente per creare nuovi PDF o modificare lo stato del PDB esistente.

L'utente `rdsadmin` è un account utente comune. È possibile eseguire pacchetti RDS per Oracle presenti in questo account, ma non è possibile accedere come `rdsadmin`. Per ulteriori informazioni,

consulta About Common Users and Local Users (Informazioni sugli utenti comuni e gli utenti locali) nella documentazione Oracle.

Famiglie di gruppi di parametri in un CDB

I CDB dispongono di classi di parametri e valori di parametro predefiniti propri. Le famiglie di gruppi di parametri CDB sono le seguenti:

- oracle-ee-cdb-21
- oracle-se2-cdb-21
- oracle-ee-cdb-19
- oracle-se2-cdb-19

Limitazioni per i CDB RDS per Oracle

RDS per Oracle supporta un sottoinsieme di funzionalità disponibili in un CDB on-premise.

Limitazioni dei CDB

Le seguenti limitazioni si applicano ai CDB RDS per Oracle:

- Non è possibile connettersi a un CDB. Ci si connette sempre al database del tenant (PDB) anziché al CDB. Specifica l'endpoint per il PDB proprio come per un non CDB. L'unica differenza è che si specifica pdb_name come nome del database, dove pdb_name è il nome scelto per il PDB.
- Non è possibile convertire un CDB nella configurazione multi-tenant in un CDB nella conversione a tenant singolo. La conversione alla configurazione multi-tenant è unidirezionale e irreversibile.
- Non è possibile abilitare o convertire in configurazione multi-tenant se l'istanza database utilizza una versione del database Oracle precedente a 19.0.0.0.ru-2022-01.rur-2022.r1.
- Non puoi utilizzare un CDB RDS per Oracle con ORDS v22 e versioni successive. Come soluzione alternativa, puoi invece utilizzare una versione precedente di ORDS o utilizzare un database non CDB Oracle Database 19c.
- Non è possibile utilizzare un RDS per Oracle CDB con ORDS 22 e versioni successive. Come soluzione alternativa, puoi invece utilizzare una versione precedente di ORDS o utilizzare un database non CDB Oracle Database 19c.

Il supporto per le seguenti funzionalità dipende dalla configurazione dell'architettura.

Funzionalità	Supportata nel tenant singolo	Supportata nel multi-tenant
Oracle Data Guard	Sì	No
Oracle Label Security	No	No
Oracle Enterprise Manager (OEM)	No	No
OEM Agent	No	No
Flussi di attività di database	Sì	No

Limitazioni del database del tenant (PDB)

Le seguenti limitazioni si applicano ai database del tenant con la configurazione multi-tenant RDS per Oracle:

- Non è possibile rimandare le operazioni del database del tenant alla finestra di manutenzione. Tutte le modifiche sono immediatamente effettive.
- Non è possibile aggiungere un database del tenant a un CDB che utilizza la configurazione a tenant singolo.
- Non è possibile aggiungere o modificare più database del tenant in un'unica operazione. È possibile aggiungerli o modificarli solo uno alla volta.
- Non è possibile modificare un database del tenant assegnandogli il nome `CDB$ROOT` o `PDB$SEED`.
- Non è possibile eliminare un database del tenant se è l'unico tenant nel CDB.
- Non tutti i tipi di classe di istanza database dispongono di risorse sufficienti per supportare più PDB in un'istanza CDB RDS per Oracle. Un numero maggiore di PDB influisce sulle prestazioni e sulla stabilità delle classi di istanza più piccole e aumenta il tempo della maggior parte delle operazioni a livello di istanza, ad esempio gli aggiornamenti del database.
- Non è possibile utilizzarne più di uno Account AWS per creare PDB nello stesso CDB. I PDB devono appartenere allo stesso account dell'istanza database su cui sono ospitati.
- Tutti i PDB in un CDB utilizzano lo stesso endpoint e lo stesso ascoltatore di database.
- Le seguenti operazioni non sono supportate a livello di PDB ma sono supportate a livello di CDB:
  - Backup e ripristino

  - Aggiornamenti del database
  - Operazioni di manutenzione
- Le seguenti funzionalità non sono supportate a livello di PDB ma sono supportate a livello di CDB:
  - Gruppi di opzioni (le opzioni sono installate in tutti i PDB dell'istanza CDB)
  - Gruppi di parametri (tutti i parametri derivano dal gruppo di parametri associato all'istanza CDB)
- Le operazioni a livello di PDB supportate nell'architettura CDB on-premise ma non supportate in un CDB RDS per Oracle sono:

> **Note**
>
> Il seguente elenco non è esaustivo.

  - PDB di applicazioni
  - PDB di proxy
  - Avvio e arresto di un PDB
  - Scollegamento e collegamento nei PDB

    Per spostare i dati dentro o fuori dal CDB, utilizzare le stesse tecniche valide per un database non CDB. Per ulteriori informazioni sulla migrazione dei dati, consulta Importazione di dati in Oracle in Amazon RDS.
  - Impostazione delle opzioni a livello di PDB

    Il PDB eredita le impostazioni delle opzioni dal gruppo di opzioni del CDB. Per ulteriori informazioni sulle impostazioni delle opzioni, consulta Utilizzo di gruppi di parametri. Per le best practice, consulta Utilizzo di gruppi di parametri di database.
  - Configurazione dei parametri in un PDB

    Il PDB eredita le impostazioni dei parametri dal CDB. Per ulteriori informazioni sull'impostazione dell'opzione, consulta Aggiunta di opzioni alle istanze database Oracle.
  - Configurazione di diversi ascoltatori per i PDB nello stesso CDB
  - Funzionalità di Oracle Flashback
  - Audit delle informazioni da un PDB

Configurazione di un CDB RDS per Oracle

La configurazione di un CDB è simile alla configurazione di un database non CDB.

Argomenti

Creazione di un'istanza CDB RDS per Oracle

In RDS per Oracle, la creazione di un CDB è quasi identica alla creazione di un database non CDB. La differenza risiede nella selezione dell'architettura multitenant Oracle quando viene creata l'istanza database e viene scelta anche la configurazione dell'architettura: multi-tenant o tenant singolo. Se si creano i tag quando si crea un CDB nella configurazione multi-tenant, RDS propaga i tag al database del tenant iniziale. Per creare un CDB, utilizza la AWS Management Console, la AWS CLI, l'API RDS.

Console

Creazione di un'istanza database

1. Accedi alla AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nell'angolo in alto a destra della console Amazon RDS, scegliere la Regione AWS in cui creare l'istanza CDB.
3. Nel riquadro di navigazione, scegliere Databases (Database).
4. Scegliere Create database (Crea database).
5. In Choose a database creation method (Seleziona metodo di creazione del database), scegliere Standard Create (Creazione standard).
6. In Engine options (Opzioni motore), selezionare Oracle.
7. In Tipo di gestione del database, selezionare Amazon RDS.
8. In Impostazioni dell'architettura, scegli Architettura multitenant Oracle.
9. Per Configurazione dell'architettura, effettua una delle seguenti operazioni:
   - Scegli Configurazione multi-tenant e procedi con il passaggio successivo.
   - Scegli Configurazione a tenant singolo e vai alla fase 11.
10. (Configurazione multi-tenant) Per Impostazioni globali del tenant, apporta le seguenti modifiche:

- Per Nome del database tenant immetti il nome del PDB. Il nome del PDB deve essere diverso dal nome del CDB, che per impostazione predefinita è `RDSCDB`.
- Per Nome utente principale del database tenant, specifica il nome dell'utente principale del PDB. Non è possibile utilizzare il nome utente principale del database del tenant per accedere al CDB.
- Inserisci una password in Password principale del database tenant o scegli Genera automaticamente una password.
- Per Set di caratteri del database tenant, scegli un set di caratteri per il PDB. È possibile scegliere un set di caratteri del database del tenant diverso dal set di caratteri del CDB.

  Il set di caratteri del PDB predefinito è AL32UTF8. Se scegli un set di caratteri del PDB non predefinito, la creazione del CDB potrebbe essere più lenta.

> **Note**
>
> Non è possibile creare più database del tenant come parte del processo di creazione del CDB. È possibile aggiungere i PDB solo a un CDB esistente.

11. (Configurazione a tenant singolo) Scegli le impostazioni desiderate in base alle opzioni elencate in Impostazioni per istanze database. Tieni presente quanto segue:
    - In Nome utente master, immettere il nome di un utente locale nel PDB. Non è possibile utilizzare il nome utente master per accedere alla root del CDB.
    - In Nome database iniziale immettere il nome del PDB. Non è possibile assegnare un nome al CDB, che ha il nome predefinito `RDSCDB`.
12. Scegliere Crea database.

AWS CLI

Per creare un CDB nella configurazione multi-tenant, utilizzate il create-db-instancecomando con i seguenti parametri:

- `--db-instance-identifier`
- `--db-instance-class`
- `--engine { oracle-ee-cdb | oracle-se2-cdb }`

- `--master-username`
- `--master-user-password`
- `--multi-tenant` (per la configurazione a tenant singolo, non specificare `multi-tenant` o specifica `--no-multi-tenant`)
- `--allocated-storage`
- `--backup-retention-period`

Per informazioni su ciascuna impostazione, consulta Impostazioni per istanze database.

L'esempio seguente crea un'istanza DB RDS for Oracle denominata *my-cdb-inst*nella configurazione multi-tenant. Se si specifica `--no-multi-tenant` o non si specifica `--multi-tenant`, la configurazione CDB predefinita è a tenant singolo. Il motore è `oracle-ee-cdb`: un comando che specifica `oracle-ee` e `--multi-tenant` non riesce con un errore. Il database del tenant iniziale è denominato *mypdb*.

Example

PerLinux, o: macOS Unix

```
aws rds create-db-instance \
    --engine oracle-ee-cdb \
    --db-instance-identifier my-cdb-inst \
    --multi-tenant \
    --db-name mypdb \
    --allocated-storage 250 \
    --db-instance-class db.t3.large \
    --master-username pdb_admin \
    --master-user-password pdb_admin_password \
    --backup-retention-period 3
```

Per Windows:

```
aws rds create-db-instance ^
    --engine oracle-ee-cdb ^
    --db-instance-identifier my-cdb-inst ^
    --multi-tenant ^
    --db-name mypdb ^
    --allocated-storage 250 ^
    --db-instance-class db.t3.large ^
    --master-username pdb_admin ^
```

```
    --master-user-password pdb_admin_password ^
    --backup-retention-period 3
```

> **Note**
>
> Specifica una password diversa dal prompt mostrato qui come best practice per la sicurezza.

Questo comando genera un output simile al seguente. Il nome del database, il set di caratteri, il set di caratteri nazionali e l'utente principale non sono inclusi nell'output. È possibile visualizzare queste informazioni utilizzando il comando CLI `describe-tenant-databases`.


```
{
    "DBInstance": {
        "DBInstanceIdentifier": "my-cdb-inst",
        "DBInstanceClass": "db.t3.large",
        "MultiTenant": true,
        "Engine": "oracle-ee-cdb",
        "DBResourceId": "db-ABCDEFGJIJKLMNOPQRSTUVWXYZ",
        "DBInstanceStatus": "creating",
        "AllocatedStorage": 250,
        "PreferredBackupWindow": "04:59-05:29",
        "BackupRetentionPeriod": 3,
        "DBSecurityGroups": [],
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sg-0a1bcd2e",
                "Status": "active"
            }
        ],
        "DBParameterGroups": [
            {
                "DBParameterGroupName": "default.oracle-ee-cdb-19",
                "ParameterApplyStatus": "in-sync"
            }
        ],
        "DBSubnetGroup": {
            "DBSubnetGroupName": "default",
            "DBSubnetGroupDescription": "default",
            "VpcId": "vpc-1234567a",
            "SubnetGroupStatus": "Complete",
            ...
```

API RDS

Per creare un'istanza database tramite l'API Amazon RDS, chiama l'operazione CreateDBInstance.

Per informazioni su ciascuna impostazione, consulta Impostazioni per istanze database.

Connessione a un PDB nel CDB RDS per Oracle

È possibile utilizzare un'utilità come SQL*Plus per connettersi a un PDB. Per scaricare Oracle Instant Client, che include una versione autonoma di SQL*Plus, consulta Download di Oracle Instant Client.

Per stabilire la connessione tra SQL*Plus e il PDB, è necessario includere le seguenti informazioni:

- Nome del PDB
- Nome utente e password del database
- Endpoint dell'istanza database
- Numero della porta

Per informazioni su come trovare le informazioni precedenti, consulta Esito dell'endpoint dell'istanza database RDS per Oracle.

Example Connessione al PDB tramite SQL*Plus

Negli esempi seguenti, sostituire *master_user_name* con l'utente master. Sostituire inoltre l'endpoint dell'istanza database e quindi includere il numero di porta e il SID Oracle. Il valore SID è il nome del PDB specificato al momento della creazione dell'istanza database e non l'identificatore dell'istanza database.

Per LinuxmacOS, oUnix:

```
sqlplus 'master_user_name@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=endpoint)
(PORT=port))(CONNECT_DATA=(SID=pdb_name)))'
```

Per Windows:

```
sqlplus master_user_name@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=endpoint)
(PORT=port))(CONNECT_DATA=(SID=pdb_name)))
```

Verrà visualizzato un output simile al seguente.

```
SQL*Plus: Release 19.0.0.0.0 Production on Mon Aug 21 09:42:20 2021
```

Dopo l'immissione della password dell'utente, verrà visualizzato il prompt SQL.

```
SQL>
```

> **Note**
>
> Il formato più breve della stringa di connessione (Easy connect o EZCONNECT), ad esempio `sqlplus` *`username`*`/`*`password`*`@`*`LONGER-THAN-63-CHARS-RDS-ENDPOINT-HERE`*`:1521/`*`database-identifier`*, potrebbe comportare un limite relativo al numero massimo di caratteri e non deve pertanto essere utilizzato per la connessione.

Backup e ripristino di un CDB

È possibile eseguire il backup e il ripristino del CDB utilizzando snapshot di database RDS o Recovery Manager (RMAN).

Backup e ripristino di un CDB utilizzando snapshot di database

Gli snapshot di database funzionano in modo simile nelle architetture CDB e non CDB. Le differenze principali sono le seguenti:

- Quando ripristini uno snapshot di database di un CDB, non puoi rinominare il CDB. Per impostazione predefinita, il CDB viene denominato `RDSCDB` e il nome non può essere modificato.
- Quando ripristini uno snapshot di database di un CDB, non puoi rinominare i PDB. È possibile modificare il nome del PDB utilizzando il comando modify-tenant-database.
- Per trovare i database del tenant in uno snapshot, utilizza il comando della CLI describe-db-snapshot-tenant-database.
- Non è possibile interagire direttamente con i database del tenant in uno snapshot CDB che utilizza la configurazione dell'architettura multi-tenant. Se ripristini lo snapshot di database, ripristini tutti i relativi database del tenant.
- RDS per Oracle copia implicitamente i tag di un database del tenant nel database del tenant di uno snapshot di database. Quando si ripristina un database del tenant, i tag vengono visualizzati nel database ripristinato.

- Se si ripristina uno snapshot di database e si specificano nuovi tag utilizzando il parametro `--tags`, i nuovi tag sovrascrivono tutti i tag esistenti.
- Se si esegue uno snapshot di database di un'istanza CDB con tag e si specifica `--copy-tags-to-snapshot`, RDS per Oracle copia i tag dai database del tenant ai database del tenant presenti nello snapshot.

Per ulteriori informazioni, consulta Considerazioni su Oracle Database.

Backup e ripristino di un CDB utilizzando RMAN

Per informazioni su come eseguire il backup e il ripristino di un CDB o di un database a tenant singolo utilizzando RMAN, consulta Esecuzione di attività RMAN comuni per le istanze database Oracle.

Conversione di un database non CDB RDS per Oracle in un database CDB

È possibile modificare l'architettura di un database Oracle dall'architettura non CDB all'architettura multitenant Oracle, chiamata anche architettura CDB, con il comando. `modify-db-instance` Nella maggior parte dei casi, questa tecnica è preferibile alla creazione di un nuovo CDB e all'importazione di dati. L'operazione di conversione comporta tempi di inattività.

Quando si aggiorna la versione del motore di database, non è possibile modificare l'architettura del database durante la stessa operazione. Pertanto, per aggiornare un database non CDB Oracle Database 19c a un database CDB Oracle Database 21c, è necessario prima convertire il database non CDB in un database CDB in un passaggio, quindi aggiornare il CDB 19c risultante a database CDB 21c in un passaggio separato.

I requisiti dell'operazione di conversione non CDB sono elencati di seguito:

- È necessario specificare `oracle-ee-cdb` o `oracle-se2-cdb` per il tipo di motore di database. Questi sono gli unici valori supportati.
- Il motore di database deve utilizzare Oracle Database 19c con un aggiornamento della versione di aprile 2021 o successive.

L'operazione presenta le seguenti limitazioni:

- Non è possibile convertire un database CDB in un database non CDB. È solo possibile convertire un database non CDB in un database CDB.

- Non è possibile convertire una configurazione non CDB in una configurazione multi-tenant in una singola chiamata `modify-db-instance`. Dopo aver convertito un database non CDB in un database CDB, il CDB è nella configurazione a tenant singolo. Per convertire la configurazione a tenant singolo in quella multi-tenant, esegui nuovamente `modify-db-instance`. Per ulteriori informazioni, consulta Conversione della configurazione a tenant singolo in multi-tenant.
- Non è possibile convertire un database primario o di replica con Oracle Data Guard abilitato. Per convertire un non CDB con repliche di lettura, elimina prima tutte le repliche di lettura.
- Non è possibile aggiornare la versione del motore di database e convertire un database non CDB in un database CDB durante la stessa operazione.
- Le considerazioni relative ai gruppi di opzioni e parametri sono le stesse valide per l'aggiornamento del motore di database. Per ulteriori informazioni, consulta Considerazioni sugli aggiornamenti di Oracle DB.

Console

Conversione di un database non CDB in un database CDB

1. Accedi alla AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nell'angolo in alto a destra della console Amazon RDS, scegliere la Regione AWS in cui si trova l'istanza database.
3. Nel riquadro di navigazione, scegliere Database e quindi l'istanza non CDB da convertire in istanza CDB.
4. Scegli Modifica.
5. In Impostazioni dell'architettura, seleziona Architettura multitenant Oracle. Dopo la conversione, il CDB sarà nella configurazione a tenant singolo.
6. (Facoltativo) In Gruppo di parametri database, scegliere un nuovo gruppo di parametri per l'istanza CDB. Le stesse considerazioni relative ai gruppi di parametri valgono per la conversione di un'istanza database e per l'aggiornamento di un'istanza database. Per ulteriori informazioni, consulta Considerazioni sui gruppi di parametri.
7. (Facoltativo) In Gruppo di opzioni, selezionare un nuovo gruppo di opzioni per l'istanza CDB. Le stesse considerazioni relative ai gruppi di opzioni valgono per la conversione di un'istanza database e per l'aggiornamento di un'istanza database. Per ulteriori informazioni, consulta Considerazioni su gruppi di opzioni.

8. Quando tutte le modifiche sono come le desideri, seleziona Continue (Continua) e controlla il riepilogo delle modifiche.
9. (Facoltativo) Scegliere Applica immediatamente per applicare immediatamente le modifiche. In alcuni casi, la chiusura di questa opzione può causare un'interruzione. Per ulteriori informazioni, consulta Impostazione delle modifiche alla pianificazione.
10. Nella pagina di conferma esaminare le modifiche. Se sono corrette, selezionare Modifica istanza database.

    Oppure scegliere Back (Indietro) per cambiare le modifiche o Cancel (Annulla) per annullare le modifiche.

AWS CLI

Per convertire il contenuto non CDB dell'istanza DB in un CDB nella configurazione single-tenant, imposta su o nel comando. `--engine` `oracle-ee-cdb` `oracle-se2-cdb` AWS CLI modify-db-instance Per ulteriori informazioni, consulta Impostazioni per istanze database.

L'esempio seguente converte l'istanza DB denominata *my-non-cdb*e specifica un gruppo di opzioni e un gruppo di parametri personalizzati.

Example

PerLinux, omacOS: Unix

```
aws rds modify-db-instance \
    --db-instance-identifier my-non-cdb \
    --engine oracle-ee-cdb \
    --option-group-name custom-option-group \
    --db-parameter-group-name custom-parameter-group
```

Per Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier my-non-cdb ^
    --engine oracle-ee-cdb ^
    --option-group-name custom-option-group ^
    --db-parameter-group-name custom-parameter-group
```

API RDS

Per convertire un database non CDB in un database CDB, specificare `Engine` nell'operazione dell'API RDS l'operazione ModifyDBInstance.

Conversione della configurazione a tenant singolo in multi-tenant

È possibile modificare l'architettura di un CDB RDS per Oracle dalla configurazione a tenant singolo alla configurazione multi-tenant. Prima e dopo la conversione, il CDB contiene un database a tenant singolo (PDB).

Durante la conversione, RDS per Oracle migra i seguenti metadati nel nuovo database del tenant:

- Il nome utente principale
- Il nome del database
- Il set di caratteri
- Il set di caratteri nazionali

Prima della conversione, è possibile visualizzare le informazioni precedenti utilizzando il comando `describe-db-instances`. Dopo la conversione, è possibile visualizzare le informazioni utilizzando il comando `describe-tenant-database`.

La conversione presenta i seguenti requisiti e limitazioni:

- Dopo aver convertito la configurazione dell'architettura a tenant singolo in una configurazione multi-tenant, non è possibile riconvertire successivamente l'architettura nella configurazione a tenant singolo. L'operazione è irreversibile.
- I tag per l'istanza database si propagano al database del tenant iniziale creato durante la conversione.
- Non è possibile convertire un database primario o di replica con Oracle Data Guard abilitato.
- Non è possibile aggiornare la versione del motore di database e convertire nella configurazione multi-tenant durante la stessa operazione.
- La policy IAM deve disporre delle autorizzazioni per creare un database del tenant.

Console

Per convertire un CDB con la configurazione a tenant singolo in quella multi-tenant

1. Accedi alla AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nell'angolo in alto a destra della console Amazon RDS, scegliere la Regione AWS in cui si trova l'istanza database.
3. Nel riquadro di navigazione, scegliere Database e quindi l'istanza non CDB da convertire in istanza CDB.
4. Scegli Modifica.
5. In Impostazioni dell'architettura, seleziona Architettura multitenant Oracle.
6. Per Configurazione dell'architettura seleziona Configurazione multi-tenant.
7. (Facoltativo) In Gruppo di parametri database, scegliere un nuovo gruppo di parametri per l'istanza CDB. Le stesse considerazioni relative ai gruppi di parametri valgono per la conversione di un'istanza database e per l'aggiornamento di un'istanza database.
8. (Facoltativo) In Gruppo di opzioni, selezionare un nuovo gruppo di opzioni per l'istanza CDB. Le stesse considerazioni relative ai gruppi di opzioni valgono per la conversione di un'istanza database e per l'aggiornamento di un'istanza database.
9. Quando tutte le modifiche sono come le desideri, seleziona Continue (Continua) e controlla il riepilogo delle modifiche.
10. Scegliere Apply immediately (Applica immediatamente). Questa opzione è necessaria quando si passa a una configurazione multi-tenant. Tieni presente che questa opzione può causare tempi di inattività in alcuni casi.
11. Nella pagina di conferma esaminare le modifiche. Se sono corrette, selezionare Modifica istanza database.

    Oppure scegliere Back (Indietro) per cambiare le modifiche o Cancel (Annulla) per annullare le modifiche.

AWS CLI

Per convertire un CDB utilizzando la configurazione single-tenant in una configurazione multi-tenant, specificare nel comando. `--multi-tenant` AWS CLI modify-db-instance

L'esempio seguente converte l'istanza database denominata `my-st-cdb` dalla configurazione a tenant singolo alla configurazione multi-tenant. L'opzione `--apply-immediately` è obbligatoria.

Example

Per, o: Linux macOS Unix

```
aws rds modify-db-instance --region us-east-1\
    --db-instance-identifier my-st-cdb \
    --multi-tenant \
    --apply-immediately
```

Per Windows:

```
aws rds modify-db-instance --region us-east-1 ^
    --db-instance-identifier my-st-cdb ^
    --multi-tenant ^
    --apply-immediately
```

L'output è simile al seguente.

```
{
    "DBInstance": {
        "DBInstanceIdentifier": "my-st-cdb",
        "DBInstanceClass": "db.r5.large",
        "MultiTenant": false,
        "Engine": "oracle-ee-cdb",
        "DBResourceId": "db-AB1CDE2FGHIJK34LMNOPRLXTXU",
        "DBInstanceStatus": "modifying",
        "MasterUsername": "admin",
        "DBName": "ORCL",
        ...
        "EngineVersion": "19.0.0.0.ru-2022-01.rur-2022-01.r1",
        "AutoMinorVersionUpgrade": true,
        "ReadReplicaDBInstanceIdentifiers": [],
        "LicenseModel": "bring-your-own-license",
        "OptionGroupMemberships": [
            {
                "OptionGroupName": "default:oracle-ee-cdb-19",
                "Status": "in-sync"
            }
        ],
```

```
        ...
        "PendingModifiedValues": {
            "MultiTenant": "true"
        }
    }
}
```

Aggiunta di un database del tenant RDS per Oracle all'istanza CDB

Nella configurazione multi-tenant RDS per Oracle, un database del tenant è un PDB. Per aggiungere un database del tenant, verifica che siano soddisfatti i seguenti prerequisiti:

- Il CDB ha la configurazione multi-tenant abilitata. Per ulteriori informazioni, consulta Configurazione multi-tenant dell'architettura CDB.
- Disponi delle necessarie autorizzazioni IAM per creare il database del tenant.

È possibile aggiungere un database del tenant utilizzando la AWS Management Console, la AWS CLI o l'API RDS. Non è possibile aggiungere più database del tenant in un'unica operazione pertanto è necessario aggiungerli uno alla volta. Se il CDB ha la conservazione dei backup abilitata, Amazon RDS esegue il backup dell'istanza database prima e dopo l'aggiunta di un nuovo database del tenant.

Console

Per aggiungere un database del tenant all'istanza database

1. Accedi alla AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nell'angolo in alto a destra della console Amazon RDS, scegli la Regione AWS in cui desideri creare il database del tenant.
3. Nel riquadro di navigazione, scegli Databases (Database).
4. Scegli l'istanza CDB in cui aggiungere un database del tenant. L'istanza database deve utilizzare la configurazione multi-tenant dell'architettura CDB.
5. Scegli Operazioni, quindi Aggiungi database del tenant.
6. Per Impostazioni globali del database effettua le seguenti operazioni:
    - Per Nome del database tenant immetti il nome del nuovo PDB.
    - Per Nome utente principale del database tenant, specifica il nome dell'utente principale per il PDB. Questo utente principale è diverso dall'utente principale del CDB.

- Inserisci una password in Password principale del database tenant o seleziona Genera automaticamente una password.
- Per Set di caratteri del database tenant, scegli un set di caratteri per il PDB. L'impostazione predefinita è AL32UTF8. È possibile scegliere un set di caratteri del PDB diverso dal set di caratteri del CDB.
- Per Set di caratteri nazionali del database tenant, scegli un set di caratteri per il PDB. L'impostazione predefinita è AL32UTF8. Il set di caratteri nazionali specifica la codifica solo per le colonne che utilizzano il tipo di dati NCHAR (NCHAR, NVARCHAR2 e NCLOB) e non influisce sui metadati del database.

Per ulteriori informazioni sulle impostazioni precedenti, consulta Impostazioni per istanze database.

7. Scegli Aggiungi tenant.

AWS CLI

Per aggiungere un database tenant al tuo CDB conAWS CLI, usa il comando create-tenant-databasecon i seguenti parametri richiesti:

- `--db-instance-identifier`
- `--tenant-db-name`
- `--master-username`
- `--master-user-password`

L'esempio seguente crea un database tenant denominato *mypdb2* nell'istanza CDB di RDS for Oracle denominata. *my-cdb-inst* Il set di caratteri del PDB è UTF-16.

Example

Per, oLinux: macOS Unix

```
aws rds create-tenant-database --region us-east-1 \
    --db-instance-identifier my-cdb-inst \
    --tenant-db-name mypdb2 \
    --master-username mypdb2-admin \
    --master-user-password mypdb2-pwd \
```

```
    --character-set-name UTF-16
```

Per Windows:

```
aws rds create-tenant-database --region us-east-1 \
    --db-instance-identifier my-cdb-inst ^
    --tenant-db-name mypdb2 ^
    --master-username mypdb2-admin ^
    --master-user-password mypdb2-pwd ^
    --character-set-name UTF-16
```

L'output è simile al seguente.

```
...}
    "TenantDatabase" :
         {
            "DbiResourceId" : "db-abc123",
            "TenantDatabaseResourceId" : "tdb-bac567",
            "TenantDatabaseArn" : "arn:aws:rds:us-east-1:123456789012:db:my-cdb-
inst:mypdb2",
            "DBInstanceIdentifier" : "my-cdb-inst",
            "TenantDBName" : "mypdb2",
            "Status" : "creating",
            "MasterUsername" : "mypdb2",
            "CharacterSetName" : "UTF-16",
            ...
        }
}...
```

Modifica di un database del tenant RDS per Oracle

È possibile modificare solo il nome PDB e la password dell'utente principale di un database del tenant nel CDB. Tieni presenti i seguenti requisiti e limitazioni:

- Per modificare le impostazioni di un database del tenant nell'istanza database, il database del tenant deve esistere.
- Non è possibile modificare più database del tenant in un'unica operazione. È possibile modificare un solo database del tenant alla volta.
- Non è possibile modificare il nome di un database del tenant in `CDB$ROOT` o `PDB$SEED`.

Un PDB può essere modificato usando la AWS Management Console, la AWS CLI o l'API RDS.

Console

Per modificare il nome PDB o la password principale di un database del tenant

1. Accedi alla AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nell'angolo in alto a destra della console Amazon RDS, scegli la Regione AWS in cui desideri creare il database del tenant.
3. Nel riquadro di navigazione, scegli Databases (Database).
4. Scegli il database del tenant di cui desideri modificare il nome del database o la password dell'utente principale.
5. Scegli Modifica.
6. Per Impostazioni globali del database effettua una delle seguenti operazioni:
   - Per Nome del database tenant immetti il nuovo nome del nuovo PDB.
   - Per Password principale del database tenant, immetti una nuova password.
7. Scegli Modifica tenant.

AWS CLI

Per modificare un database tenant utilizzando ilAWS CLI, chiamate il modify-tenant-databasecomando con i seguenti parametri:

- `--db-instance-identifier` *valore*
- `--tenant-db-name` *value*
- `[--new-tenant-db-name` *value*`]`
- `[--master-user-password` *value*`]`

L'esempio seguente rinomina il database del tenant `pdb1` in `pdb-hr` nell'istanza database `my-cdb-inst`.

Example

Per LinuxmacOS, oUnix:

```
aws rds modify-tenant-database --region us-east-1 \
    --db-instance-identifier my-cdb-inst \
    --tenant-db-name pdb1 \
    --new-tenant-db-name pdb-hr
```

Per Windows:

```
aws rds modify-tenant-database --region us-east-1 ^
    --db-instance-identifier my-cdb-inst ^
    --tenant-db-name pdb1 ^
    --new-tenant-db-name pdb-hr
```

Questo comando genera un output simile al seguente.

```
{
    "TenantDatabase" : {
        "DbiResourceId" : "db-abc123",
        "TenantDatabaseResourceId" : "tdb-bac567",
        "TenantDatabaseArn" : "arn:aws:rds:us-east-1:123456789012:db:my-cdb-inst:pdb1",
        "DBInstanceIdentifier" : "my-cdb-inst",
        "TenantDBName" : "pdb1",
        "Status" : "modifying",
        "MasterUsername" : "tenant-admin-user"
        "Port" : "6555",
        "CharacterSetName" : "UTF-16",
        "MaxAllocatedStorage" : "1000",
        "ParameterGroups": [
            {
                "ParameterGroupName": "pdb1-params",
                "ParameterApplyStatus": "in-sync"
            }
        ],
        "OptionGroupMemberships": [
            {
                "OptionGroupName": "pdb1-options",
                "Status": "in-sync"
            }
        ],
        "PendingModifiedValues": {
            "TenantDBName": "pdb-hr"
        }
    }
```

```
}
```

Eliminazione di un database del tenant RDS per Oracle dal CDB

È possibile eliminare un database del tenant (PDB) utilizzando la AWS Management Console, la AWS CLI o l'API RDS. Considera i seguenti prerequisiti e limitazioni:

- Il database del tenant e l'istanza database devono esistere.
- Affinché l'eliminazione abbia esito positivo, è necessario che si verifichi una delle seguenti situazioni:
  - Il database del tenant e l'istanza database sono disponibili.

    > **Note**
    >
    > È possibile acquisire uno snapshot finale, ma solo se il database del tenant e l'istanza database erano disponibili prima dell'emissione del comando `delete-tenant-database`.

  - Il database del tenant è in fase di creazione.
  - L'istanza database sta modificando il database del tenant.
- Non è possibile eliminare più database del tenant in un'unica operazione.
- Non è possibile eliminare un database del tenant se è l'unico tenant nel CDB.

Console

Per eliminare un database del tenant

1. Accedi alla AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel pannello di navigazione, scegli Database, quindi scegli il database del tenant che vuoi eliminare.
3. In Actions (Azioni), selezionare Delete (Elimina).
4. Per creare uno snapshot DB finale per l’istanza database, abilitare Create final snapshot? (Crea snapshot finale?).
5. Se si è scelto di creare uno snapshot finale, immettere il Final snapshot name (Nome dello snapshot finale).

6. Immettere **`delete me`** nella casella.
7. Scegliere Delete (Elimina).

AWS CLI

Per eliminare un database tenant utilizzando ilAWS CLI, chiamate il delete-tenant-databasecomando con i seguenti parametri:

- `--db-instance-identifier` *`value`*
- `--tenant-db-name` *`value`*
- `[--skip-final-snapshot | --no-skip-final-snapshot]`
- `[--final-snapshot-identifier` *`value`*`]`

L'esempio seguente elimina il database tenant denominato *`pdb-test`* dal CDB denominato. *`my-cdb-inst`* Per impostazione predefinita, l'operazione crea uno snapshot finale.

Example

Per, o: Linux macOS Unix

```
aws rds delete-tenant-database --region us-east-1 \
    --db-instance-identifier my-cdb-inst \
    --tenant-db-name pdb-test \
    --final-snapshot-identifier final-snap-pdb-test
```

Per Windows:

```
aws rds delete-tenant-database --region us-east-1 ^
    --db-instance-identifier my-cdb-inst ^
    --tenant-db-name pdb-test ^
    --final-snapshot-identifier final-snap-pdb-test
```

Questo comando genera un output simile al seguente.

```
{
    "TenantDatabase" : {
        "DbiResourceId" : "db-abc123",
        "TenantDatabaseResourceId" : "tdb-bac456",
```

```
        "TenantDatabaseArn" : "arn:aws:rds:us-east-1:123456789012:db:my-cdb-inst:pdb-
test",
        "DBInstanceIdentifier" : "my-cdb-inst",
        "TenantDBName" : "pdb-test",
        "Status" : "deleting",
        "MasterUsername" : "pdb-test-admin"
        "Port" : "6555",
        "CharacterSetName" : "UTF-16",
        "MaxAllocatedStorage" : "1000",
        "ParameterGroups": [
            {
                "ParameterGroupName": "tenant-1-params",
                "ParameterApplyStatus": "in-sync"
            }
        ],
        "OptionGroupMemberships": [
            {
                "OptionGroupName": "tenant-1-options",
                "Status": "in-sync"
            }
        ]
    }
}
```

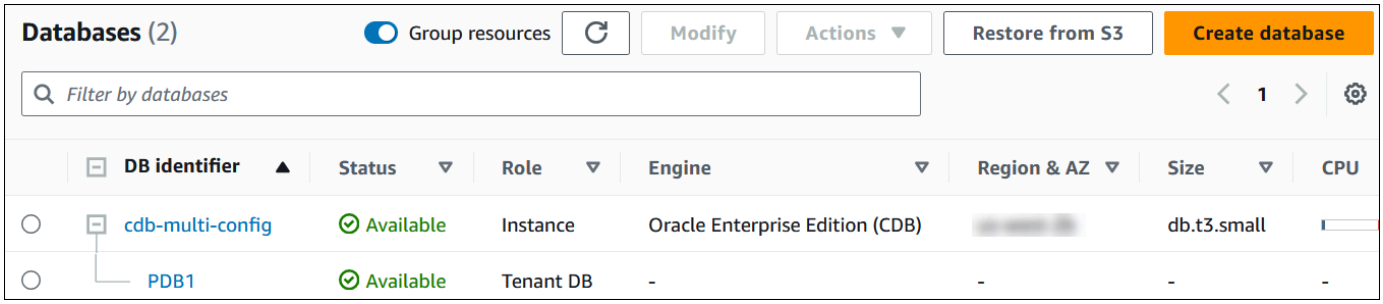
Visualizzazione dei dettagli del database del tenant

È possibile visualizzare i dettagli su un database del tenant nello stesso modo in cui è possibile visualizzare i dettagli su un database non CDB o CDB.

Console

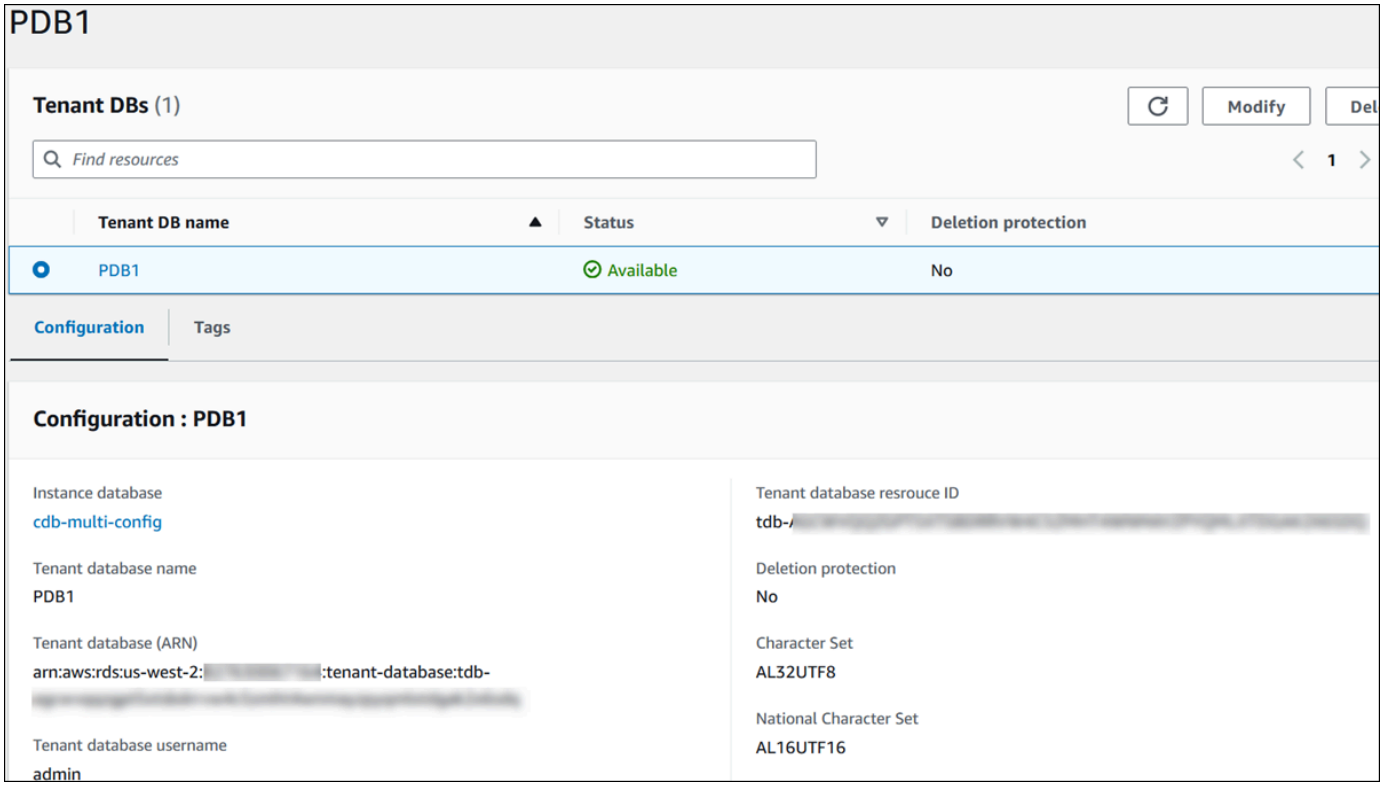
Per visualizzare i dettagli su un database del tenant

1. Accedi alla AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nell'angolo in alto a destra della console Amazon RDS, scegliere la Regione AWS in cui si trova l'istanza database.
3. Nel riquadro di navigazione, scegli Databases (Database).



Nell'immagine precedente, l'unico database del tenant (PDB) appare come figlio dell'istanza database.

4. Scegli il nome di un database del tenant.



AWS CLI

Per visualizzare i dettagli sui tuoi PDB, usa il AWS CLI comando describe-tenant-databases.

L'esempio seguente descrive tutti i database del tenant nella regione specificata.

Example

Per LinuxmacOS, oUnix:

```
aws rds describe-tenant-databases --region us-east-1
```

Per Windows:

```
aws rds describe-tenant-databases --region us-east-1
```

Questo comando genera un output simile al seguente.

```
    "TenantDatabases" : [
         {
            "DBInstanceIdentifier" : "my-cdb-inst",
            "TenantDBName" : "pdb-test",
            "Status" : "available",
            "MasterUsername" : "pdb-test-admin",
            "DbiResourceId" : "db-abc123",
            "TenantDatabaseResourceId" : "tdb-bac456",
            "TenantDatabaseArn" : "arn:aws:rds:us-east-1:123456789012:db:my-cdb-
inst:pdb-test",
            "CharacterSetName": "AL32UTF8",
            "NcharCharacterSetName": "AL16UTF16",
            "DeletionProtection": false,
            "PendingModifiedValues": {
                 "MasterUserPassword": "****"
            },
            "TagList": []
         },
         {

            "DBInstanceIdentifier" : "my-cdb-inst2",
            "TenantDBName" : "pdb-dev",
            "Status" : "modifying",
            "MasterUsername" : "masterrdsuser"
            "DbiResourceId" : "db-xyz789",
            "TenantDatabaseResourceId" : "tdb-ghp890",
            "TenantDatabaseArn" : "arn:aws:rds:us-east-1:123456789012:db:my-cdb-
inst2:pdb-dev",
            "CharacterSetName": "AL32UTF8",
            "NcharCharacterSetName": "AL16UTF16",
            "DeletionProtection": false,
            "PendingModifiedValues": {
                 "MasterUserPassword": "****"
            },
```

```
            "TagList": []
        },
        ... other truncated data
```

L'esempio seguente descrive i database del tenant sull'istanza database `my-cdb-inst` nella regione specificata.

Example

Per LinuxmacOS, oUnix:

```
aws rds describe-tenant-databases --region us-east-1 \
    --db-instance-identifier my-cdb-inst
```

Per Windows:

```
aws rds describe-tenant-databases --region us-east-1 ^
    --db-instance-identifier my-cdb-inst
```

Questo comando genera un output simile al seguente.

```
{
    "TenantDatabase": {
        "TenantDatabaseCreateTime": "2023-10-19T23:55:30.046Z",
        "DBInstanceIdentifier": "my-cdb-inst",
        "TenantDBName": "pdb-hr",
        "Status": "creating",
        "MasterUsername": "tenant-admin-user",
        "DbiResourceId": "db-abc123",
        "TenantDatabaseResourceId": "tdb-bac567",
        "TenantDatabaseARN": "arn:aws:rds:us-west-2:579508833180:pdb-hr:tdb-
abcdefghi1jklmno2p3qrst4uvw5xy6zabc7defghi8jklmn90op",
        "CharacterSetName": "AL32UTF8",
        "NcharCharacterSetName": "AL16UTF16",
        "DeletionProtection": false,
        "PendingModifiedValues": {
            "MasterUserPassword": "****"
        },
        "TagList": [
            {
                "Key": "TEST",
                "Value": "testValue"
```

```
            }
        ]
    }
}
```

L'esempio seguente descrive il database del tenant pdb1 su un'istanza database my-cdb-inst nella Regione Stati Uniti orientali (Virginia settentrionale).

Example

Per LinuxmacOS, oUnix:

```
aws rds describe-tenant-databases --region us-east-1 \
--db-instance-identifier my-cdb-inst \
--tenant-db-name pdb1
```

Per Windows:

```
aws rds describe-tenant-databases --region us-east-1 ^
--db-instance-identifier my-cdb-inst ^
--tenant-db-name pdb1
```

Questo comando genera un output simile al seguente.

```
{
    "TenantDatabases" : [
        {
            "DbiResourceId" : "db-abc123",
            "TenantDatabaseResourceId" : "tdb-bac567",
            "TenantDatabaseArn" : "arn:aws:rds:us-east-1:123456789012:db:my-cdb-
inst:pdb1"
            "DBInstanceIdentifier" : "my-cdb-inst",
            "TenantDBName" : "pdb1",
            "Status" : "ACTIVE",
            "MasterUsername" : "masterawsuser"
            "Port" : "1234",
            "CharacterSetName": "UTF-8",
            "ParameterGroups": [
                {
                    "ParameterGroupName": "tenant-custom-pg",
                    "ParameterApplyStatus": "in-sync"
                }
```

```
            ],
            {
            "OptionGroupMemberships": [
                {
                    "OptionGroupName": "tenant-custom-og",
                    "Status": "in-sync"
                }
            ]
        }
    ]
}
```

Aggiornamento del CDB

È possibile aggiornare un CDB a una versione diversa di Oracle Database. Ad esempio, è possibile aggiornare un database CDB da Oracle Database 19c a Oracle Database 21c. Non è possibile modificare l'architettura del database durante un aggiornamento. Pertanto, non è possibile aggiornare un database non CDB a un database CDB oppure aggiornare un database CDB a un database non CDB.

La procedura per aggiornare un database CDB a un database CDB è la stessa valida per l'aggiornamento da un database CDB non CDB a un database non CDB. Per ulteriori informazioni, consulta Aggiornamento del motore di database RDS per Oracle.

Amministrazione dell'istanza database RDS per Oracle

Di seguito sono riportate le attività di gestione comuni che si eseguono con una istanza database RDS per Oracle. Alcune attività sono uguali per tutte le istanze database RDS. Altri invece sono specifiche di RDS for Oracle.

Le attività riportate di seguito sono comuni a tutti i database RDS, ma per Oracle Database è necessario effettuare considerazioni particolari. Ad esempio, ci si connette a un database Oracle utilizzando i client Oracle SQL*Plus e SQL Developer.

Area attività	Documentazione di riferimento
Classi delle istanze, storage e PIOPS Se si sta creando un'istanza di produzione, occorre conoscere il funzionamento di classi di istanza, tipi di storage e IOPS con provisioning in Amazon RDS.	Classi di istanza RDS for Oracle Tipi di storage Amazon RDS
Implementazioni Multi-AZ Un'istanza database in produzione deve utilizzare implement azioni Multi-AZ. Le implementazioni Multi-AZ forniscono alle istanze database maggior disponibilità, longevità dei dati e tolleranza ai guasti.	Configurazione e gestione di un'implementazione multi-AZ
Amazon VPC Se l'account AWS ha un cloud privato virtuale (VPC) predefini to, l'istanza database viene creata automaticamente all'inter no di tale VPC. Se l'account non dispone di un VPC predefinito e desideri che l'istanza database sia in un VPC, è necessario creare il VPC e i gruppi di sottoreti prima di creare l'istanza.	Uso di un'istanza database in un VPC
Gruppi di sicurezza Per impostazione predefinita, le istanze database utilizzano un firewall che impedisce l'accesso. Per accedere all'istanza database, assicurati di aver creato un gruppo di sicurezza con gli indirizzi IP e la configurazione di rete corretti.	Controllo dell'accesso con i gruppi di sicurezza

Area attività	Documentazione di riferimento
Gruppi di parametri Se l'istanza database richiede parametri database specifici, crea un gruppo di parametri prima di creare l'istanza database.	Utilizzo di gruppi di parametri
Gruppi di opzioni Se l'istanza database richiede opzioni database specifiche, crea un gruppo di opzioni prima di creare l'istanza database.	Aggiunta di opzioni alle istanze database Oracle
Connessione all'istanza database Dopo aver creato un gruppo di sicurezza e averlo associato a un'istanza database, è possibile effettuare la connessione all'istanza database mediante un'applicazione cliente SQL standard, come Oracle SQL Plus.	Connessione all'istanza database RDS per Oracle
Backup e ripristino È possibile configurare l'istanza database affinché effettui backup automatici o acquisisca snapshot manuali e poi esegua il ripristino istanze da backup o snapshot.	Backup, ripristino ed esportazione dei dati
Monitoraggio Puoi monitorare un'istanza DB Oracle utilizzando i parametri, gli eventi e il monitoraggio avanzato di CloudWatch Amazon RDS.	Visualizzazione dei parametri nella console Amazon RDS Visualizzazione di eventi Amazon RDS
File di log È possibile accedere ai file di log per le istanze database Oracle.	Monitoraggio dei file di log di Amazon RDS

Di seguito, è riportata una descrizione per implementazioni specifiche di Amazon RDS di attività DBA comuni per RDS Oracle. Per offrire un'esperienza di servizio gestito, Amazon RDS non fornisce accesso shell alle istanze database. RDS limita anche l'accesso ad alcune procedure di sistema

e tabelle che richiedono privilegi avanzati. In molte delle attività, è possibile eseguire il pacchetto `rdsadmin`, che è uno strumento specifico di Amazon RDS che consente di amministrare il database.

Seguono alcune attività DBA comuni per le istanze database che eseguono Oracle:

- Attività di sistema

Disconnessione di una sessione	Metodo Amazon RDS: `rdsadmin.rdsadmin_util.disconnect` Metodo Oracle: `alter system disconnect session`
Terminazione di una sessione	Metodo Amazon RDS: `rdsadmin.rdsadmin_util.kill` Metodo Oracle: `alter system kill session`
Annullamento di una istruzione SQL in una sessione	Metodo Amazon RDS: `rdsadmin.rdsadmin_util.cancel` Metodo Oracle: `alter system cancel sql`
Abilitazione e disabilitazione delle sessioni limitate	Metodo Amazon RDS: `rdsadmin.rdsadmin_util.restricted_session` Metodo Oracle: `alter system enable restricted session`
Scaricamento del pool condiviso	Metodo Amazon RDS: `rdsadmin.rdsadmin_util.flush_shared_pool` Metodo Oracle: `alter system flush shared_pool`
Scaricamento della cache del buffer	Metodo Amazon RDS: `rdsadmin.rdsadmin_util.flush_buffer_cache` Metodo Oracle: `alter system flush buffer_cache`
Concedere privilegi SELECT o EXECUTE agli oggetti SYS	Metodo Amazon RDS: `rdsadmin.rdsadmin_util.grant_sys_object` Metodo Oracle: `grant`

Revoca del privilegio SELECT o EXECUTE in oggetti SYS	Metodo Amazon RDS: `rdsadmin.rdsadmin_util.revoke_sys_object` Metodo Oracle: `revoke`
Gestione delle viste RDS_X$ per le istanze di Oracle DB	Metodo Amazon RDS: `rdsadmin.rdsadmin_util.create_sys_x$_view` Metodo Oracle: `CREATE VIEW`
Concessione di privilegi a utenti non-master	Metodo Amazon RDS: `grant`
Creazione delle funzionalità personalizzate per verificare le password	Metodo Amazon RDS: `rdsadmin.rdsadmin_password_verify.create_verify_function` Metodo Amazon RDS: `rdsadmin.rdsadmin_password_verify.create_passthrough_verify_fcn`
Impostazione di un server DNS personalizzato	—
Elenco degli eventi diagnostici di sistema consentiti	Metodo Amazon RDS: `rdsadmin.rdsadmin_util.list_allowed_system_events` Metodo Oracle: —
Impostazione degli eventi di diagnostica del sistema	Metodo Amazon RDS: `rdsadmin.rdsadmin_util.set_allowed_system_events` Metodo Oracle: `ALTER SYSTEM SET EVENTS 'set_event_clause'`

Elenco degli eventi diagnostici di sistema impostati	Metodo Amazon RDS: `rdsadmin.rdsadmin_util.list_set_system_events` Metodo Oracle: `ALTER SESSION SET EVENTS 'IMMEDIATE EVENTDUMP(SYSTEM)'`
Annullamento dell'impostazione degli eventi diagnostici del sistema	Metodo Amazon RDS: `rdsadmin.rdsadmin_util.unset_system_event` Metodo Oracle: `ALTER SYSTEM SET EVENTS 'unset_event_clause'`

- Attività di database

Modifica del nome globale di un database	Metodo Amazon RDS: `rdsadmin.rdsadmin_util.rename_global_name` Metodo Oracle: `alter database rename`
Creazione e dimensionamento di spazi tabelle	Metodo Amazon RDS: `create tablespace` Metodo Oracle: `alter database`
Impostazione dello spazio di tabella predefinito	Metodo Amazon RDS: `rdsadmin.rdsadmin_util.alter_default_tablespace` Metodo Oracle: `alter database default tablespace`
Impostazione dello spazio di tabella temporaneo predefinito	Metodo Amazon RDS: `rdsadmin.rdsadmin_util.alter_default_temp_tablespace` Metodo Oracle: `alter database default temporary tablespace`
Creazione di un spazio di tabella temporaneo nell'archivio dell'istanza	Metodo Amazon RDS: `rdsadmin.rdsadmin_util.create_inst_store_tmp_tblspace` Metodo Oracle: `create temporary tablespace`

Checkpoint di un database	Metodo Amazon RDS: `rdsadmin.rdsadmin_util.checkpoint` Metodo Oracle: `alter system checkpoint`
Impostazione del ripristino distribuito	Metodo Amazon RDS: `rdsadmin.rdsadmin_util.enable_distr_recovery` Metodo Oracle: `alter system enable distributed recovery`
Impostazione del fuso orario del database	Metodo Amazon RDS: `rdsadmin.rdsadmin_util.alter_db_time_zone` Metodo Oracle: `alter database set time_zone`
Lavorare con le tabelle esterne Oracle	—
Generazione di report sulle prestazioni con AWR (Automatic Workload Repository)	Metodo Amazon RDS: procedure `rdsadmin.rdsadmin_diagnostic_util` Metodo Oracle: pacchetto `dbms_workload_repository`
Modifica dei collegamenti di database per l'utilizzo con le istanze database in un VPC	—
Impostazione dell'edizione predefinita per un'istanza database	Metodo Amazon RDS: `rdsadmin.rdsadmin_util.alter_default_edition` Metodo Oracle: `alter database default edition`
Abilitazione dell'audit per la tabella SYS.AUD$	Metodo Amazon RDS: `rdsadmin.rdsadmin_master_util.audit_all_sys_aud_table` Metodo Oracle: `audit`

Disabilitazione dell'audit per la tabella SYS.AUD$	Metodo Amazon RDS: `rdsadmin.rdsadmin_master_util.noaudit_all_sys_aud_table` Metodo Oracle: `noaudit`
Pulizia di compilazioni dell'indice online interrotte	Metodo Amazon RDS: `rdsadmin.rdsadmin_dbms_repair.online_index_clean` Metodo Oracle: `dbms_repair.online_index_clean`
Ignorare blocchi corrotti	Metodo Amazon RDS: diverse procedure `rdsadmin.rdsadmin_dbms_repair` Metodo Oracle: pacchetto `dbms_repair`
Ridimensionamento di spazi di tabella, file di dati e file temporanei	Metodo Amazon RDS: procedura `rdsadmin.rdsadmin_util.resize_temp_tablespace` , `rdsadmin.rdsadmin_util.resize_tempfile` o `rdsadmin.rdsadmin_util.autoextend_tempfile` Procedura `rdsadmin.rdsadmin_util.resize_datafile` o `rdsadmin.rdsadmin_util.autoextend_datafile` Metodo Oracle: —
Eliminazione del cestino riciclaggio	Metodo Amazon RDS: `EXEC rdsadmin.rdsadmin_util.purge_dba_recyclebin` Metodo Oracle: `purge dba_recyclebin`
Impostazione dei valori di default visualizzati per la redazione completa	Metodo Amazon RDS: `EXEC rdsadmin.rdsadmin_util.dbms_redact_upd_full_rdct_val` Metodo Oracle: `exec dbms_redact.UPDATE_FULL_REDACTION_VALUES`

- Attività di log

Impostazione accesso forzato	Metodo Amazon RDS: `rdsadmin.rdsadmin_util.force_logging` Metodo Oracle: `alter database force logging`
Impostazione di accesso supplementare	Metodo Amazon RDS: `rdsadmin.rdsadmin_util.alter_supplemental_logging` Metodo Oracle: `alter database add supplemental log`
Cambio dei file di log online	Metodo Amazon RDS: `rdsadmin.rdsadmin_util.switch_logfile` Metodo Oracle: `alter system switch logfile`
Aggiunta di log redo online	Metodo Amazon RDS: `rdsadmin.rdsadmin_util.add_logfile`
Eliminazione di log redo online	Metodo Amazon RDS: `rdsadmin.rdsadmin_util.drop_logfile`
Ridimensionamento di log redo online	—
Conservazione dei log redo archiviati	Metodo Amazon RDS: `rdsadmin.rdsadmin_util.set_configuration`

Download dei log di ripristino archiviati da Simple Storage Service (Amazon S3)	Metodo Amazon RDS: `rdsadmin.rdsadmin_archive_log_download.download_log_with_seqnum` Metodo Amazon RDS: `rdsadmin.rdsadmin_archive_log_download.download_logs_in_seqnum_range`
Accesso ai log di ripristino online e archiviati	Metodo Amazon RDS: `rdsadmin.rdsadmin_master_util.create_archivelog_dir` Metodo Amazon RDS: `rdsadmin.rdsadmin_master_util.create_onlinelog_dir`

- Attività RMAN

Convalida dei file di database in RDS per Oracle	Metodo Amazon RDS: `rdsadmin_rman_util. procedure` Metodo Oracle: `RMAN VALIDATE`

Abilitazione e disabilitazione del monitoraggio delle modifiche dei blocchi.	Metodo Amazon RDS: `rdsadmin_rman_util . procedure` Metodo Oracle: `ALTER DATABASE`
Controllo incrociato dei log redo archiviati	Metodo Amazon RDS: `rdsadmin_rman_util .crosscheck_archivelog` Metodo Oracle: `RMAN BACKUP`
Backup dei redo log file archiviati	Metodo Amazon RDS: `rdsadmin_rman_util . procedure` Metodo Oracle: `RMAN BACKUP`
Esecuzione di un backup di database completo	Metodo Amazon RDS: `rdsadmin_rman_util .backup_database_full` Metodo Oracle: `RMAN BACKUP`
Esecuzione di un backup di database incrementale	Metodo Amazon RDS: `rdsadmin_rman_util .backup_database_incremental` Metodo Oracle: `RMAN BACKUP`

Backup di uno spazio di tabella	Metodo Amazon RDS: `rdsadmin_rman_util.backup_database_tablespace` Metodo Oracle: `RMAN BACKUP`

- Attività Oracle Scheduler

Modifica dei processi di DBMS_SCHEDULER	Metodo Amazon RDS: `dbms_scheduler.set_attribute` Metodo Oracle: `dbms_scheduler.set_attribute`
Modifica delle finestre di AutoTask manutenzione	Metodo Amazon RDS: `dbms_scheduler.set_attribute` Metodo Oracle: `dbms_scheduler.set_attribute`
Impostazione del fuso orario per i job di Oracle Scheduler	Metodo Amazon RDS: `dbms_scheduler.set_scheduler_attribute` Metodo Oracle: `dbms_scheduler.set_scheduler_attribute`

Disattivazione dei processi Oracle Scheduler di proprietà di SYS	Metodo Amazon RDS: `rdsadmin.rdsadmin_dbms_scheduler.disable` Metodo Oracle: `dbms_scheduler.disable`
Attivazione dei processi Oracle Scheduler di proprietà di SYS	Metodo Amazon RDS: `rdsadmin.rdsadmin_dbms_scheduler.enable` Metodo Oracle: `dbms_scheduler.enable`
Modifica dell'intervallo di ripetizione di Oracle Scheduler dei processi di tipo CALENDAR	Metodo Amazon RDS: `rdsadmin.rdsadmin_dbms_scheduler.set_attribute` Metodo Oracle: `dbms_scheduler.set_attribute`
Modifica dell'intervallo di ripetizione di Oracle Scheduler dei processi di tipo NAMED	Metodo Amazon RDS: `rdsadmin.rdsadmin_dbms_scheduler.set_attribute` Metodo Oracle: `dbms_scheduler.set_attribute`

Disattivazione del commit automatico per la creazione di processi in Oracle Scheduler	Metodo Amazon RDS: `rdsadmin.rdsadmin_dbms_scheduler.set_no_commit_flag` Metodo Oracle: `dbms_isched.set_no_commit_flag`

- Attività diagnostiche

Elenco degli incidenti	Metodo Amazon RDS: `rdsadmin.rdsadmin_adrci_util.list_adrci_incidents` Metodo Oracle: comando ADRCI `show incident`
Elenco dei problemi	Metodo Amazon RDS: `rdsadmin.rdsadmin_adrci_util.list_adrci_problem` Metodo Oracle: comando ADRCI `show problem`
Creazione di pacchetti incidenti	Metodo Amazon RDS: `rdsadmin.rdsadmin_adrci_util.create_adrci_package` Metodo Oracle: comando ADRCI `ips create package`

Visualizzazione di file di traccia	Metodo Amazon RDS: `rdsadmin.rdsadmin_adrci_util.show_adrci_tracefile` Metodo Oracle: comando ADRCI `show tracefile`

- Altre attività

Creazione ed eliminazione di directory nello spazio di archiviazione dati principale	Metodo Amazon RDS: `rdsadmin.rdsadmin_util.create_directory` Metodo Oracle: `CREATE DIRECTORY` Metodo Amazon RDS: `rdsadmin.rdsadmin_util.drop_directory` Metodo Oracle: `DROP DIRECTORY`
Generazione di un elenco dei file in una directory di istanze database	Metodo Amazon RDS: `rdsadmin.rds_file_util.listdir` Metodo Oracle: —
Lettura dei file in una directory di istanze database	Metodo Amazon RDS: `rdsadmin.rds_file_util.read_text_file` Metodo Oracle: —

Accesso ai file Opatch	Metodo Amazon RDS: `rdsadmin.rds_file_util.read_text_file` o `rdsadmin.tracefile_listing` Metodo Oracle: `opatch`
Impostazione dei parametri per le attività di advisor	Metodo Amazon RDS: `rdsadmin.rdsadmin_util.advisor_task_set_parameter` Metodo Oracle: varie procedure di pacchetti archiviati
Disattivazione di AUTO_STATS_ADVISOR_TASK	Metodo Amazon RDS: `rdsadmin.rdsadmin_util.advisor_task_drop` Metodo Oracle: —
Riattivazione di AUTO_STATS_ADVISOR_TASK	Metodo Amazon RDS: `rdsadmin.rdsadmin_util.dbms_stats_init` Metodo Oracle: —

Puoi anche utilizzare le procedure di Amazon RDS per l'integrazione di Amazon S3 con Oracle e per l'esecuzione delle attività di database di OEM Management Agent. Per ulteriori informazioni, consultare Integrazione Amazon S3 e Esecuzione delle attività di database con Management Agent.

Esecuzione di attività di sistema comuni per le istanze database Oracle

Di seguito, viene descritto come eseguire determinate attività DBA comuni relative al sistema nelle istanze database Amazon RDS che eseguono Oracle. Per fornire un'esperienza di servizio gestito, Amazon RDS non fornisce accesso shell alle istanze database e limita l'accesso a certe procedure e tabelle di sistema che richiedono privilegi avanzati.

Argomenti

Disconnessione di una sessione

Puoi usare la procedura in Amazon RDS per disconnettere la sessione corrente terminando il processo server dedicato `rdsadmin.rdsadmin_util.disconnect`. La procedura `disconnect` include i seguenti parametri.

Nome del parametro	Tipo di dati	Default	Campo obbligatorio	Descrizione
`sid`	numero	—	Sì	L'identificatore di sessione.
`serial`	numero	—	Sì	Il numero di serie della sessione.
`method`	varchar	“IMMEDIATO”	No	I valori validi sono `'IMMEDIATE'` e `'POST_TRANSACTION'`.

L'esempio seguente disconnette una sessione.

```
begin
    rdsadmin.rdsadmin_util.disconnect(
        sid    => sid,
        serial => serial_number);
end;
/
```

Per ottenere l'identificatore di sessione e il numero di serie di sessione, eseguire una query sulla visualizzazione `V$SESSION`. L'esempio seguente ottiene tutte le sessioni per l'utente `AWSUSER`.

```
SELECT SID, SERIAL#, STATUS FROM V$SESSION WHERE USERNAME = 'AWSUSER';
```

Il database deve essere aperto per utilizzare questo metodo. Per ulteriori informazioni sulla disconnessione di una sessione, consulta ALTER SYSTEM nella documentazione di Oracle.

Terminazione di una sessione

Per terminare una sessione, utilizzare la procedura Amazon RDS `rdsadmin.rdsadmin_util.kill`. La procedura `kill` include i seguenti parametri.

Nome del parametro	Tipo di dati	Default	Campo obbligatorio	Descrizione
`sid`	numero	—	Sì	L'identificatore di sessione.
`serial`	numero	—	Sì	Il numero di serie della sessione.
`method`	varchar	null	No	I valori validi sono `'IMMEDIATE'` e `'PROCESS'`. Se specifichi `IMMEDIATE`, ottieni lo stesso effetto dell'istruzione riportata di seguito: `ALTER SYSTEM KILL SESSION 'sid,serial#' IMMEDIATE` Se specifichi `PROCESS`, vengono terminati i processi associati a una sessione. Specifica `PROCESS` solo se la terminazione della sessione con `IMMEDIATE` non riesce.

Per ottenere l'identificatore di sessione e il numero di serie di sessione, eseguire una query sulla visualizzazione `V$SESSION`. L'esempio seguente ottiene tutte le sessioni per l'utente *`AWSUSER`*.

```
SELECT SID, SERIAL#, STATUS FROM V$SESSION WHERE USERNAME = 'AWSUSER';
```

Nell'esempio seguente viene terminata una sessione.

```
BEGIN
    rdsadmin.rdsadmin_util.kill(
        sid    => sid,
        serial => serial_number,
        method => 'IMMEDIATE');
END;
/
```

Nell'esempio seguente vengono terminati i processi associati a una sessione.

```
BEGIN
    rdsadmin.rdsadmin_util.kill(
        sid    => sid,
        serial => serial_number,
        method => 'PROCESS');
END;
/
```

Annullamento di una istruzione SQL in una sessione

Per annullare un'istruzione SQL in una sessione, puoi utilizzare la procedura in Amazon RDS `rdsadmin.rdsadmin_util.cancel`.

> **Note**
>
> Questa procedura è supportata per Oracle Database 19c (19.0.0) e per tutte le versioni principali e secondarie di RDS for Oracle.

La procedura `cancel` include i seguenti parametri.

Nome del parametro	Tipo di dati	Default	Campo obbligatorio	Descrizione
`sid`	numero	—	Sì	L'identificatore di sessione.

Nome del parametro	Tipo di dati	Default	Campo obbligatorio	Descrizione
`serial`	numero	—	Sì	Il numero di serie della sessione.
`sql_id`	varchar2	null	No	L'identificatore SQL nell'istruzione SQL.

L'esempio seguente annulla un'istruzione SQL in una sessione.

```
begin
    rdsadmin.rdsadmin_util.cancel(
        sid    => sid,
        serial => serial_number,
        sql_id => sql_id);
end;
/
```

Per ottenere l'identificatore di sessione, il numero di serie di sessione e l'identificativo SQL di un'istruzione SQL, eseguire una query sulla visualizzazione `V$SESSION`. L'esempio seguente ottiene tutte le sessioni e gli identificativi SQL per l'utente `AWSUSER`.

```
select SID, SERIAL#, SQL_ID, STATUS from V$SESSION where USERNAME = 'AWSUSER';
```

Abilitazione e disabilitazione delle sessioni limitate

Puoi usare la procedura in Amazon RDS per abilitare e disabilitare sessioni limitate `rdsadmin.rdsadmin_util.restricted_session`. La procedura `restricted_session` include i seguenti parametri.

Nome del parametro	Tipo di dati	Default	Sì	Descrizione
`p_enable`	booleano	true	No	Impostato su `true` per abilitare le sessioni limitate, su `false` per

Nome del parametro	Tipo di dati	Default	Sì	Descrizione
				disabilitare le sessioni limitate.

L'esempio seguente mostra come abilitare e disabilitare le sessioni limitate.

```
/* Verify that the database is currently unrestricted. */

SELECT LOGINS FROM V$INSTANCE;

LOGINS
-------
ALLOWED

/* Enable restricted sessions */

EXEC rdsadmin.rdsadmin_util.restricted_session(p_enable => true);


/* Verify that the database is now restricted. */

SELECT LOGINS FROM V$INSTANCE;

LOGINS
----------
RESTRICTED


/* Disable restricted sessions */

EXEC rdsadmin.rdsadmin_util.restricted_session(p_enable => false);


/* Verify that the database is now unrestricted again. */

SELECT LOGINS FROM V$INSTANCE;

LOGINS
-------
ALLOWED
```

Scaricamento del pool condiviso

Puoi usare la procedura in Amazon RDS per scaricare il pool condiviso `rdsadmin.rdsadmin_util.flush_shared_pool`. La procedura `flush_shared_pool` non ha parametri.

L'esempio seguente scarica il pool condiviso.

```
EXEC rdsadmin.rdsadmin_util.flush_shared_pool;
```

Scaricamento della cache del buffer

Puoi usare la procedura in Amazon RDS per scaricare la cache del buffe `rdsadmin.rdsadmin_util.flush_buffer_cache`. La procedura `flush_buffer_cache` non ha parametri.

L'esempio seguente scarica la cache del buffer.

```
EXEC rdsadmin.rdsadmin_util.flush_buffer_cache;
```

Scaricamento della cache smart flash del database

Per scaricare la cache smart flash del database, utilizza la procedura Amazon RDS `rdsadmin.rdsadmin_util.flush_flash_cache`. La procedura `flush_flash_cache` non ha parametri. Nell'esempio seguente viene scaricata la cache smart flash del database.

```
EXEC rdsadmin.rdsadmin_util.flush_flash_cache;
```

Per ulteriori informazioni sull'utilizzo della cache smart flash del database con RDS per Oracle, consulta Archiviazione di dati temporanei in un archivio dell'istanza RDS per Oracle.

Concedere privilegi SELECT o EXECUTE agli oggetti SYS

Solitamente si trasferiscono i privilegi utilizzando ruoli che possono contenere molti oggetti. Puoi concedere privilegi a un singolo oggetto utilizzando la procedura in Amazon RDS `rdsadmin.rdsadmin_util.grant_sys_object`. La procedura concede solo i privilegi già concessi all'utente master tramite un ruolo o una concessione diretta.

La procedura `grant_sys_object` include i seguenti parametri.

> **Important**
>
> Per tutti i valori dei parametri, utilizzare maiuscole a meno che non sia stato creato l'utente con un identificatore con distinzione tra maiuscole e minuscole. Ad esempio, se esegui `CREATE USER myuser` o `CREATE USER MYUSER`, il dizionario dati memorizza `MYUSER`. Tuttavia, se si utilizzano virgolette doppie in `CREATE USER "MyUser"`, il dizionario dati memorizza `MyUser`.

Nome del parametro	Tipo di dati	Default	Campo obbligatorio	Descrizione
`p_obj_name`	varchar2	—	Sì	Il nome dell'oggetto per il quale concedere privilegi . L'oggetto può essere una directory, funzione, pacchetto, procedura , sequenza, tabella o visualizzazione. I nomi degli oggetti devono essere scritti allo stesso modo in cui appaiono in `DBA_OBJECTS` . La maggior parte degli oggetti di sistema è scritta in maiuscolo, quindi consigliamo di provare prima il maiuscolo.
`p_grantee`	varchar2	—	Sì	Il nome dell'oggetto al quale concedere privilegi. L'oggetto può essere uno schema o un ruolo.
`p_privilege`	varchar2	null	Sì	—

Nome del parametro	Tipo di dati	Default	Campo obbligatorio	Descrizione
p_grant_option	booleano	false	No	Impostato su true per utilizzare l'opzione concessione. Il parametro p_grant_option è supportato per 12.1.0.2.v4 e versioni successive, tutte le versioni 12.2.0.1 e tutte le versioni 19.0.0.

L'esempio seguente concede certi privilegi su un oggetto denominato V_$SESSION a un utente denominato USER1.

```
begin
    rdsadmin.rdsadmin_util.grant_sys_object(
        p_obj_name  => 'V_$SESSION',
        p_grantee   => 'USER1',
        p_privilege => 'SELECT');
end;
/
```

L'esempio seguente concede certi privilegi su un oggetto denominato V_$SESSION a un utente denominato USER1 con l'opzione concessione.

```
begin
    rdsadmin.rdsadmin_util.grant_sys_object(
        p_obj_name     => 'V_$SESSION',
        p_grantee      => 'USER1',
        p_privilege    => 'SELECT',
        p_grant_option => true);
end;
/
```

Per poter concedere privilegi per un oggetto, quei privilegi devono essere concessi all'account direttamente con l'opzione concessione o tramite un ruolo concesso utilizzando with admin

option. Nel caso più comune, potresti voler concedere SELECT a una visualizzazione DBA concessa al ruolo SELECT_CATALOG_ROLE. Se quel ruolo non è già concesso direttamente all'utente utilizzando with admin option, non sarai in grado di trasferire il privilegio. Se disponi del privilegio DBA, puoi concedere il ruolo direttamente a un altro utente.

L'esempio seguente concede SELECT_CATALOG_ROLE e EXECUTE_CATALOG_ROLE a USER1. Siccome with admin option viene utilizzato, USER1 può ora garantire l'accesso agli oggetti SYS che sono stati concessi a SELECT_CATALOG_ROLE.

```
GRANT SELECT_CATALOG_ROLE TO USER1 WITH ADMIN OPTION;
GRANT EXECUTE_CATALOG_ROLE to USER1 WITH ADMIN OPTION;
```

Gli oggetti già concessi a PUBLIC non devono essere concessi nuovamente. Se utilizzi la procedura grant_sys_object per concedere nuovamente l'accesso, la chiamata di procedura va a buon fine.

Revoca del privilegio SELECT o EXECUTE in oggetti SYS

Puoi revocare privilegi in un singolo oggetto usando la procedura in Amazon RDS rdsadmin.rdsadmin_util.revoke_sys_object. La procedura revoca solo i privilegi che l'account master è già stato concesso tramite un ruolo o una concessione diretta.

La procedura revoke_sys_object include i seguenti parametri.

Nome del parametro	Tipo di dati	Default	Campo obbligatorio	Descrizione
p_obj_name	varchar2	—	Sì	Il nome dell'oggetto per il quale revocare privilegi . L'oggetto può essere una directory, funzione, pacchetto, procedura , sequenza, tabella o visualizzazione. I nomi degli oggetti devono essere scritti allo stesso modo in cui appaiono in DBA_OBJECTS . LA maggior parte degli

Nome del parametro	Tipo di dati	Default	Campo obbligatorio	Descrizione
				oggetti di sistema sono scritti in maiuscolo, quindi consigliamo di provare prima il maiuscolo.
p_revokee	varchar2	—	Sì	Il nome dell'oggetto per il quale revocare privilegi. L'oggetto può essere uno schema o un ruolo.
p_privilege	varchar2	null	Sì	—

L'esempio seguente revoca certi privilegi su un oggetto denominato V_$SESSION a un utente denominato USER1.

```
begin
    rdsadmin.rdsadmin_util.revoke_sys_object(
        p_obj_name  => 'V_$SESSION',
        p_revokee   => 'USER1',
        p_privilege => 'SELECT');
end;
/
```

Gestione delle viste RDS_X$ per le istanze di Oracle DB

Potrebbe essere necessario accedere alle tabelle SYS.X$ fisse, accessibili solo daSYS.
Per creare SYS.RDS_X$ viste su X$ tabelle idonee, utilizzate le procedure incluse nel rdsadmin.rdsadmin_util pacchetto. Al tuo utente principale viene automaticamente concesso il privilegio SELECT … WITH GRANT OPTION sulle RDS_X$ viste.

Le rdsadmin.rdsadmin_util procedure sono disponibili nelle seguenti versioni del motore di database:

- 21.0.0.0.ru-2023-10.rur-2023-10.r1e versioni successive di Oracle Database 21c
- 19.0.0.0.ru-2023-10.rur-2023-10.r1e versioni successive di Oracle Database 19c

> **Important**
>
> Internamente, il `rdsadmin.rdsadmin_util` pacchetto crea viste sulle X$ tabelle. Le X$ tabelle sono oggetti di sistema interni che non sono descritti nella documentazione di Oracle Database. Si consiglia di testare viste specifiche nel database non di produzione e di creare viste nel database di produzione solo sotto la guida di Oracle Support.

Elenca le tabelle fisse X$ idonee per l'uso nelle viste RDS_X$

Per elencare le tabelle X$ idonee all'uso nelle RDS_X$ viste, utilizza la procedura RDS. `rdsadmin.rdsadmin_util.list_allowed_sys_x$_views` Questa procedura non accetta parametri. Le seguenti istruzioni elencano tutte le X$ tabelle idonee (output di esempio incluso).

```
SQL> SET SERVEROUTPUT ON
SQL> SELECT * FROM TABLE(rdsadmin.rdsadmin_util.list_allowed_sys_x$_views);

'X$BH'
'X$K2GTE'
'X$KCBWBPD'
'X$KCBWDS'
'X$KGLLK'
'X$KGLOB'
'X$KGLPN'
'X$KSLHOT'
'X$KSMSP'
'X$KSPPCV'
'X$KSPPI'
'X$KSPPSV'
'X$KSQEQ'
'X$KSQRS'
'X$KTUXE'
'X$KQRFP'
```

L'elenco delle X$ tabelle idonee può cambiare nel tempo. Per assicurarti che l'elenco delle tabelle X$ fisse idonee sia aggiornato, esegui nuovamente periodicamente`list_allowed_sys_x$_views`.

Creazione di viste SYS.RDS_X$

Per creare una RDS_X$ vista su una X$ tabella idonea, utilizzare la procedura RDS. `rdsadmin.rdsadmin_util.create_sys_x$_view` È possibile creare viste solo per le tabelle

elencate nell'output dirdsadmin.rdsadmin_util.list_allowed_sys_x$_views. La procedura create_sys_x$_view include i seguenti parametri.

Nome del parametro	Tipo di dati	Default	Campo obbligatorio	Descrizione
p_x$_tbl	varchar2	Null	Sì	Un nome di X$ tabella valido. Il valore deve essere una delle X $ tabelle riportate dalist_allowed_sys_x$_views .
p_force_creation	Boolean	FALSE	No	Un valore che indica se forzare la creazione di una RDS_X$ vista già esistente per una X$ tabella. Per impostazione predefinita, RDS non crea una vista se esiste già. Per forzare la creazione, imposta questo parametro suTRUE.

L'esempio seguente crea la SYS.RDS_X$KGLOB vista sulla tabellaX$KGLOB. Il formato per il nome della vista èRDS_*X$tablename*.

```
SQL> SET SERVEROUTPUT ON
SQL> EXEC rdsadmin.rdsadmin_util.create_sys_x$_view('X$KGLOB');

PL/SQL procedure successfully completed.
```

La seguente query sul dizionario di dati elenca la vista SYS.RDS_X$KGLOB e ne mostra lo stato. Al tuo utente principale viene automaticamente concesso il privilegio SELECT ... WITH GRANT OPTION su questa visualizzazione.

```
SQL> SET SERVEROUTPUT ON
SQL> COL OWNER FORMAT A30
SQL> COL OBJECT_NAME FORMAT A30
SQL> COL STATUS FORMAT A30
SQL> SET LINESIZE 200
SQL> SELECT OWNER, OBJECT_NAME, STATUS
FROM DBA_OBJECTS
WHERE OWNER = 'SYS' AND OBJECT_NAME = 'RDS_X$KGLOB';

OWNER                          OBJECT_NAME                    STATUS
------------------------------ ------------------------------
 ------------------------------
SYS                            RDS_X$KGLOB                    VALID
```

> **Important**
>
> X$non è garantito che le tabelle rimangano invariate prima e dopo un aggiornamento. RDS per Oracle elimina e ricrea le RDS_X$ visualizzazioni sulle X$ tabelle durante un aggiornamento del motore. Quindi concede il `SELECT ... WITH GRANT OPTION` privilegio all'utente principale. Dopo un aggiornamento, concedi i privilegi agli utenti del database secondo necessità sulle viste corrispondenti. RDS_X$

Elenco delle visualizzazioni SYS.RDS_X$

Per elencare le RDS_X$ viste esistenti, utilizzare la procedura RDS. `rdsadmin.rdsadmin_util.list_created_sys_x$_views` La procedura elenca solo le viste create dalla procedura`create_sys_x$_view`. L'esempio seguente elenca le X$ tabelle con RDS_X$ viste corrispondenti (output di esempio incluso).

```
SQL> SET SERVEROUTPUT ON
SQL> COL XD_TBL_NAME FORMAT A30
SQL> COL STATUS FORMAT A30
SQL> SET LINESIZE 200
SQL> SELECT * FROM TABLE(rdsadmin.rdsadmin_util.list_created_sys_x$_views);

XD_TBL_NAME                    STATUS
------------------------------ ------------------------------
X$BH                           VALID
X$K2GTE                        VALID
X$KCBWBPD                      VALID
```

```
3 rows selected.
```

Eliminazione delle visualizzazioni RDS_X$

Per eliminare una SYS.RDS_X$ visualizzazione, utilizzare la procedura RDS. `rdsadmin.rdsadmin_util.drop_sys_x$_view` È possibile eliminare solo le viste elencate nell'output di`rdsadmin.rdsadmin_util.list_allowed_sys_x$_views`. La procedura `drop_sys_x$_view` accetta il seguente parametro.

Nome del parametro	Tipo di dati	Default	Campo obbligatorio	Descrizione
`p_x$_tbl`	varchar2	Null	Sì	Un nome di tabella X$ fisso valido. Il valore deve essere una delle tabelle X$ fisse riportate da`list_created_sys_x$_views` .

L'esempio seguente elimina la RDS_X$KGLOB vista creata nella tabellaX$KGLOB.

```
SQL> SET SERVEROUTPUT ON
SQL> EXEC rdsadmin.rdsadmin_util.drop_sys_x$_view('X$KGLOB');

PL/SQL procedure successfully completed.
```

L'esempio seguente mostra che la vista SYS.RDS_X$KGLOB è stata eliminata (incluso l'output di esempio).

```
SQL> SET SERVEROUTPUT ON
SQL> COL OWNER FORMAT A30
SQL> COL OBJECT_NAME FORMAT A30
SQL> COL STATUS FORMAT A30
SQL> SET LINESIZE 200
SQL> SELECT OWNER, OBJECT_NAME, STATUS
FROM DBA_OBJECTS
```

```
WHERE OWNER = 'SYS' AND OBJECT_NAME = 'RDS_X$KGLOB';

no rows selected
```

Concessione di privilegi a utenti non-master

È possibile concedere privilegi per molti oggetti nello schema SYS utilizzando il ruolo SELECT_CATALOG_ROLE. Il ruolo SELECT_CATALOG_ROLE offre agli utenti privilegi SELECT per visualizzazioni del dizionario dati. L'esempio seguente concede il ruolo SELECT_CATALOG_ROLE a un utente denominato user1.

```
GRANT SELECT_CATALOG_ROLE TO user1;
```

È possibile concedere privilegi EXECUTE per molti oggetti nello schema SYS utilizzando il ruolo EXECUTE_CATALOG_ROLE. Il ruolo EXECUTE_CATALOG_ROLE offre agli utenti privilegi EXECUTE per pacchetti e procedure nel dizionario dati. L'esempio seguente concede il ruolo EXECUTE_CATALOG_ROLE a un utente denominato user1.

```
GRANT EXECUTE_CATALOG_ROLE TO user1;
```

L'esempio seguente ottiene le autorizzazioni che permettono i ruoli SELECT_CATALOG_ROLE e EXECUTE_CATALOG_ROLE.

```
  SELECT *
    FROM ROLE_TAB_PRIVS
   WHERE ROLE IN ('SELECT_CATALOG_ROLE','EXECUTE_CATALOG_ROLE')
ORDER BY ROLE, TABLE_NAME ASC;
```

L'esempio seguente crea un utente non-master denominato user1, concede il privilegio CREATE SESSION e il privilegio SELECT in un database denominato sh.sales.

```
CREATE USER user1 IDENTIFIED BY PASSWORD;
GRANT CREATE SESSION TO user1;
GRANT SELECT ON sh.sales TO user1;
```

Creazione delle funzionalità personalizzate per verificare le password

Puoi creare una funzionalità di verifica della password personalizzata in due modi:

- Se desideri utilizzare la verifica standard e archiviare la funzione nello schema SYS, utilizza la procedura `create_verify_function`.
- Se desideri utilizzare la verifica personalizzata p desideri archiviare la funzione nello schema SYS, utilizza la procedura `create_passthrough_verify_fcn`.

La procedura create_verify_function

Puoi creare una funzione personalizzata per verificare le password usando la procedura in Amazon RDS `rdsadmin.rdsadmin_password_verify.create_verify_function`. La procedura `create_verify_function` è supportata per RDS for Oracle versione 12.1.0.2.v5 e tutte le successive versioni principali e secondarie.

La procedura `create_verify_function` include i seguenti parametri.

Nome del parametro	Tipo di dati	Default	Campo obbligatorio	Descrizione
`p_verify_function_name`	varchar2	—	Sì	Il nome per la funzionalità personalizzata. La funzionalità viene creata per te nello schema SYS. Assegni questa funzione a profili di utente.
`p_min_length`	numero	8	No	Il numero minimo di caratteri necessari.
`p_max_length`	numero	256	No	Il numero massimo di caratteri permessi.
`p_min_letters`	numero	1	No	Il numero minimo di lettere necessarie.
`p_min_uppercase`	numero	0	No	Il numero minimo di lettere maiuscole necessarie.

Nome del parametro	Tipo di dati	Default	Campo obbligatorio	Descrizione
p_min_lowercase	numero	0	No	Il numero minimo di lettere minuscole necessarie.
p_min_digits	numero	1	No	Il numero minimo di cifre necessarie.
p_min_special	numero	0	No	Il numero minimo di caratteri speciali necessari.
p_min_different_chars	numero	3	No	Il numero minimo di caratteri diversi necessari tra la password vecchia e quella nuova.
p_disallow_username	booleano	true	No	Impostato su `true` per non consentire il nome utente nella password.
p_disallow_reverse	booleano	true	No	Impostato su `true` per non consentire l'inversione del nome utente nella password.
p_disallow_db_name	booleano	true	No	Impostato su `true` per non consentire il database o il nome del server nella password.
p_disallow_simple_strings	booleano	true	No	Impostato su `true` per non consentire stringhe semplici come password.

Nome del parametro	Tipo di dati	Default	Campo obbligatorio	Descrizione
p_disallow_whitespace	booleano	false	No	Impostato su true per non consentire gli spazi vuoti nella password.
p_disallow_at_sign	booleano	false	No	Impostare su true per non consentire il carattere @ nella password.

Puoi creare funzionalità multiple di verifica della password.

Ci sono limitazioni riguardo al nome della funzionalità personalizzata. La funzione personalizzata non può avere lo stesso nome di un oggetto di sistema esistente. La lunghezza del nome non può superare i 30 caratteri. Inoltre, il nome deve includere una delle seguenti stringhe: PASSWORD, VERIFY, COMPLEXITY, ENFORCE o STRENGTH.

L'esempio seguente crea una funzionalità denominata CUSTOM_PASSWORD_FUNCTION. La funzionalità richiede una password che includa almeno 12 caratteri, 2 caratteri maiuscoli, 1 cifra, 1 carattere speciale e che non consenta il carattere @.

```
begin
    rdsadmin.rdsadmin_password_verify.create_verify_function(
        p_verify_function_name => 'CUSTOM_PASSWORD_FUNCTION',
        p_min_length           => 12,
        p_min_uppercase        => 2,
        p_min_digits           => 1,
        p_min_special          => 1,
        p_disallow_at_sign     => true);
end;
/
```

Per vedere il testo della funzionalità di verifica, eseguire una query a DBA_SOURCE. L'esempio seguente ottiene il testo da una funzionalità di password personalizzata denominata CUSTOM_PASSWORD_FUNCTION.

```
COL TEXT FORMAT a150
```

```
  SELECT TEXT
    FROM DBA_SOURCE
   WHERE OWNER = 'SYS'
     AND NAME = 'CUSTOM_PASSWORD_FUNCTION'
ORDER BY LINE;
```

Per associare la funzionalità di verifica con un profilo utente, utilizza `alter profile`. L'esempio seguente associa una funzionalità di verifica con un profilo utente `DEFAULT`.

```
ALTER PROFILE DEFAULT LIMIT PASSWORD_VERIFY_FUNCTION CUSTOM_PASSWORD_FUNCTION;
```

Per vedere quali profili utente sono associati alle funzionalità di verifica, eseguire una query a `DBA_PROFILES`. L'esempio seguente ottiene i profili che sono associati alla funzionalità di verifica personalizzata denominata `CUSTOM_PASSWORD_FUNCTION`.

```
SELECT * FROM DBA_PROFILES WHERE RESOURCE_NAME = 'PASSWORD' AND LIMIT =
 'CUSTOM_PASSWORD_FUNCTION';


PROFILE                    RESOURCE_NAME                    RESOURCE  LIMIT
-------------------------  -------------------------------  --------
 ------------------------
DEFAULT                    PASSWORD_VERIFY_FUNCTION         PASSWORD
 CUSTOM_PASSWORD_FUNCTION
```

L'esempio seguente ottiene tutti i profili e la funzionalità di verifica della password alla quale sono associati.

```
SELECT * FROM DBA_PROFILES WHERE RESOURCE_NAME = 'PASSWORD_VERIFY_FUNCTION';

PROFILE                    RESOURCE_NAME                    RESOURCE  LIMIT
-------------------------  -------------------------------  --------
 ------------------------
DEFAULT                    PASSWORD_VERIFY_FUNCTION         PASSWORD
 CUSTOM_PASSWORD_FUNCTION
RDSADMIN                   PASSWORD_VERIFY_FUNCTION         PASSWORD  NULL
```

La procedura create_passthrough_verify_fcn

La procedura `create_passthrough_verify_fcn` è supportata per RDS for Oracle versione 12.1.0.2.v7 e tutte le successive versioni principali e secondarie.

Puoi creare una funzione personalizzata per verificare le password usando la procedura in Amazon RDS `rdsadmin.rdsadmin_password_verify.create_passthrough_verify_fcn`. La procedura `create_passthrough_verify_fcn` include i seguenti parametri.

Nome del parametro	Tipo di dati	Default	Campo obbligatorio	Descrizione
`p_verify_function_name`	varchar2	—	Sì	Il nome per la funzionalità di verifica personalizzata. Questa è una funzionalità wrapper creata per te nello schema SYS e non contiene nessuna logica di verifica. Assegni questa funzione a profili di utente.
`p_target_owner`	varchar2	—	Sì	Il proprietario dello schema per la funzionalità di verifica personalizzata.
`p_target_function_name`	varchar2	—	Sì	Il nome della funzionalità personalizzata esistente che contiene una logica di verifica. La funzionalità personalizzata deve restituire un booleano. La funzionalità deve restituire `true` se la password è valida e `false` se la password non è valida.

L'esempio seguente crea una funzionalità di verifica della password che utilizza la logica dalla funzionalità denominata `PASSWORD_LOGIC_EXTRA_STRONG`.

```
begin
    rdsadmin.rdsadmin_password_verify.create_passthrough_verify_fcn(
        p_verify_function_name => 'CUSTOM_PASSWORD_FUNCTION',
        p_target_owner         => 'TEST_USER',
        p_target_function_name => 'PASSWORD_LOGIC_EXTRA_STRONG');
end;
/
```

Per associare la funzionalità di verifica con un profilo utente, utilizza `alter profile`. L'esempio seguente associa la funzionalità di verifica con un profilo utente `DEFAULT`.

```
ALTER PROFILE DEFAULT LIMIT PASSWORD_VERIFY_FUNCTION CUSTOM_PASSWORD_FUNCTION;
```

Impostazione di un server DNS personalizzato

Amazon RDS supporta l'accesso di rete in uscita sull'istanza database che esegue Oracle. Per ulteriori informazioni sull'accesso di rete in uscita, inclusi i prerequisiti, consulta Configurazione dell'accesso UTL_HTTP utilizzando certificati e un portafoglio Oracle.

Amazon RDS Oracle permette la risoluzione Domain Name Service (DNS) da un server DNS personalizzato di proprietà del cliente. È possibile risolvere solo nomi di dominio completamente qualificati dall'istanza database Amazon RDS tramite il server DNS personalizzato.

Dopo aver impostato il server dei nomi DNS personalizzato, ci vogliono circa 30 minuti per propagare le modifiche all'istanza database. Dopo che le modifiche vengono propagate all'istanza database, tutto il traffico di rete in uscita che richiede una ricerca DNS esegue una query al server DNS tramite la porta 53.

Per impostare un server DNS personalizzato per l'istanza database Amazon RDS for Oracle, procedi come segue:

- Dal set di opzioni DHCP collegate al Virtual Private Cloud (VPC), imposta l'opzione `domain-name-servers` sull'indirizzo IP del server dei nomi DNS. Per ulteriori informazioni, consulta la pagina relativa ai Set di opzioni DHCP.

> **Note**
>
> L'opzione `domain-name-servers` accetta fino a quattro valori, ma l'istanza database Amazon RDS usa solo il primo valore.

- Assicurati che il server DNS possa risolvere tutte le query di ricerca, compresi i nomi DNS pubblici, i nomi DNS privati Amazon EC2 e i nomi DNS specifici per i clienti. Se il traffico di rete in uscita contiene ricerche DNS che il server DNS non può gestire, il server DNS deve avere fornitori DNS upstream appropriati configurati.
- Configura il server DNS per produrre risposte UDP (User Datagram Protocol) di 512 byte o meno.
- Configura il server DNS per produrre risposte TCP (Transmission Control Protocol) di 1024 byte o meno.
- Configura il server DNS per consentire il traffico in entrata dalle istanze database Amazon RDS tramite la porta 53. Se il server DNS si trova in un Amazon VPC, il VPC deve avere un gruppo di sicurezza che contiene regole in entrata che permettono traffico UDP e TCP tramite la porta 53. Se il server DNS non si trova in un Amazon VPC, deve avere una whitelist firewall appropriata per permettere traffico in entrata UDP e TCP tramite la porta 53.

  Per ulteriori informazioni, consulta Gruppi di sicurezza per il VPC e Aggiunta e rimozione di regole.
- Configura il VPC dell'istanza database Amazon RDS per permettere traffico in uscita tramite la porta 53. Il VPC deve avere un gruppo di sicurezza che contiene regole in uscita che permettono traffico UDP e TCP tramite la porta 53.

  Per ulteriori informazioni, consulta Gruppi di sicurezza per il VPC e Aggiunta e rimozione di regole.
- Il percorso di routing tra l'istanza database Amazon RDS e il server DNS deve essere configurata correttamente per consentire traffico DNS.
  - Se l'istanza database Amazon RDS e il server DNS non si trovano nello stesso VPC, una connessione peer deve essere configurata tra loro. Per ulteriori informazioni, consulta Che cos'è il VPC in peering?

Impostazione e annullamento dell'impostazione degli eventi diagnostici di sistema

Per impostare e annullare l'impostazione degli eventi diagnostici a livello di sessione, è possibile utilizzare l'istruzione Oracle SQL `ALTER SESSION SET EVENTS`. Tuttavia, per impostare gli eventi a livello di sistema non è possibile utilizzare Oracle SQL. Utilizzare invece le procedure evento di

sistema nel pacchetto `rdsadmin.rdsadmin_util`. Le procedure evento di sistema sono disponibili nelle seguenti versioni del motore:

- Tutte le versioni di Oracle Database 21c
- 19.0.0.0.ru-2020-10.rur-2020-10.r1 e versioni successive di Oracle Database 19c

  Per ulteriori informazioni, consultare Versione 19.0.0.0.ru-2020-10.rur-2020-10.r1 nelle Note di rilascio di Amazon RDS for Oracle.
- 12.2.0.1.ru-2020-10.rur-2020-10.r1 e versioni successive di Oracle Database 12c Release 2 (12.2.0.1)

  Per ulteriori informazioni, consultare Versione 12.2.0.1.ru-2020-10.rur-2020-10.r1 nelle Note di rilascio di Amazon RDS for Oracle.
- 12.1.0.2.V22 e versioni successive di Oracle Database 12c Release 1 (12.1.0.2)

  Per ulteriori informazioni, consultare Versione 12.1.0.2.v22 nelle Note di rilascio di Amazon RDS for Oracle.

para

> **Important**
>
> Internamente, il pacchetto `rdsadmin.rdsadmin_util` imposta gli eventi utilizzando l'istruzione `ALTER SYSTEM SET EVENTS`. Questa istruzione `ALTER SYSTEM` non è documentata nella documentazione di Oracle Database. Alcuni eventi di diagnostica del sistema possono generare grandi quantità di informazioni di traccia, causare contese o influire sulla disponibilità del database. Si consiglia di testare eventi diagnostici specifici nel database non di produzione e impostare gli eventi nel database di produzione solo sotto la guida del supporto Oracle.

Elenco degli eventi diagnostici di sistema consentiti

Per elencare gli eventi di sistema che è possibile impostare, attenersi alla Amazon RDS procedura `rdsadmin.rdsadmin_util.list_allowed_system_events`. Questa procedura non accetta parametri.

Nell'esempio seguente sono elencati tutti gli eventi di sistema che è possibile impostare.

```
SET SERVEROUTPUT ON
EXEC rdsadmin.rdsadmin_util.list_allowed_system_events;
```

Nell'output di esempio seguente sono elencati i numeri degli eventi e le relative descrizioni. Utilizzare le Amazon RDS procedure `set_system_event` per impostare questi eventi e `unset_system_event` per disimpostarli.

```
604   - error occurred at recursive SQL level
942   - table or view does not exist
1401  - inserted value too large for column
1403  - no data found
1410  - invalid ROWID
1422  - exact fetch returns more than requested number of rows
1426  - numeric overflow
1427  - single-row subquery returns more than one row
1476  - divisor is equal to zero
1483  - invalid length for DATE or NUMBER bind variable
1489  - result of string concatenation is too long
1652  - unable to extend temp segment by  in tablespace
1858  - a non-numeric character was found where a numeric was expected
4031  - unable to allocate  bytes of shared memory ("","","","")
6502  - PL/SQL: numeric or value error
10027 - Specify Deadlock Trace Information to be Dumped
10046 - enable SQL statement timing
10053 - CBO Enable optimizer trace
10173 - Dynamic Sampling time-out error
10442 - enable trace of kst for ORA-01555 diagnostics
12008 - error in materialized view refresh path
12012 - error on auto execute of job
12504 - TNS:listener was not given the SERVICE_NAME in CONNECT_DATA
14400 - inserted partition key does not map to any partition
31693 - Table data object  failed to load/unload and is being skipped due to error:
```

> **Note**
>
> L'elenco degli eventi di sistema consentiti può cambiare nel tempo. Per assicurarti di avere l'elenco degli eventi idonei più recente, usa `rdsadmin.rdsadmin_util.list_allowed_system_events`.

Impostazione degli eventi di diagnostica del sistema

Per impostare un evento di sistema, usa la Amazon RDS procedura `rdsadmin.rdsadmin_util.set_system_event`. È possibile impostare solo gli eventi elencati nell'output di `rdsadmin.rdsadmin_util.list_allowed_system_events`. La procedura `set_system_event` include i seguenti parametri.

Nome del parametro	Tipo di dati	Default	Campo obbligatorio	Descrizione
`p_event`	numero	—	Sì	Il numero dell'evento di sistema. Il valore deve essere uno dei numeri degli eventi segnalati da `list_allowed_system_events` .
`p_level`	numero	—	Sì	Il livello dell'evento. Per le descrizioni dei valori di livello diversi, consulta la documentazione di Oracle Database o Oracle Support.

La procedura `set_system_event` costruisce ed esegue le istruzioni `ALTER SYSTEM SET EVENTS` richieste secondo i seguenti principi:

- Il tipo di evento (`context` o `errorstack`) viene determinato automaticamente.
- Un'istruzione nel modulo `ALTER SYSTEM SET EVENTS 'event LEVEL event_level'` imposta gli eventi di contesto. Questa notazione è equivalente a `ALTER SYSTEM SET EVENTS 'event TRACE NAME CONTEXT FOREVER, LEVEL event_level'`.
- Un'istruzione nel modulo `ALTER SYSTEM SET EVENTS 'event ERRORSTACK (event_level)'` imposta gli eventi stack di errore. Questa notazione è equivalente a `ALTER SYSTEM SET EVENTS 'event TRACE NAME ERRORSTACK LEVEL event_level'`.

Nell'esempio seguente viene impostato l'evento 942 al livello 3 e l'evento 10442 al livello 10. L'output di esempio è incluso.

```
SQL> SET SERVEROUTPUT ON
SQL> EXEC rdsadmin.rdsadmin_util.set_system_event(942,3);
Setting system event 942 with: alter system set events '942 errorstack (3)'

PL/SQL procedure successfully completed.

SQL> EXEC rdsadmin.rdsadmin_util.set_system_event(10442,10);
Setting system event 10442 with: alter system set events '10442 level 10'

PL/SQL procedure successfully completed.
```

Elenco degli eventi diagnostici di sistema impostati

Per elencare gli eventi di sistema correntemente impostati, utilizza la Amazon RDS procedura `rdsadmin.rdsadmin_util.list_set_system_events`. Questa procedura segnala solo gli eventi impostati a livello di sistema da `set_system_event`.

Nell'esempio seguente vengono elencati gli eventi di sistema attivi.

```
SET SERVEROUTPUT ON
EXEC rdsadmin.rdsadmin_util.list_set_system_events;
```

Nell'output di esempio seguente viene illustrato l'elenco degli eventi, il tipo di evento, il livello in cui gli eventi sono attualmente impostati e l'ora in cui è stato impostato l'evento.

```
942 errorstack (3) - set at 2020-11-03 11:42:27
10442 level 10 - set at 2020-11-03 11:42:41

PL/SQL procedure successfully completed.
```

Annullamento dell'impostazione degli eventi diagnostici del sistema

Per annullare l'impostazione di un evento di sistema, attenersi alla Amazon RDS procedura `rdsadmin.rdsadmin_util.unset_system_event`. È possibile annullare solo gli eventi elencati nell'output di `rdsadmin.rdsadmin_util.list_allowed_system_events`. La procedura `unset_system_event` accetta il seguente parametro.

Nome del parametro	Tipo di dati	Default	Campo obbligatorio	Descrizione
`p_event`	numero	—	Sì	Il numero dell'evento di sistema. Il valore deve essere uno dei numeri degli eventi segnalati da `list_allowed_system_events` .

Nell'esempio seguente vengono disimpostati gli eventi 942 e 10442. L'output di esempio è incluso.

```
SQL> SET SERVEROUTPUT ON
SQL> EXEC rdsadmin.rdsadmin_util.unset_system_event(942);
Unsetting system event 942 with: alter system set events '942 off'

PL/SQL procedure successfully completed.

SQL> EXEC rdsadmin.rdsadmin_util.unset_system_event(10442);
Unsetting system event 10442 with: alter system set events '10442 off'

PL/SQL procedure successfully completed.
```

Esecuzione di attività di database comuni per le istanze database Oracle

Di seguito, viene descritto come eseguire determinate attività DBA comuni relative ai database nelle istanze database Amazon RDS che eseguono Oracle. Per offrire un'esperienza di servizio gestito, Amazon RDS non fornisce accesso shell alle istanze database. Amazon RDS limita anche l'accesso ad alcune procedure di sistema e tabelle che richiedono privilegi avanzati.

Argomenti

- Modifica del nome globale di un database
- Creazione e dimensionamento di spazi tabelle
- Impostazione dello spazio di tabella predefinito
- Impostazione dello spazio di tabella temporaneo predefinito
- Creazione di un spazio di tabella temporaneo nell'archivio dell'istanza

- Aggiunta di un file temporaneo all'archivio dell'istanza in una replica di lettura
- Rilascio di file temporanei in una replica di lettura
- Checkpoint di un database
- Impostazione del ripristino distribuito
- Impostazione del fuso orario del database
- Lavorare con le tabelle esterne Oracle
- Generazione di report sulle prestazioni con AWR (Automatic Workload Repository)
- Modifica dei collegamenti di database per l'utilizzo con le istanze database in un VPC
- Impostazione dell'edizione predefinita per un'istanza database
- Abilitazione dell'audit per la tabella SYS.AUD$
- Disabilitazione dell'audit per la tabella SYS.AUD$
- Pulizia di compilazioni dell'indice online interrotte
- Ignorare blocchi corrotti
- Ridimensionamento di spazi di tabella, file di dati e file temporanei
- Eliminazione del cestino riciclaggio
- Impostazione dei valori di default visualizzati per la redazione completa

Modifica del nome globale di un database

Puoi usare la procedura in Amazon RDS per modificare il nome globale di un database `rdsadmin.rdsadmin_util.rename_global_name`. La procedura `rename_global_name` include i seguenti parametri.

Nome del parametro	Tipo di dati	Default	Campo obbligatorio	Descrizione
`p_new_global_name`	varchar2	—	Sì	Il nuovo nome globale per il database.

Il database deve essere aperto affinché la modifica del nome abbia luogo. Per ulteriori informazioni sulla modifica del nome globale di un database, consulta ALTER DATABASE nella documentazione di Oracle.

L'esempio seguente modifica il nome globale del database in `new_global_name`.

```
EXEC rdsadmin.rdsadmin_util.rename_global_name(p_new_global_name => 'new_global_name');
```

Creazione e dimensionamento di spazi tabelle

Amazon RDS supporta Oracle Managed Files (OMF) solo per i file di dati, i file di log e i file di controllo. Quando crei file di dati e file di log, non puoi specificare i nomi fisici dei file.

Per impostazione predefinita, se non si specifica una dimensione del file di dati, gli spazi di tabella vengono creati con il valore predefinito di `AUTOEXTEND ON` e nessuna dimensione massima. Nell'esempio seguente, lo spazio di tabella *`users1`* è estensibile automaticamente.

```
CREATE TABLESPACE users1;
```

A causa di queste impostazioni predefinite, gli spazi tabelle possono aumentare e occupare tutto lo storage allocato. Consigliamo di specificare una dimensione massima appropriata per spazi tabelle permanenti e temporanei e che monitori attentamente l'utilizzo di spazio.

L'esempio seguente crea uno spazio tabella denominato *`users2`* con una dimensione iniziale di 1 gigabyte. Poiché la dimensione di un file di dati è specificata, ma `AUTOEXTEND ON` non è specificato, lo spazio di tabella non è estensibile automaticamente.

```
CREATE TABLESPACE users2 DATAFILE SIZE 1G;
```

L'esempio seguente crea uno spazio tabella denominato *`users3`* con una dimensione iniziale di 1 gigabyte, estensibile automaticamente e una dimensione massima di 10 gigabyte:

```
CREATE TABLESPACE users3 DATAFILE SIZE 1G AUTOEXTEND ON MAXSIZE 10G;
```

L'esempio seguente crea uno spazio tabella temporaneo denominato *`temp01`*.

```
CREATE TEMPORARY TABLESPACE temp01;
```

Puoi ridimensionare uno spazio tabella di file di grandi dimensioni utilizzando `ALTER TABLESPACE`. Puoi specificare le dimensioni in kilobyte (K), megabyte (M), gigabyte (G), o terabyte (T). L'esempio seguente ridimensiona uno spazio tabella di un file di grandi dimensioni denominato *`users_bf`* a 200 MB.

```
ALTER TABLESPACE users_bf RESIZE 200M;
```

L'esempio seguente aggiunge un file di dati aggiuntivo a uno spazio tabella di un file di piccole dimensioni denominato *users_sf*.

```
ALTER TABLESPACE users_sf ADD DATAFILE SIZE 100000M AUTOEXTEND ON NEXT 250m
 MAXSIZE UNLIMITED;
```

Impostazione dello spazio di tabella predefinito

Puoi usare la procedura in Amazon RDS per impostare lo spazio di tabella predefinito `rdsadmin.rdsadmin_util.alter_default_tablespace`. La procedura `alter_default_tablespace` include i seguenti parametri.

Nome del parametro	Tipo di dati	Default	Campo obbligatorio	Descrizione
`tablespace_name`	varchar	—	Sì	Il nome dello spazio tabella predefinito.

L'esempio seguente imposta lo spazio tabella predefinito su *users2*:

```
EXEC rdsadmin.rdsadmin_util.alter_default_tablespace(tablespace_name => 'users2');
```

Impostazione dello spazio di tabella temporaneo predefinito

Puoi usare la procedura in Amazon RDS per impostare lo spazio di tabella temporaneo predefinito `rdsadmin.rdsadmin_util.alter_default_temp_tablespace`. La procedura `alter_default_temp_tablespace` include i seguenti parametri.

Nome del parametro	Tipo di dati	Default	Campo obbligatorio	Descrizione
`tablespace_name`	varchar	—	Sì	Il nome dello spazio tabella predefinito temporaneo.

L'esempio seguente imposta lo spazio tabella predefinito temporaneo su *`temp01`*.

```
EXEC rdsadmin.rdsadmin_util.alter_default_temp_tablespace(tablespace_name => 'temp01');
```

Creazione di un spazio di tabella temporaneo nell'archivio dell'istanza

Per creare uno spazio di tabella temporaneo nell'archivio dell'istanza, utilizza la procedura Amazon RDS `rdsadmin.rdsadmin_util.create_inst_store_tmp_tblspace`. La procedura `create_inst_store_tmp_tblspace` include i seguenti parametri.

Nome del parametro	Tipo di dati	Default	Campo obbligatorio	Descrizione
`p_tablespace_name`	varchar	—	Sì	Il nome dello spazio di tabella temporaneo.

L'esempio seguente crea lo spazio di tabella temporaneo *`temp01`* nell'archivio dell'istanza.

```
EXEC rdsadmin.rdsadmin_util.create_inst_store_tmp_tblspace(p_tablespace_name =>
 'temp01');
```

> **Important**
>
> Durante l'esecuzione di `rdsadmin_util.create_inst_store_tmp_tblspace`, lo spazio di tabella temporaneo appena creato non viene impostato automaticamente

come spazio di tabella temporaneo predefinito. Per impostarlo come predefinito, consulta Impostazione dello spazio di tabella temporaneo predefinito.

Per ulteriori informazioni, consulta Archiviazione di dati temporanei in un archivio dell'istanza RDS per Oracle.

Aggiunta di un file temporaneo all'archivio dell'istanza in una replica di lettura

Quando si crea uno spazio di tabella temporaneo in un'istanza database primaria, la replica di lettura non crea i file temporanei. Supponi che nella replica di lettura esista uno spazio di tabella temporaneo vuoto per uno dei seguenti motivi:

- Hai eliminato un file temporaneo dallo spazio di tabella temporaneo nella replica di lettura. Per ulteriori informazioni, consulta Rilascio di file temporanei in una replica di lettura.
- Hai creato un nuovo spazio di tabella temporaneo nell'istanza database primaria. In questo caso, RDS per Oracle sincronizza i metadati con la replica di lettura.

Puoi aggiungere un file temporaneo allo spazio di tabella temporaneo vuoto e archiviare il file temporaneo nell'archivio dell'istanza. Per creare un file temporaneo nell'archivio dell'istanza, utilizza la procedura Amazon RDS `rdsadmin.rdsadmin_util.add_inst_store_tempfile`. Puoi utilizzare questa procedura solo in una replica di lettura. La procedura include i seguenti parametri.

Nome del parametro	Tipo di dati	Default	Campo obbligatorio	Descrizione
`p_tablespace_name`	varchar	—	Sì	Il nome dello spazio di tabella temporaneo nella replica di lettura.

Nell'esempio seguente, lo spazio di tabella temporaneo vuoto *`temp01`* è presente nella replica di lettura. Esegui il comando seguente per creare un file temporaneo per la tabella e archiviarlo nell'archivio dell'istanza.

```
EXEC rdsadmin.rdsadmin_util.add_inst_store_tempfile(p_tablespace_name => 'temp01');
```

Per ulteriori informazioni, consulta Archiviazione di dati temporanei in un archivio dell'istanza RDS per Oracle.

Rilascio di file temporanei in una replica di lettura

Non è possibile rilasciare uno spazio di tabella temporaneo esistente in una replica di lettura. Puoi modificare l'archivio dei file temporanei in una replica di lettura da Amazon EBS nell'archivio dell'istanza o dall'archivio dell'istanza ad Amazon EBS. Per raggiungere questi obiettivi, effettua le seguenti operazioni:

1. Rilascia i file temporanei correnti nello spazio di tabella temporaneo nella replica di lettura.
2. Crea nuovi file temporanei in un altro archivio.

Per rilasciare i file temporanei, utilizza la procedura Amazon RDS `rdsadmin.rdsadmin_util.drop_replica_tempfiles`. Puoi utilizzare questa procedura solo nelle repliche di lettura. La procedura `drop_replica_tempfiles` include i seguenti parametri.

Nome del parametro	Tipo di dati	Default	Campo obbligatorio	Descrizione
`p_tablespace_name`	varchar	—	Sì	Il nome dello spazio di tabella temporaneo nella replica di lettura.

Supponi che uno spazio di tabella temporaneo denominata *`temp01`* si trovi nell'archivio dell'istanza della replica di lettura. Rilascia tutti i file temporanei in questo spazio di tabella eseguendo il comando seguente.

```
EXEC rdsadmin.rdsadmin_util.drop_replica_tempfiles(p_tablespace_name => 'temp01');
```

Per ulteriori informazioni, consulta Archiviazione di dati temporanei in un archivio dell'istanza RDS per Oracle.

Checkpoint di un database

Puoi usare la procedura in Amazon RDS per eseguire il checkpoint del database `rdsadmin.rdsadmin_util.checkpoint`. La procedura `checkpoint` non ha parametri.

L'esempio seguente esegue il checkpoint del database.

```
EXEC rdsadmin.rdsadmin_util.checkpoint;
```

Impostazione del ripristino distribuito

Puoi usare le procedure `rdsadmin.rdsadmin_util.enable_distr_recovery` e `disable_distr_recovery` in Amazon RDS per impostare il ripristino distribuito. Le procedure non hanno parametri.

L'esempio seguente abilita il ripristino distribuito.

```
EXEC rdsadmin.rdsadmin_util.enable_distr_recovery;
```

L'esempio seguente disabilita il ripristino distribuito.

```
EXEC rdsadmin.rdsadmin_util.disable_distr_recovery;
```

Impostazione del fuso orario del database

È possibile impostare il fuso orario del database Amazon RDS Oracle nei modi seguenti:

- L'opzione `Timezone`

  L'opzione `Timezone` modifica il fuso orario a livello di host e interessa tutte le colonne e valori data come `SYSDATE`. Per ulteriori informazioni, consulta Fuso orario Oracle.
- La procedura Amazon RDS `rdsadmin.rdsadmin_util.alter_db_time_zone`

  La procedura `alter_db_time_zone` modifica il fuso orario solo per certi tipi di dati e non modifica `SYSDATE`. Ci sono limitazioni aggiuntive per l'impostazione del fuso orario elencato nella Documentazione di Oracle.

> **Note**
>
> È inoltre possibile impostare il fuso orario predefinito per Oracle Scheduler. Per ulteriori informazioni, consulta Impostazione del fuso orario per i job di Oracle Scheduler.

La procedura `alter_db_time_zone` include i seguenti parametri.

Nome del parametro	Tipo di dati	Default	Campo obbligatorio	Descrizione
p_new_tz	varchar2	—	Sì	Il nuovo fuso orario come regione denominata o un offset assoluto da Coordinated Universal Time (UTC). Gli offset validi sono compresi tra -12.00 e +14.00.

Il seguente esempio imposta il fuso orario su UTC più tre ore.

```
EXEC rdsadmin.rdsadmin_util.alter_db_time_zone(p_new_tz => '+3:00');
```

L'esempio seguente imposta il fuso orario sul fuso orario della regione Africa/Algeri.

```
EXEC rdsadmin.rdsadmin_util.alter_db_time_zone(p_new_tz => 'Africa/Algiers');
```

Dopo aver modificato il fuso orario utilizzando la procedura `alter_db_time_zone`, devi riavviare l'istanza database per rendere effettive le modifiche. Per ulteriori informazioni, consulta Riavvio di un'istanza database. Per informazioni sull'aggiornamento dei fusi orari, consulta Considerazioni sul fuso orario.

Lavorare con le tabelle esterne Oracle

Le Tabelle esterne Oracle sono tabelle con dati che non si trovano nel database. Invece, i dati si trovano nei file esterni ai quali il database può accedere. Utilizzando le tabelle esterne, puoi accedere ai dati senza caricarli nel database. Per ulteriori informazioni sulle tabelle esterne, consulta Gestione delle tabelle esterne nella documentazione Oracle.

Con Amazon RDS, puoi archiviare i file della tabella esterna negli oggetti della directory. Puoi creare un oggetto di directory o puoi utilizzare uno predefinito nel database Oracle, ad esempio la directory DATA_PUMP_DIR. Per informazioni sulla creazione di oggetti di directory, consulta Creazione ed eliminazione di directory nello spazio di archiviazione dati principale. Puoi eseguire query sulla

visualizzazione ALL_DIRECTORIES per elencare gli oggetti di directory per l'istanza database Amazon RDS Oracle.

> **Note**
>
> Gli oggetti directory puntano allo spazio principale dello storage dei dati (volume Amazon EBS) utilizzato dall'istanza. Lo spazio usato—insieme a file di dati, log delle modifiche, audit, traccia e altri file— fa parte dello storage allocato.

Puoi spostare un file di dati esterno da un database Oracle a un altro utilizzando il pacchetto DBMS_FILE_TRANSFER o UTL_FILE. I file dati esterni si spostano da una directory nel database origine a una directory specificata nel database di destinazione. Per ulteriori informazioni su DBMS_FILE_TRANSFER, consulta Importazione utilizzando Oracle Data Pump.

Dopo aver spostato il file dei dati esterno puoi crearci una tabella esterna. L'esempio seguente crea una tabella esterna che utilizza il file `emp_xt_file1.txt` nella directory USER_DIR1.

```
CREATE TABLE emp_xt (
  emp_id      NUMBER,
  first_name  VARCHAR2(50),
  last_name   VARCHAR2(50),
  user_name   VARCHAR2(20)
)
ORGANIZATION EXTERNAL (
  TYPE ORACLE_LOADER
  DEFAULT DIRECTORY USER_DIR1
  ACCESS PARAMETERS (
    RECORDS DELIMITED BY NEWLINE
    FIELDS TERMINATED BY ','
    MISSING FIELD VALUES ARE NULL
    (emp_id,first_name,last_name,user_name)
  )
  LOCATION ('emp_xt_file1.txt')
)
PARALLEL
REJECT LIMIT UNLIMITED;
```

Immaginiamo che desideri spostare i dati che si trovano nell'istanza database Amazon RDS Oracle nel file di dati esterno. In questo caso, puoi popolare il file di dati esterno creando una tabella esterna

e selezionando i dati dalla tabella nel database. Ad esempio, la seguente istruzione SQL crea la tabella esterna `orders_xt` eseguendo la query sulla tabella `orders` nel database.

```
CREATE TABLE orders_xt
  ORGANIZATION EXTERNAL
   (
     TYPE ORACLE_DATAPUMP
     DEFAULT DIRECTORY DATA_PUMP_DIR
     LOCATION ('orders_xt.dmp')
   )
   AS SELECT * FROM orders;
```

In questo esempio, i dati sono popolati nel file `orders_xt.dmp` nella directory DATA_PUMP_DIR.

Generazione di report sulle prestazioni con AWR (Automatic Workload Repository)

Per raccogliere i dati sulle prestazioni e generare report, Oracle consiglia AWR (Automatic Workload Repository). AWR richiede Oracle Database Enterprise Edition e una licenza per i pacchetti di diagnostica e ottimizzazione. Per abilitare AWR, impostare il parametro di inizializzazione `CONTROL_MANAGEMENT_PACK_ACCESS` su `DIAGNOSTIC` o `DIAGNOSTIC+TUNING`.

Utilizzo di report AWR in RDS

Per generare report AWR, puoi eseguire script quali `awrrpt.sql`. Questi script vengono installati nel server host del database. In Amazon RDS non è possibile accedere direttamente all'host. Tuttavia, puoi ottenere copie di script SQL da un'altra installazione di Oracle Database.

Puoi inoltre utilizzare AWR eseguendo procedure nel pacchetto PL/SQL `SYS.DBMS_WORKLOAD_REPOSITORY`. Puoi utilizzare questo pacchetto per gestire baseline e snapshot, nonché visualizzare report ASH e AWR. Ad esempio, per generare un report AWR in formato di testo, esegui la procedura `DBMS_WORKLOAD_REPOSITORY.AWR_REPORT_TEXT`. Tuttavia, non puoi raggiungere questi report AWR da AWS Management Console.

Quando utilizzi AWR, ti consigliamo di utilizzare le procedure `rdsadmin.rdsadmin_diagnostic_util`. Puoi utilizzare queste procedure per generare quanto segue:

- Report AWR
- Report ASH (Active Session History)
- Report ADDM (Automatic Database Diagnostic Monitor)

- File di dump di Export di Oracle Data Pump di dati AWR

Le procedure `rdsadmin_diagnostic_util` salvano i report nel file system dell'istanza database. Puoi accedere a questi report dalla console. Puoi inoltre accedere ai report utilizzando le procedure `rdsadmin.rds_file_util` e accedere ai report copiati in Simple Storage Service (Amazon S3) mediante l'opzione Integrazione S3. Per ulteriori informazioni, consulta Lettura dei file in una directory di istanze database e Integrazione Amazon S3.

Puoi utilizzare le procedure `rdsadmin_diagnostic_util` nelle seguente per versioni del motore database Amazon RDS for Oracle:

- Tutte le versioni di Oracle Database 21c
- 19.0.0.0.ru-2020-04.rur-2020-04.r1 e versioni successive di Oracle Database 19c
- 12.2.0.1.ru-2020-04.rur-2020-04.r1 e versioni successive di Oracle Database 12c Release 2 (12.2)
- 12.1.0.2.V20 e versioni successive di Oracle Database 12c Release 1 (12.1)

Per un blog che spiega come utilizzare i report diagnostici in uno scenario di replica, consulta il post relativo alla generazione di report AWR per le repliche di lettura di Amazon RDS per Oracle.

Parametri comuni per il pacchetto di utilità di diagnostica

In genere i seguenti parametri vengono utilizzati durante la gestione di AWR e ADDM con il pacchetto `rdsadmin_diagnostic_util`.

Parametro	Tipo di dati	Defaul	Campo obbligatorio	Descrizione
`begin_snap_id`	`NUMBER`	—	Sì	ID dello snapshot iniziale.
`end_snap_id`	`NUMBER`	—	Sì	ID dello snapshot finale.
`dump_directory`	`VARCHAR`	`BDUMP`	No	La directory in cui scrivere il report o esportare il file. Se si specifica una directory non predefinita, l'utente che esegue le procedure `rdsadmin_diagnostic_util`

Parametro	Tipo di dati	Defaul	Campo obbligatorio	Descrizione
				deve disporre delle autorizzazioni di scrittura per la directory.

Parametro	Tipo di dati	Defaul	Campo obbligatorio	Descrizione
`p_tag`	VARCHA	—	No	Una stringa che può essere utilizzata per distinguere tra i backup per indicare lo scopo o l'utilizzo dei backup, ad esempio `incremental` o `daily`. Puoi specificare fino a 30 caratteri. I caratteri validi sono: `a-z`, `A-Z`, `0-9`, un carattere di sottolineatura (_), un trattino (-) e un punto (.). Il tag non rileva la distinzione tra maiuscole e minuscole. RMAN memorizza sempre i tag in maiuscolo, indipendentemente dalle maiuscole e minuscole utilizzate durante l'inserimento. I tag non devono essere univoci, quindi più backup possono avere lo stesso tag. Se non specifichi un tag, RMAN assegna automaticamente un tag di default utilizzando il formato `TAG`*`YYYYMMDDTHHMMSS`*, dove *`YYYY`* è l'anno, *`MM`* è il mese, *`DD`* è il giorno, *`HH`* è l'ora (nel formato 24 ore), *`MM`* sono i minuti e *`SS`* sono i secondi. La data e l'ora indicano quando RMAN ha avviato il backup. Ad esempio, un backup con il tag di default `TAG20190927T214517` indica un backup iniziato il 27/09/2019 alle 21:45:17. Il parametro `p_tag` è supportato nelle seguenti versioni del motore database RDS for Oracle: • Oracle Database 21c (21.0.0) • Oracle Database 19c (19.0.0), usando le versioni 19.0.0.0.ru-2021-10.rur-2021-10.r1 e successive • Oracle Database 12c Release 2 (12.2) usando le versioni 12.2.0.1.ru-2021-10.rur-2021-10.r1 o successive • Oracle Database 12c Release 1 (12.1) usando la versione 12.1.0.2.v26 Oracle successive

Parametro	Tipo di dati	Defaul	Campo obbligatorio	Descrizione
report_type	VARCHA	HTML	No	Il formato del report. I valori validi sono TEXT e HTML.
dbid	NUMBEF	—	No	Un identificatore di database (DBID) valido visualizzato nella vista DBA_HIST_DATABASE_INSTANCE per Oracle. Se questo parametro non viene specificato, RDS utilizza il DBID corrente, mostrato nella vista V$DATABASE.DBID .

In genere i seguenti parametri vengono utilizzati durante la gestione di ASH con il pacchetto rdsadmin_diagnostic_util.

Parametro	Tipo di dati	Default	Campo obbligatorio	Descrizione
begin_time	DATE	—	Sì	L'ora di inizio dell'analisi ASH.
end_time	DATE	—	Sì	L'ora di fine dell'analisi ASH.
slot_width	NUMBER	0	No	La durata degli slot (in secondi) utilizzati nella sezione "Attività superiore" del report ASH. Se questo parametro non è specificato, l'intervallo di tempo tra begin_time e end_time utilizza un massimo di 10 slot.
sid	NUMBER	Null	No	L'ID della sessione
sql_id	VARCHAR2	Null	No	L'ID SQL.
wait_class	VARCHAR2	Null	No	Il nome della classe di attesa.

Parametro	Tipo di dati	Default	Campo obbligatorio	Descrizione
service_hash	NUMBER	Null	No	L'hash del nome del servizio.
module_name	VARCHAR2	Null	No	Il nome del modulo.
action_name	VARCHAR2	Null	No	Il nome dell'operazione.
client_id	VARCHAR2	Null	No	L'ID specifico dell'applicazione della sessione del database.
plsql_entry	VARCHAR2	Null	No	Il punto di ingresso PL/SQL.

Generazione di un report AWR

Per generare un report AWR, utilizza la procedura `rdsadmin.rdsadmin_diagnostic_util.awr_report`.

Nell'esempio seguente viene generato un report AWR per l'intervallo di snapshot da 101 a 106. Il file di testo di output è denominato `awrrpt_101_106.txt`. Puoi accedere a questo report da AWS Management Console.

```
EXEC rdsadmin.rdsadmin_diagnostic_util.awr_report(101,106,'TEXT');
```

Nell'esempio seguente viene generato un report HTML per l'intervallo di snapshot da 63 a 65. Il file HTML di output è denominato `awrrpt_63_65.html`. La procedura scrive il report nella directory di database non predefinita denominata `AWR_RPT_DUMP`.

```
EXEC rdsadmin.rdsadmin_diagnostic_util.awr_report(63,65,'HTML','AWR_RPT_DUMP');
```

Estrazione di dati AWR in un file di dump

Per estrarre i dati AWR in un file di dump, utilizza la procedura `rdsadmin.rdsadmin_diagnostic_util.awr_extract`.

Nell'esempio seguente viene estratto l'intervallo di snapshot da 101 a 106. Il file di dump di output è denominato `awrextract_101_106.dmp`. Puoi accedere a questo file tramite la console.

```
EXEC rdsadmin.rdsadmin_diagnostic_util.awr_extract(101,106);
```

Nell'esempio seguente viene estratto l'intervallo di snapshot da 63 a 65. Il file di dump di output è denominato `awrextract_63_65.dmp`. Il file viene archiviato nella directory di database non predefinita denominata `AWR_RPT_DUMP`.

```
EXEC rdsadmin.rdsadmin_diagnostic_util.awr_extract(63,65,'AWR_RPT_DUMP');
```

Generazione di un report ADDM

Per generare un report ADDM, utilizza la procedura `rdsadmin.rdsadmin_diagnostic_util.addm_report`.

Nell'esempio seguente viene generato un report ADDM per l'intervallo di snapshot da 101 a 106. Il file di testo di output è denominato `addmrpt_101_106.txt`. È possibile accedere al report tramite la console.

```
EXEC rdsadmin.rdsadmin_diagnostic_util.addm_report(101,106);
```

Nell'esempio seguente viene generato un report ADDM per l'intervallo di snapshot da 63 a 65. Il file di testo di output è denominato `addmrpt_63_65.txt`. Il file viene archiviato nella directory di database non predefinita denominata `ADDM_RPT_DUMP`.

```
EXEC rdsadmin.rdsadmin_diagnostic_util.addm_report(63,65,'ADDM_RPT_DUMP');
```

Generazione di un report ASH

Per generare un report ASH, utilizza la procedura `rdsadmin.rdsadmin_diagnostic_util.ash_report`.

Nell'esempio seguente viene generato un report ASH che include i dati da 14 minuti fa fino all'ora corrente. Il nome del file di output utilizza il formato `ashrpt`*`begin_time`*`end_time`*`.txt`*, dove

begin_time e *end_time* utilizzano il formato YYYYMMDDHH24MISS. Puoi accedere al file tramite la console.

```
BEGIN
    rdsadmin.rdsadmin_diagnostic_util.ash_report(
        begin_time     =>     SYSDATE-14/1440,
        end_time       =>     SYSDATE,
        report_type    =>     'TEXT');
END;
/
```

Nell'esempio seguente viene generato un report ASH che include i dati dal 18 novembre 2019 alle 18.07 fino al 18 novembre 2019 alle 18.15. Il nome del report HTML di output è `ashrpt_20190918180700_20190918181500.html`. Il file viene archiviato nella directory di database non predefinita denominata `AWR_RPT_DUMP`.

```
BEGIN
    rdsadmin.rdsadmin_diagnostic_util.ash_report(
        begin_time     =>    TO_DATE('2019-09-18 18:07:00', 'YYYY-MM-DD HH24:MI:SS'),
        end_time       =>    TO_DATE('2019-09-18 18:15:00', 'YYYY-MM-DD HH24:MI:SS'),
        report_type    =>    'html',
        dump_directory =>    'AWR_RPT_DUMP');
END;
/
```

Accesso ai report AWR dalla console o da CLI

Per accedere ai report AWR o esportare file di dump, puoi usare o. AWS Management Console AWS CLI Per ulteriori informazioni, consulta Download di un file di log di database.

Modifica dei collegamenti di database per l'utilizzo con le istanze database in un VPC

Per utilizzare i collegamenti di database Oracle con le istanze database Amazon RDS nello stesso Virtual Private Cloud (VPC) o VPC in peering, le due istanze database devono avere un instradamento valido tra loro. Verifica l'instradamento valido tra le istanze database utilizzando le tabelle di routing VPC e la lista di controllo accessi della rete (ACL).

Il gruppo di sicurezza di ogni istanza database deve permettere l'ingresso e l'uscita dall'altra istanza database. Le regole in entrata e in uscita possono riferirsi ai gruppi di sicurezza dallo stesso VPC o da un VPC in peering. Per ulteriori informazioni, consulta Aggiornamento dei gruppi di sicurezza a gruppi di sicurezza VPC in peering di riferimento.

Se hai configurato un server DNS personalizzato utilizzando il set opzioni DHCP nel VPC, il server DNS personalizzato deve essere in grado di risolvere il nome del target del collegamento di database. Per ulteriori informazioni, consulta Impostazione di un server DNS personalizzato.

Per ulteriori informazioni sull'utilizzo di collegamenti di database con Oracle Data Pump, consulta Importazione utilizzando Oracle Data Pump.

Impostazione dell'edizione predefinita per un'istanza database

Puoi ridefinire gli oggetti di database in un ambiente privato che si chiama edizione. Puoi utilizzare una ridefinizione basata sull'edizione per aggiornare gli oggetti di database dell'applicazione con tempo di inattività minimo.

Puoi impostare l'edizione predefinita di un'istanza database Oracle Amazon RDS usando la procedura in Amazon RDS `rdsadmin.rdsadmin_util.alter_default_edition`.

L'esempio seguente imposta l'edizione predefinita per l'istanza database Oracle Amazon RDS su `RELEASE_V1`.

```
EXEC rdsadmin.rdsadmin_util.alter_default_edition('RELEASE_V1');
```

L'esempio seguente imposta l'edizione predefinita per l'istanza database Amazon RDS Oracle a quella predefinita di Oracle.

```
EXEC rdsadmin.rdsadmin_util.alter_default_edition('ORA$BASE');
```

Per ulteriori informazioni sulla ridefinizione Oracle basata sull'edizione, consulta le Informazioni sulla edizione e sulla ridefinizione basata sull'edizione nella documentazione di Oracle.

Abilitazione dell'audit per la tabella SYS.AUD$

Puoi utilizzare la procedura `SYS.AUD$` in Amazon RDS per abilitare l'audit sulla tabella di trail dell'audit del database `rdsadmin.rdsadmin_master_util.audit_all_sys_aud_table`. L'unica proprietà dell'audit supportata è `ALL`. Non è possibile sottoporre o non sottoporre ad audit singole istruzioni o operazioni.

L'abilitazione dell'audit è supportata per le istanze database Oracle che eseguono le seguenti versioni:

- Oracle Database 21c (21.0.0)

- Oracle Database 19c (19.0.0)
- Oracle Database 12c Release 2 (12.2)
- Oracle Database 12c Release 1 (12.1.0.2.v14) e versioni successive

La procedura `audit_all_sys_aud_table` include i seguenti parametri.

Nome del parametro	Tipo di dati	Default	Campo obbligatorio	Descrizione
`p_by_access`	booleano	true	No	Impostato su `true` per l'audit di `BY ACCESS`. Impostato su `false` per l'audit di `BY SESSION`.

> **Note**
>
> In un CDB a tenant singolo funzionano le seguenti operazioni, ma nessun meccanismo visibile al cliente può rilevare lo stato corrente delle operazioni. Le informazioni di verifica non sono disponibili all'interno del PDB. Per ulteriori informazioni, consulta Limitazioni per i CDB RDS per Oracle.

La seguente query restituisce l'attuale configurazione dell'audit di `SYS.AUD$` per un database.

```
SELECT * FROM DBA_OBJ_AUDIT_OPTS WHERE OWNER='SYS' AND OBJECT_NAME='AUD$';
```

I seguenti comandi abilitano l'audit di `ALL` su `SYS.AUD$ BY ACCESS`.

```
EXEC rdsadmin.rdsadmin_master_util.audit_all_sys_aud_table;

EXEC rdsadmin.rdsadmin_master_util.audit_all_sys_aud_table(p_by_access => true);
```

Il seguente comando abilita l'audit di `ALL` su `SYS.AUD$ BY SESSION`.

```
EXEC rdsadmin.rdsadmin_master_util.audit_all_sys_aud_table(p_by_access => false);
```

Per ulteriori informazioni, consulta la sezione relativa all'AUDIT (Audit tradizionale) nella documentazione Oracle.

Disabilitazione dell'audit per la tabella SYS.AUD$

Puoi utilizzare la procedura `SYS.AUD$` in Amazon RDS per disabilitare l'audit sulla tabella di trail dell'audit del database `rdsadmin.rdsadmin_master_util.noaudit_all_sys_aud_table`. Questa procedura non richiede parametri.

La seguente query restituisce l'attuale configurazione dell'audit di `SYS.AUD$` per un database:

```
SELECT * FROM DBA_OBJ_AUDIT_OPTS WHERE OWNER='SYS' AND OBJECT_NAME='AUD$';
```

Il seguente comando disabilita l'audit di `ALL` su `SYS.AUD$`.

```
EXEC rdsadmin.rdsadmin_master_util.noaudit_all_sys_aud_table;
```

Per ulteriori informazioni, consulta la sezione relativa a NOAUDIT (Audit tradizionale) nella documentazione Oracle.

Pulizia di compilazioni dell'indice online interrotte

Per pulire compilazioni dell'indice online non riuscite, utilizza la procedura Amazon RDS `rdsadmin.rdsadmin_dbms_repair.online_index_clean`.

La procedura `online_index_clean` include i seguenti parametri.

Nome del parametro	Tipo di dati	Default	Campo obbligatorio	Descrizione
`object_id`	binary_integer	`ALL_INDEX_ID`	No	L'ID oggetto dell'indice. In genere, puoi utilizzare l'ID oggetto dal testo dell'errore ORA-08104.
`wait_for_lock`	binary_integer	`rdsadmin.rdsadmin_`	No	Specifica `rdsadmin.rdsadmin_`

Nome del parametro	Tipo di dati	Default	Campo obbligatorio	Descrizione
		`dbms_repair.lock_wait`		`dbms_repair.lock_wait` , l'impostazione predefinita, per cercare di ottenere un blocco sull'oggetto sottostante e riprovare finché non viene raggiunto un limite interno se il blocco non va a buon fine. Specifica `rdsadmin.rdsadmin_dbms_repair.lock_nowait` per cercare di ottenere un blocco sull'oggetto sottostante ma non riprovare se il blocco non va a buon fine.

L'esempio seguente pulisce una compilazione di indice online non riuscita:

```
declare
  is_clean boolean;
begin
  is_clean := rdsadmin.rdsadmin_dbms_repair.online_index_clean(
    object_id     => 1234567890,
    wait_for_lock => rdsadmin.rdsadmin_dbms_repair.lock_nowait
  );
end;
/
```

Per ulteriori informazioni, consulta ONLINE_INDEX_CLEAN Function nella documentazione Oracle.

Ignorare blocchi corrotti

Per ignorare blocchi corrotti durante le scansioni di indici e tabelle, utilizza il pacchetto `rdsadmin.rdsadmin_dbms_repair`.

Le seguenti procedure eseguono il wrapping della funzionalità della procedura `sys.dbms_repair.admin_table` e non accettano parametri:

- `rdsadmin.rdsadmin_dbms_repair.create_repair_table`
- `rdsadmin.rdsadmin_dbms_repair.create_orphan_keys_table`
- `rdsadmin.rdsadmin_dbms_repair.drop_repair_table`
- `rdsadmin.rdsadmin_dbms_repair.drop_orphan_keys_table`
- `rdsadmin.rdsadmin_dbms_repair.purge_repair_table`
- `rdsadmin.rdsadmin_dbms_repair.purge_orphan_keys_table`

Le seguenti procedure accettano gli stessi parametri delle loro controparti nel pacchetto `DBMS_REPAIR` per database Oracle:

- `rdsadmin.rdsadmin_dbms_repair.check_object`
- `rdsadmin.rdsadmin_dbms_repair.dump_orphan_keys`
- `rdsadmin.rdsadmin_dbms_repair.fix_corrupt_blocks`
- `rdsadmin.rdsadmin_dbms_repair.rebuild_freelists`
- `rdsadmin.rdsadmin_dbms_repair.segment_fix_status`
- `rdsadmin.rdsadmin_dbms_repair.skip_corrupt_blocks`

Per ulteriori informazioni sulla gestione del danneggiamento del database, vedere DBMS_REPAIR nella documentazione Oracle.

Example Risposta a blocchi danneggiati

Questo esempio mostra il flusso di lavoro di base per rispondere ai blocchi danneggiati. I passaggi dipenderanno dalla posizione e dalla natura del danneggiamento del blocco.

> **Important**
>
> Prima di tentare di riparare i blocchi danneggiati, esaminare attentamente la documentazione di DBMS_REPAIR.

Per saltare i blocchi danneggiati durante le scansioni di indice e tabella

1. Esegui le procedure seguenti per creare tabelle di riparazione se non esistono già.

```
EXEC rdsadmin.rdsadmin_dbms_repair.create_repair_table;
EXEC rdsadmin.rdsadmin_dbms_repair.create_orphan_keys_table;
```

2. Esegui le procedura seguenti per verificare l'esistenza di record e cancellarli se appropriato.

```
SELECT COUNT(*) FROM SYS.REPAIR_TABLE;
SELECT COUNT(*) FROM SYS.ORPHAN_KEY_TABLE;
SELECT COUNT(*) FROM SYS.DBA_REPAIR_TABLE;
SELECT COUNT(*) FROM SYS.DBA_ORPHAN_KEY_TABLE;

EXEC rdsadmin.rdsadmin_dbms_repair.purge_repair_table;
EXEC rdsadmin.rdsadmin_dbms_repair.purge_orphan_keys_table;
```

3. Esegui la procedura seguente per verificare la presenza di blocchi corrotti.

```
SET SERVEROUTPUT ON
DECLARE v_num_corrupt INT;
BEGIN
  v_num_corrupt := 0;
  rdsadmin.rdsadmin_dbms_repair.check_object (
    schema_name => '&corruptionOwner',
    object_name => '&corruptionTable',
    corrupt_count =>  v_num_corrupt
  );
  dbms_output.put_line('number corrupt: '||to_char(v_num_corrupt));
END;
/

COL CORRUPT_DESCRIPTION FORMAT a30
COL REPAIR_DESCRIPTION FORMAT a30

SELECT OBJECT_NAME, BLOCK_ID, CORRUPT_TYPE, MARKED_CORRUPT,
```

```
       CORRUPT_DESCRIPTION, REPAIR_DESCRIPTION
FROM   SYS.REPAIR_TABLE;

SELECT SKIP_CORRUPT
FROM   DBA_TABLES
WHERE  OWNER = '&corruptionOwner'
AND    TABLE_NAME = '&corruptionTable';
```

4. Esegui la procedura `skip_corrupt_blocks` per abilitare o disabilitare l'omissione della corruzione per le tabelle interessate. A seconda della situazione, potrebbe essere necessario estrarre i dati in una nuova tabella e quindi eliminare la tabella contenente il blocco danneggiato.

   Esegui la procedura seguente per ignorare la corruzione per le tabelle interessate.

   ```
   begin
     rdsadmin.rdsadmin_dbms_repair.skip_corrupt_blocks (
       schema_name => '&corruptionOwner',
       object_name => '&corruptionTable',
       object_type => rdsadmin.rdsadmin_dbms_repair.table_object,
       flags => rdsadmin.rdsadmin_dbms_repair.skip_flag);
   end;
   /
   select skip_corrupt from dba_tables where owner = '&corruptionOwner' and table_name
    = '&corruptionTable';
   ```

   Esegui la procedura seguente per non ignorare la corruzione.

   ```
   begin
     rdsadmin.rdsadmin_dbms_repair.skip_corrupt_blocks (
       schema_name => '&corruptionOwner',
       object_name => '&corruptionTable',
       object_type => rdsadmin.rdsadmin_dbms_repair.table_object,
       flags => rdsadmin.rdsadmin_dbms_repair.noskip_flag);
   end;
   /

   select skip_corrupt from dba_tables where owner = '&corruptionOwner' and table_name
    = '&corruptionTable';
   ```

5. Dopo aver completato tutti i lavori di riparazione, eseguire le procedure seguenti per eliminare le tabelle di ripristino.

```
EXEC rdsadmin.rdsadmin_dbms_repair.drop_repair_table;
EXEC rdsadmin.rdsadmin_dbms_repair.drop_orphan_keys_table;
```

Ridimensionamento di spazi di tabella, file di dati e file temporanei

Come impostazione predefinita, gli spazi tabelle Oracle sono creati con l'estensione automatica attivata e nessuna dimensione massima. A causa delle impostazioni predefinite, gli spazi tabella possono a volte diventare troppo grandi. Consigliamo di specificare una dimensione massima appropriata per spazi tabelle permanenti e temporanei e che monitori attentamente l'utilizzo di spazio.

Ridimensionamento degli spazi di tabella permanenti

Per ridimensionare uno spazio di tabella permanente in un'istanza database RDS per Oracle, utilizza una delle seguenti procedure Amazon RDS:

- `rdsadmin.rdsadmin_util.resize_datafile`
- `rdsadmin.rdsadmin_util.autoextend_datafile`

La procedura `resize_datafile` include i seguenti parametri.

Nome del parametro	Tipo di dati	Default	Campo obbligatorio	Descrizione
`p_data_file_id`	numero	—	Sì	L'identificatore del file di dati da ridimensionare.
`p_size`	varchar2	—	Sì	Le dimensioni del file di dati. Specifica le dimensioni in byte (impostazione predefinita), kilobyte (K), megabyte (M) o gigabyte (G).

La procedura `autoextend_datafile` include i seguenti parametri.

Nome del parametro	Tipo di dati	Default	Campo obbligatorio	Descrizione
p_data_file_id	numero	—	Sì	L'identificatore del file di dati da ridimensionare.
p_autoextend_state	varchar2	—	Sì	Lo stato della funzionalità di estensione automatica. Specifica ON per estendere automaticamente il file di dati e OFF per disattivare l'estensione automatica.
p_next	varchar2	—	No	Le dimensioni del successivo incremento del file di dati. Specifica le dimensioni in byte (impostazione predefinita), kilobyte (K), megabyte (M) o gigabyte (G).
p_maxsize	varchar2	—	No	Lo spazio massimo su disco consentito per l'estensione automatica. Specifica le dimensioni in byte (impostazione predefinita), kilobyte (K), megabyte (M) o gigabyte (G). È possibile specificare UNLIMITED per rimuovere il limite di dimensione del file.

L'esempio seguente ridimensiona il file di dati da 4 a 500 MB.

```
EXEC rdsadmin.rdsadmin_util.resize_datafile(4,'500M');
```

L'esempio seguente disattiva l'estensione automatica per il file di dati 4. Attiva l'estensione automatica per il file di dati 5, con un incremento di 128 MB e nessuna dimensione massima.

```
EXEC rdsadmin.rdsadmin_util.autoextend_datafile(4,'OFF');
EXEC rdsadmin.rdsadmin_util.autoextend_datafile(5,'ON','128M','UNLIMITED');
```

Ridimensionamento degli spazi di tabella temporanei

Per ridimensionare uno spazio di tabella temporaneo in un'istanza database RDS per Oracle, inclusa una replica di lettura, utilizza una delle seguenti procedure Amazon RDS:

- `rdsadmin.rdsadmin_util.resize_temp_tablespace`
- `rdsadmin.rdsadmin_util.resize_tempfile`
- `rdsadmin.rdsadmin_util.autoextend_tempfile`

La procedura `resize_temp_tablespace` include i seguenti parametri.

Nome del parametro	Tipo di dati	Default	Campo obbligatorio	Descrizione
`p_temp_tablespace_name`	varchar2	—	Sì	Il nome dello spazio tabella temporaneo da ridimensionare.
`p_size`	varchar2	—	Sì	La dimensione dello spazio di tabella. Specifica le dimensioni in byte (impostazione predefinita), kilobyte (K), megabyte (M) o gigabyte (G).

La procedura `resize_tempfile` include i seguenti parametri.

Nome del parametro	Tipo di dati	Default	Campo obbligatorio	Descrizione
`p_temp_file_id`	numero	—	Sì	L'identificatore del file temporaneo da ridimensionare.
`p_size`	varchar2	—	Sì	Le dimensioni del file temporaneo. Specifica le dimensioni in byte (impostazione predefinita), kilobyte (K), megabyte (M) o gigabyte (G).

La procedura `autoextend_tempfile` include i seguenti parametri.

Nome del parametro	Tipo di dati	Default	Campo obbligatorio	Descrizione
`p_temp_file_id`	numero	—	Sì	L'identificatore del file temporaneo da ridimensionare.
`p_autoextend_state`	varchar2	—	Sì	Lo stato della funzionalità di estensione automatica. Specifica `ON` per estendere automaticamente il file temporaneo e `OFF` per disattivare l'estensione automatica.

Nome del parametro	Tipo di dati	Default	Campo obbligatorio	Descrizione
p_next	varchar2	—	No	Le dimensioni del successivo incremento del file temporaneo. Specifica le dimensioni in byte (impostazione predefinita), kilobyte (K), megabyte (M) o gigabyte (G).
p_maxsize	varchar2	—	No	Lo spazio massimo su disco consentito per l'estensione automatica. Specifica le dimensioni in byte (impostazione predefinita), kilobyte (K), megabyte (M) o gigabyte (G). È possibile specificare UNLIMITED per rimuovere il limite di dimensione del file.

I seguenti esempi ridimensionano uno spazio di tabella temporaneo denominato TEMP alla dimensione di 4 GB.

```
EXEC rdsadmin.rdsadmin_util.resize_temp_tablespace('TEMP','4G');
```

```
EXEC rdsadmin.rdsadmin_util.resize_temp_tablespace('TEMP','4096000000');
```

Il seguente esempio ridimensiona uno spazio tabella temporaneo basato sul file temporaneo con l'identificatore file 1 alla dimensione di 2 MB.

```
EXEC rdsadmin.rdsadmin_util.resize_tempfile(1,'2M');
```

L'esempio seguente disattiva l'estensione automatica per il file temporaneo 1. Imposta la dimensione massima dell'estensione automatica del file temporaneo da 2 a 10 GB, con un incremento di 100 MB.

```
EXEC rdsadmin.rdsadmin_util.autoextend_tempfile(1,'OFF');
EXEC rdsadmin.rdsadmin_util.autoextend_tempfile(2,'ON','100M','10G');
```

Per maggiori informazioni sulle repliche di lettura per le istanze database Oracle, consulta Utilizzo di repliche di lettura per Amazon RDS per Oracle.

Eliminazione del cestino riciclaggio

Quando si rilascia una tabella, il database Oracle non rimuove immediatamente lo spazio di storage. Il database rinomina la tabella inserendola insieme agli eventuali oggetti associati in un cestino riciclaggio. L'eliminazione del cestino riciclaggio rimuove questi elementi e rilascia il relativo spazio di storage.

Per rimuovere l'intero cestino riciclaggio, utilizza la procedura Amazon RDS `rdsadmin.rdsadmin_util.purge_dba_recyclebin`. Tuttavia, questa procedura non può eliminare dal cestino riciclaggio gli oggetti `SYS` e `RDSADMIN`. Se è necessario eliminare questi oggetti, contatta AWS Support.

Nell'esempio seguente viene eliminato l'intero cestino riciclaggio.

```
EXEC rdsadmin.rdsadmin_util.purge_dba_recyclebin;
```

Impostazione dei valori di default visualizzati per la redazione completa

Per modificare i valori di default visualizzati per la redazione completa sull'istanza Oracle Amazon RDS, utilizza la procedura `rdsadmin.rdsadmin_util.dbms_redact_upd_full_rdct_val` di Amazon RDS. Tenere presente che viene creata una policy di redazione con il pacchetto PL/SQL `DBMS_REDACT`, come spiegato nella documentazione del database Oracle. La procedura `dbms_redact_upd_full_rdct_val` specifica i caratteri da visualizzare per i diversi tipi di dati influenzati da una policy esistente.

La procedura `dbms_redact_upd_full_rdct_val` include i seguenti parametri.

Nome del parametro	Tipo di dati	Default	Campo obbligatorio	Descrizione
p_number_val	number	Null	No	Modifica il valore di default per le colonne del tipo di dati NUMBER.
p_binfloat_val	binary_float	Null	No	Modifica il valore di default per le colonne del tipo di dati BINARY_FLOAT .
p_bindouble_val	binary_double	Null	No	Modifica il valore di default per le colonne del tipo di dati BINARY_DOUBLE .
p_char_val	char	Null	No	Modifica il valore di default per le colonne del tipo di dati CHAR.
p_varchar_val	varchar2	Null	No	Modifica il valore di default per le colonne del tipo di dati VARCHAR2.
p_nchar_val	nchar	Null	No	Modifica il valore di default per le colonne del tipo di dati NCHAR.
p_nvarchar_val	nvarchar2	Null	No	Modifica il valore di default per le colonne del tipo di dati NVARCHAR2 .
p_date_val	data	Null	No	Modifica il valore di default per le colonne del tipo di dati DATE.

Nome del parametro	Tipo di dati	Default	Campo obbligatorio	Descrizione
p_ts_val	timestamp	Null	No	Modifica il valore di default per le colonne del tipo di dati TIMESTAMP .
p_tswtz_val	timestamp with time zone	Null	No	Modifica il valore di default per le colonne del tipo di dati TIMESTAMP WITH TIME ZONE.
p_blob_val	blob	Null	No	Modifica il valore di default per le colonne del tipo di dati BLOB.
p_clob_val	clob	Null	No	Modifica il valore di default per le colonne del tipo di dati CLOB.
p_nclob_val	nclob	Null	No	Modifica il valore di default per le colonne del tipo di dati NCLOB.

L'esempio seguente modifica il valore di default redatto in * per il tipo di dati CHAR:

```
EXEC rdsadmin.rdsadmin_util.dbms_redact_upd_full_rdct_val(p_char_val => '*');
```

L'esempio seguente modifica i valori di default redatti per i tipi di dati NUMBER, DATE e CHAR:

```
BEGIN
rdsadmin.rdsadmin_util.dbms_redact_upd_full_rdct_val(
    p_number_val=>1,
    p_date_val=>to_date('1900-01-01','YYYY-MM-DD'),
    p_varchar_val=>'X');
END;
/
```

Dopo aver modificato i valori di default per la redazione completa con la procedura `dbms_redact_upd_full_rdct_val`, riavvia l'istanza database per rendere effettiva la modifica. Per ulteriori informazioni, consulta Riavvio di un'istanza database.

Esecuzione di attività comuni relative ai log per le istanze database Oracle

Di seguito, viene descritto come eseguire determinate attività DBA comuni relative all'accesso alle istanze database Amazon RDS che eseguono Oracle. Per fornire un'esperienza di servizio gestito, Amazon RDS non fornisce accesso shell alle istanze database e limita l'accesso a certe procedure e tabelle di sistema che richiedono privilegi avanzati.

Per ulteriori informazioni, consulta File di log del database Oracle.

Argomenti

- Impostazione accesso forzato
- Impostazione di accesso supplementare
- Cambio dei file di log online
- Aggiunta di log redo online
- Eliminazione di log redo online
- Ridimensionamento di log redo online
- Conservazione dei log redo archiviati
- Accesso ai log di ripristino online e archiviati
- Download dei log di ripristino archiviati da Simple Storage Service (Amazon S3)

Impostazione accesso forzato

In modalità accesso forzato, Oracle registra tutte le modifiche nel database ad eccezione delle modifiche in spazi tabella temporanei e segmenti temporanei (le clausole `NOLOGGING` vengono ignorate). Per ulteriori informazioni, consulta Specificare la modalità FORCE LOGGING (ACCESSO FORZATO) nella documentazione Oracle.

Puoi usare la procedura in Amazon RDS per impostare il logging forzato `rdsadmin.rdsadmin_util.force_logging`. La procedura `force_logging` include i seguenti parametri.

Nome del parametro	Tipo di dati	Default	Sì	Descrizione
p_enable	booleano	true	No	Imposta su true per impostare il database nella modalità accesso forzato, false per rimuovere il database dalla modalità accesso forzato.

L'esempio seguente imposta il database in modalità accesso forzato.

```
EXEC rdsadmin.rdsadmin_util.force_logging(p_enable => true);
```

Impostazione di accesso supplementare

Se abiliti la registrazione supplementare, LogMiner dispone delle informazioni necessarie per supportare le righe concatenate e le tabelle con cluster. Per ulteriori informazioni, consulta la pagina Accesso supplementare nella documentazione Oracle.

L'Oracle Database non abilita l'accesso supplementare come impostazione predefinita.
Puoi usare la procedura in Amazon RDS per abilitare e disabilitare il logging supplementare rdsadmin.rdsadmin_util.alter_supplemental_logging. Per ulteriori informazioni sul modo in cui Amazon RDS gestisce la conservazione dei log delle modifiche archiviati per istanze database Oracle, consulta Conservazione dei log redo archiviati.

La procedura alter_supplemental_logging include i seguenti parametri.

Nome del parametro	Tipo di dati	Default	Campo obbligatorio	Descrizione
p_action	varchar2	—	Sì	'ADD' per aggiungere l'accesso supplementare, 'DROP' per rilasciare l'accesso supplementare.

Nome del parametro	Tipo di dati	Default	Campo obbligatorio	Descrizione
p_type	varchar2	null	No	Tipo di accesso supplementare. I valori validi sono 'ALL', 'FOREIGN KEY', 'PRIMARY KEY', 'UNIQUE' o PROCEDURAL .

L'esempio seguente abilita l'accesso supplementare.

```
begin
    rdsadmin.rdsadmin_util.alter_supplemental_logging(
        p_action => 'ADD');
end;
/
```

L'esempio seguente abilita l'accesso supplementare per tutte le colonne di lunghezza fissa massima.

```
begin
    rdsadmin.rdsadmin_util.alter_supplemental_logging(
        p_action => 'ADD',
        p_type   => 'ALL');
end;
/
```

L'esempio seguente abilita l'accesso supplementare per le colonne chiave primarie.

```
begin
    rdsadmin.rdsadmin_util.alter_supplemental_logging(
        p_action => 'ADD',
        p_type   => 'PRIMARY KEY');
end;
/
```

Cambio dei file di log online

Puoi usare la procedura in Amazon RDS per cambiare file di lo `rdsadmin.rdsadmin_util.switch_logfile`. La procedura `switch_logfile` non ha parametri.

L'esempio seguente cambia i file di log.

```
EXEC rdsadmin.rdsadmin_util.switch_logfile;
```

Aggiunta di log redo online

Un'istanza database Amazon RDS che esegue Oracle inizia con quattro log redo online, di 128 MB ciascuno. Puoi usare la procedura in Amazon RDS per aggiungere ulteriori log red `rdsadmin.rdsadmin_util.add_logfile`.

La procedura `add_logfile` include i seguenti parametri.

> **Note**
>
> I parametri sono si escludono a vicenda.

Nome del parametro	Tipo di dati	Default	Campo obbligatorio	Descrizione
`bytes`	positivo	null	No	Le dimensioni del file di log in byte.
`p_size`	varchar2	—	Sì	Le dimensioni del file di log. Puoi specificare le dimensioni in kilobyte (K), megabyte (M) o gigabyte (G).

Il seguente comando aggiunge un file di log di 100 MB.

```
EXEC rdsadmin.rdsadmin_util.add_logfile(p_size => '100M');
```

Eliminazione di log redo online

Puoi utilizzare la procedura in Amazon RDS per rilasciare i log red `rdsadmin.rdsadmin_util.drop_logfile`. La procedura `drop_logfile` include i seguenti parametri.

Nome del parametro	Tipo di dati	Default	Campo obbligatorio	Descrizione
`grp`	positivo	—	Sì	Il numero di gruppo del log.

L'esempio seguente rilascia il log con il numero di gruppo 3.

```
EXEC rdsadmin.rdsadmin_util.drop_logfile(grp => 3);
```

Puoi solo rilasciare log che hanno uno stato di inutilizzato o inattivo. L'esempio seguente ottiene gli stati dei log.

```
SELECT GROUP#, STATUS FROM V$LOG;

GROUP#     STATUS
---------- ----------------
1          CURRENT
2          INACTIVE
3          INACTIVE
4          UNUSED
```

Ridimensionamento di log redo online

Un'istanza database Amazon RDS che esegue Oracle inizia con quattro log redo online, di 128 MB ciascuno. L'esempio seguente visualizza come si possono utilizzare le procedure Amazon RDS per ridimensionare i log da 128 MB ciascuno a 512 MB ciascuno.

```
/* Query V$LOG to see the logs.          */
```

```
/* You start with 4 logs of 128 MB each. */

SELECT GROUP#, BYTES, STATUS FROM V$LOG;

GROUP#     BYTES      STATUS
---------- ---------- ----------------
1          134217728  INACTIVE
2          134217728  CURRENT
3          134217728  INACTIVE
4          134217728  INACTIVE


/* Add four new logs that are each 512 MB */

EXEC rdsadmin.rdsadmin_util.add_logfile(bytes => 536870912);
EXEC rdsadmin.rdsadmin_util.add_logfile(bytes => 536870912);
EXEC rdsadmin.rdsadmin_util.add_logfile(bytes => 536870912);
EXEC rdsadmin.rdsadmin_util.add_logfile(bytes => 536870912);


/* Query V$LOG to see the logs. */
/* Now there are 8 logs.        */

SELECT GROUP#, BYTES, STATUS FROM V$LOG;

GROUP#     BYTES      STATUS
---------- ---------- ----------------
1          134217728  INACTIVE
2          134217728  CURRENT
3          134217728  INACTIVE
4          134217728  INACTIVE
5          536870912  UNUSED
6          536870912  UNUSED
7          536870912  UNUSED
8          536870912  UNUSED


/* Drop each inactive log using the group number. */

EXEC rdsadmin.rdsadmin_util.drop_logfile(grp => 1);
EXEC rdsadmin.rdsadmin_util.drop_logfile(grp => 3);
EXEC rdsadmin.rdsadmin_util.drop_logfile(grp => 4);
```

```
/* Query V$LOG to see the logs. */
/* Now there are 5 logs.        */

select GROUP#, BYTES, STATUS from V$LOG;

GROUP#     BYTES      STATUS
---------- ---------- ----------------
2          134217728  CURRENT
5          536870912  UNUSED
6          536870912  UNUSED
7          536870912  UNUSED
8          536870912  UNUSED


/* Switch logs so that group 2 is no longer current. */

EXEC rdsadmin.rdsadmin_util.switch_logfile;


/* Query V$LOG to see the logs.        */
/* Now one of the new logs is current. */

SQL>SELECT GROUP#, BYTES, STATUS FROM V$LOG;

GROUP#     BYTES      STATUS
---------- ---------- ----------------
2          134217728  ACTIVE
5          536870912  CURRENT
6          536870912  UNUSED
7          536870912  UNUSED
8          536870912  UNUSED


/* If the status of log 2 is still "ACTIVE", issue a checkpoint to clear it to
 "INACTIVE".  */

EXEC rdsadmin.rdsadmin_util.checkpoint;


/* Query V$LOG to see the logs.            */
/* Now the final original log is inactive. */

select GROUP#, BYTES, STATUS from V$LOG;
```

```
GROUP#     BYTES      STATUS
---------- ---------- ----------------
2          134217728  INACTIVE
5          536870912  CURRENT
6          536870912  UNUSED
7          536870912  UNUSED
8          536870912  UNUSED


# Drop the final inactive log.

EXEC rdsadmin.rdsadmin_util.drop_logfile(grp => 2);


/* Query V$LOG to see the logs.    */
/* Now there are four 512 MB logs. */

SELECT GROUP#, BYTES, STATUS FROM V$LOG;

GROUP#     BYTES      STATUS
---------- ---------- ----------------
5          536870912  CURRENT
6          536870912  UNUSED
7          536870912  UNUSED
8          536870912  UNUSED
```

Conservazione dei log redo archiviati

Puoi conservare i log di ripristino archiviati localmente nell'istanza database per utilizzarli con prodotti come Oracle LogMiner (`DBMS_LOGMNR`). Dopo aver conservato i log redo, puoi utilizzare LogMiner per analizzare i log. Per ulteriori informazioni, consulta Utilizzo di LogMiner per analizzare i file di log redo nella documentazione Oracle.

Puoi usare la procedura in Amazon RDS per mantenere i log redo archiviat `rdsadmin.rdsadmin_util.set_configuration`. La procedura `set_configuration` include i seguenti parametri.

Nome del parametro	Tipo di dati	Default	Campo obbligatorio	Descrizione
`name`	varchar	—	Sì	Il nome della configurazione da aggiornare.
`value`	varchar	—	Sì	Il valore per la configurazione.

L'esempio seguente conserva 24 ore di log redo.

```
begin
    rdsadmin.rdsadmin_util.set_configuration(
        name  => 'archivelog retention hours',
        value => '24');
end;
/
commit;
```

> **Note**
>
> La conferma è necessaria per rendere effettiva la modifica.

Puoi utilizzare la procedura Amazon RDS per visualizzare quanto a lungo i log redo archiviati vengono conservati per l'istanza databas `rdsadmin.rdsadmin_util.show_configuration`.

L'esempio seguente mostra il tempo di conservazione dei log.

```
set serveroutput on
EXEC rdsadmin.rdsadmin_util.show_configuration;
```

L'output indica l'impostazione corrente per `archivelog retention hours`. L'output seguente indica che i log redo vengono mantenuti per 48 ore.

```
NAME:archivelog retention hours
```

```
VALUE:48
DESCRIPTION:ArchiveLog expiration specifies the duration in hours before archive/redo
 log files are automatically deleted.
```

Poiché i log redo archiviati vengono conservati nell'istanza database, assicurati che l'istanza database abbia abbastanza storage allocato per i log conservati. Per determinare quanto spazio l'istanza database ha utilizzato nelle ultime X ore, puoi eseguire la query seguente, sostituendo X con il numero di ore.

```
SELECT SUM(BLOCKS * BLOCK_SIZE) bytes
  FROM V$ARCHIVED_LOG
 WHERE FIRST_TIME >= SYSDATE-(X/24) AND DEST_ID=1;
```

I log di ripristino archiviati vengono generati solo se il tempo di conservazione del backup dell'istanza database è superiore a zero. Per impostazione predefinita, il tempo di conservazione del backup è maggiore di zero.

Alla scadenza del periodo di conservazione dei log archiviati, RDS per Oracle rimuove i log di ripristino archiviati dall'istanza database. Per supportare il ripristino dell'istanza del DB a un punto temporale specifico, Amazon RDS conserva i log di ripristino archiviati al di fuori dell'istanza database in base al tempo di conservazione del backup. Per modificare il tempo di conservazione del backup, consulta Modifica di un'istanza database Amazon RDS.

> **Note**
>
> In alcuni casi, si potrebbe utilizzare JDBC su Linux per scaricare i log redo archiviati e riscontrare periodi di latenza e ripristini di connessione lunghi. In questi casi i problemi potrebbero dipendere dall'impostazione predefinita del generatore di numeri casuali nel client Java. Consigliamo di impostare i driver JDBC in modo che utilizzino un generatore di numero casuale senza blocchi.

Accesso ai log di ripristino online e archiviati

Si consiglia di accedere ai file di log redo online e archiviati per il mining con strumenti esterni quali GoldenGate, Attunity, Informatica e altri. Per accedere a questi file, effettuare le operazioni seguenti:

1. Creare oggetti di directory che forniscono l'accesso di sola lettura ai percorsi fisici dei file.

Uso di `rdsadmin.rdsadmin_master_util.create_archivelog_dir` e `rdsadmin.rdsadmin_master_util.create_onlinelog_dir`.

2. Leggere i file utilizzando PL/SQL.

   È possibile leggere i file utilizzando PL/SQL. Per ulteriori informazioni sulla lettura di file dagli oggetti di directory, consulta Generazione di un elenco dei file in una directory di istanze database e Lettura dei file in una directory di istanze database.

L'accesso ai log delle transazioni è supportato per le versioni seguenti:

- Oracle Database 21c
- Oracle Database 19c
- Oracle Database 12c Release 2 (12.2.0.1)
- Oracle Database 12c Release 1 (12.1)

Il seguente codice crea delle directory che forniscono accesso di sola lettura ai file di log redo online e archiviati:

> **Important**
>
> Questo codice revoca anche il privilegio `DROP ANY DIRECTORY`.

```
EXEC rdsadmin.rdsadmin_master_util.create_archivelog_dir;
EXEC rdsadmin.rdsadmin_master_util.create_onlinelog_dir;
```

Il seguente codice rilascia directory per i file di log redo online e archiviati.

```
EXEC rdsadmin.rdsadmin_master_util.drop_archivelog_dir;
EXEC rdsadmin.rdsadmin_master_util.drop_onlinelog_dir;
```

Il seguente codice concede e revoca il privilegio `DROP ANY DIRECTORY`.

```
EXEC rdsadmin.rdsadmin_master_util.revoke_drop_any_directory;
EXEC rdsadmin.rdsadmin_master_util.grant_drop_any_directory;
```

Download dei log di ripristino archiviati da Simple Storage Service (Amazon S3)

È possibile scaricare i log di ripristino archiviati nell'istanza database utilizzando il pacchetto `rdsadmin.rdsadmin_archive_log_download`. Se i log di ripristino archiviati non sono più presenti nell'istanza database, sarà possibile scaricarli di nuovo da Simple Storage Service (Amazon S3). Quindi è possibile estrarre i log o utilizzarli per recuperare o replicare il database.

> **Note**
>
> Non è possibile scaricare log redo archiviati sulle istanze di replica di lettura.

Download dei log di ripristino archiviati: passaggi di base

La disponibilità dei log di ripristino archiviati dipende dalle seguenti policy di conservazione:

- Policy di conservazione dei backup: i log all'interno di questa politica sono disponibili in Simple Storage Service (Amazon S3). I log esterni a questa policy vengono rimossi.
- Policy di conservazione dei log archiviati: i log all'interno di questa policy sono disponibili nell'istanza database. I log esterni a questa policy vengono rimossi.

Se i log non sono presenti nell'istanza ma sono protetti dal tempo di conservazione del backup, utilizza `rdsadmin.rdsadmin_archive_log_download` per scaricarli di nuovo. RDS per Oracle salva i log nella directory `/rdsdbdata/log/arch` nell'istanza database.

Download dei log di ripristino archiviati da Simple Storage Service (Amazon S3)

1. Configura il periodo di conservazione per assicurarti che i log redo archiviati scaricati vengano mantenuti per la durata necessaria. Assicurati di `COMMIT` la modifica.

   RDS mantiene i log scaricati in base alla policy di conservazione dei log archiviati, a partire dal momento in cui i log sono stati scaricati. Per informazioni su come impostare la policy di conservazione, consulta Conservazione dei log redo archiviati.
2. Attendere fino a 5 minuti per rendere effettiva la modifica della policy di conservazione dei log archiviati.
3. Download dei log di ripristino archiviati da Simple Storage Service (Amazon S3) tramite `rdsadmin.rdsadmin_archive_log_download`.

Per ulteriori informazioni, consulta Download di un singolo log di ripristino archiviato e Download di una serie di log di ripristino archiviati.

> **Note**
>
> RDS controlla automaticamente lo spazio di archiviazione disponibile prima del download. Se i log richiesti consumano un'alta percentuale di spazio, viene visualizzato un avviso.

4. Conferma che i log siano stati scaricati correttamente da Simple Storage Service (Amazon S3).

   È possibile visualizzare lo stato dell'attività di download in un file bdump. I file bdump hanno il percorso `/rdsdbdata/log/trace/dbtask-task-id.log`. Nel passaggio di download precedente, si esegue una istruzione `SELECT` che restituisce l'ID attività in un tipo di dati `VARCHAR2`. Per maggiori informazioni, consulta gli esempi simili in Monitoraggio dello stato di un file transfer.

Download di un singolo log di ripristino archiviato

Per scaricare un singolo log di ripristino archiviato nella directory `/rdsdbdata/log/arch` utilizza `rdsadmin.rdsadmin_archive_log_download.download_log_with_seqnum`. Questa procedura include i seguenti parametri.

Nome del parametro	Tipo di dati	Default	Campo obbligatorio	Descrizione
`seqnum`	numero	—	Sì	Il numero di sequenza del log di ripristino archiviato.

L'esempio seguente scarica il log con il numero di sequenza 20.

```
SELECT rdsadmin.rdsadmin_archive_log_download.download_log_with_seqnum(seqnum => 20)
       AS TASK_ID
FROM   DUAL;
```

Download di una serie di log di ripristino archiviati

Per scaricare un singolo log di ripristino archiviato nella directory `/rdsdbdata/log/arch` utilizza `download_logs_in_seqnum_range`. Il download è limitato a 300 registri per richiesta. La procedura `download_logs_in_seqnum_range` include i seguenti parametri.

Nome del parametro	Tipo di dati	Default	Campo obbligatorio	Descrizione
`start_seq`	numero	—	Sì	Il numero di sequenza iniziale per la serie.
`end_seq`	numero	—	Sì	Il numero di sequenza finale per la serie.

L'esempio seguente scarica il log dalla sequenza 50 a 100.

```
SELECT rdsadmin.rdsadmin_archive_log_download.download_logs_in_seqnum_range(start_seq
 => 50, end_seq => 100)
       AS TASK_ID
FROM   DUAL;
```

Esecuzione di attività RMAN comuni per le istanze database Oracle

Nella seguente sezione viene illustrato come eseguire attività DBA Oracle Recovery Manager (RMAN) sulle istanze database Amazon RDS che eseguono Oracle. Per offrire un'esperienza di servizio gestito, Amazon RDS non fornisce accesso shell alle istanze database. Limita anche l'accesso ad alcune procedure di sistema e tabelle che richiedono privilegi avanzati.

Utilizza il pacchetto Amazon RDS `rdsadmin.rdsadmin_rman_util` per eseguire backup RMAN del database Amazon RDS per Oracle su disco. Il pacchetto `rdsadmin.rdsadmin_rman_util` supporta backup completi e incrementali dei file di database, backup degli spazi tabella e backup dei log redo archiviati.

Al termine di un backup RMAN, è possibile copiare i file di backup dall'host dell'istanza database Amazon RDS for Oracle. Tale operazione può essere eseguita per il ripristino su un host diverso da RDS o per uno storage a lungo termine dei backup. Ad esempio, i file di backup possono essere

copiati in un bucket Simple Storage Service (Amazon S3). Per ulteriori informazioni, consulta l'utilizzo di Integrazione Amazon S3.

I file di backup dei backup RMAN rimangono nell'host dell'istanza database Amazon RDS fino a quando non vengono rimossi manualmente. Per rimuovere i file da una directory, è possibile utilizzare la procedura Oracle `UTL_FILE.FREMOVE`. Per ulteriori informazioni, consulta la sezione relativa alla Procedura FREMOVE nella documentazione del database Oracle.

Non puoi utilizzare l'RMAN per ripristinare RDS per le istanze database Oracle. Tuttavia, puoi utilizzare RMAN per ripristinare un backup su un'istanza on-premise o Amazon EC2. Per ulteriori informazioni, consulta l'articolo del blog Restore an Amazon RDS for Oracle instance to a self-managed instance.

> **Note**
>
> Per il backup e il ripristino su un'altra istanza database Amazon RDS for Oracle, si può continuare a utilizzare le funzionalità di backup e ripristino di Amazon RDS. Per ulteriori informazioni, consulta Backup, ripristino ed esportazione dei dati.

Argomenti

Prerequisiti per i backup RMAN

Prima di eseguire il backup del database utilizzando il pacchetto `rdsadmin.rdsadmin_rman_util`, assicurati di soddisfare i seguenti prerequisiti:

- Assicurati che il tuo database RDS per Oracle sia in modalità `ARCHIVELOG`. Per abilitare questa modalità, imposta il periodo di conservazione del backup su un valore diverso da zero.
- Se si esegue il backup di log redo archiviati o un backup completo o incrementale che include log redo archiviati, e quando si esegue il backup del database, accertarsi che la conservazione dei log redo sia impostata su un valore diverso da zero. I log redo archiviati sono necessari per rendere i file del database coerenti durante il ripristino. Per ulteriori informazioni, consulta Conservazione dei log redo archiviati.
- Assicurati che l'istanza database disponga di spazio libero sufficiente per contenere i backup. Quando esegui il backup del database, specifica un oggetto della directory Oracle come un parametro nella chiamata di procedura. RMAN inserisce i file nella directory specificata. Si possono utilizzare directory predefinite, come `DATA_PUMP_DIR`, o crearne di nuove. Per ulteriori informazioni, consulta Creazione ed eliminazione di directory nello spazio di archiviazione dati principale.

  È possibile monitorare lo spazio libero corrente in un'istanza di RDS for Oracle utilizzando la CloudWatch metrica`FreeStorageSpace`. È opportuno che lo spazio libero superi la dimensione corrente del database, sebbene RMAN esegua il backup solo dei blocchi formattati e supporti la compressione.

Parametri comuni per le procedure RMAN

Per eseguire attività con RMAN, si possono utilizzare le procedure del pacchetto Amazon RDS `rdsadmin.rdsadmin_rman_util`. Nelle procedure del pacchetto ci sono diversi parametri comuni. Il pacchetto presenta i seguenti parametri comuni.

Nome del parametro	Tipo di dati	Valori validi	Defaul	Campo obbliga rio	Descrizione
`p_directory_name`	varchar	Un nome valido della directory	—	Sì	Il nome della directory che conterrà i file di backup.

Nome del parametro	Tipo di dati	Valori validi	Defaul	Campo obbliga rio	Descrizione
		del database.			
`p_label`	varchar	`a-z`, `A-Z`, `0-9`, `'_'`, `'-'`, `'.'`	—	No	Una stringa univoca, inclusa nei nomi dei file di backup. **Note** Ha un limite di 30 caratteri.
`p_owner`	varchar	Un proprietario valido della directory specifica ta in `p_directo ry_name` .	—	Sì	Il proprietario della directory che conterrà i file di backup.

Nome del parametro	Tipo di dati	Valori validi	Defaul	Campo obbliga rio	Descrizione
`p_tag`	varchar	`a-z`, `A-Z`, `0-9`, `'_'`, `'-'`, `'.'`	NULL	No	Una stringa che può essere utilizzata per distinguere tra i backup per indicare lo scopo o l'utilizzo dei backup, ad esempio backup a livello giornaliero, settimanale o incrementale. Ha un limite di 30 caratteri. Il tag non rileva la distinzione tra maiuscole e minuscole. I tag sono sempre memorizzati in maiuscolo, indipendentemente dalle maiuscole e minuscole utilizzate durante l'inserimento. I tag non devono essere univoci, quindi più backup possono avere lo stesso tag. Se non specifichi un tag, RMAN assegna automaticamente un tag di default utilizzando il formato `TAG`*`YYYYMMDDTHHMMSS`* , dove *`YYYY`* è l'anno, *`MM`* è il mese, *`DD`* è il giorno, *`HH`* è l'ora (nel formato 24 ore), *`MM`* sono i minuti e *`SS`* sono i secondi. La data e l'ora si riferiscono a quando RMAN ha avviato il backup. Ad esempio, un backup potrebbe ricevere un tag `TAG201909 27T214517` per un backup avviato il 27/09/2019 alle 21:45:17.

Nome del parametro	Tipo di dati	Valori validi	Defaul	Campo obbliga rio	Descrizione
					Il parametro `p_tag` è supportato nelle seguenti versioni del motore database Amazon RDS for Oracle: • Oracle Database 21c (21.0.0) • Oracle Database 19c (19.0.0), usando le versioni 19.0.0.0.ru-2021-10.rur-2021-10.r1 o successive • Oracle Database 12c Release 2 (12.2) usando la versione 12.2.0.1.ru-2021-10.rur-2021-10.r1 o successive • Oracle Database 12c Release 1 (12.1) usando la versione 12.1.0.2.V26 o successive
`p_compress`	boolean	`TRUE`, `FALSE`	`FALSE`	No	Specificare `TRUE` per abilitare la compressione BASIC del backup. Specificare `FALSE` per disabilitare la compressione BASIC del backup.

Nome del parametro	Tipo di dati	Valori validi	Defaul	Campo obbliga rio	Descrizione
`p_include_archive_logs`	boolear	`TRUE`, `FALSE`	`FALSE`	No	Specificare `TRUE` per includere i log redo archiviati nel backup. Specificare `FALSE` per escludere i log redo archiviati dal backup. Se si includono log redo archiviati nel backup, la retention va impostata su un valore almeno pari a un'ora tramite la procedura `rdsadmin.rdsadmin_util.set_configuration` . Inoltre, occorre chiamare la procedura `rdsadmin.rdsadmin_rman_util.crosscheck_archivelog` subito prima di eseguire il backup. In caso contrario, il backup potrebbe non andare a buon fine perché mancano i file di log redo archiviati che sono stati eliminati dalle procedure di gestione di Amazon RDS.
`p_include_controlfile`	boolear	`TRUE`, `FALSE`	`FALSE`	No	Specificare `TRUE` per includere il file di controllo nel backup. Specificare `FALSE` per escludere il file di controllo dal backup.
`p_optimize`	boolear	`TRUE`, `FALSE`	`TRUE`	No	Specificare `TRUE` per abilitare l'ottimizzazione del backup e ridurne le dimensioni, se sono inclusi log redo archiviati. Specificare `FALSE` per disabilitare l'ottimizzazione del backup.

Nome del parametro	Tipo di dati	Valori validi	Defaul	Campo obbliga rio	Descrizione
p_parallel	numero	Un intero valido tra 1 e 254 per Oracle Database Enterpris e Edition (EE) 1 per altre edizioni Oracle Database	1	No	Numero di canali.
p_rman_to_dbms_output	boolear	TRUE, FALSE	FALSE	No	Quando TRUE, l'output RMAN viene inviato al pacchetto DBMS_OUTPUT oltre a un file nella directory BDUMP. In SQL*Plus, utilizza SET SERVEROUT PUT ON per visualizzare l'output. Quando FALSE, l'output RMAN viene solo inviato a un file nella directory BDUMP.
p_section_size_mb	numero	Intero valido	NULL	No	Le dimensioni della sezione in megabyte (MB). Convalida in parallelo dividendo ogni file nelle dimensioni specificate della sezione. Quando NULL, il parametro è ignorato.

Nome del parametro	Tipo di dati	Valori validi	Defaul	Campo obbliga rio	Descrizione
`p_validation_type`	varchar	`'PHYSICAL'` , `'PHYSICAL+LOGICAL'`	`'PHYS'`	No	Livello di rilevamento di danneggiamenti. Specifica `'PHYSICAL'` per cercare i danneggiamenti fisici. Un esempio di danneggiamento fisico è un blocco con una mancata corrispondenza nell'intestazione e piè di pagina. Specifica `'PHYSICAL+LOGICAL'` per cercare le incoerenze logiche oltre ai danneggiamenti fisici. Un esempio di danneggiamento logico è un blocco danneggiato.

Convalida dei file di database in RDS per Oracle

Puoi utilizzare il pacchetto Amazon RDS `rdsadmin.rdsadmin_rman_util` per convalidare i file di database Amazon RDS for Oracle, come file di dati, tablespace, file di controllo e file dei parametri del server (SPFiles).

Per ulteriori informazioni sulla convalida RMAN, consulta Convalida di file di dati e backup di database e VALIDATE nella documentazione Oracle.

Argomenti

- Convalida di un database
- Convalida di un database del tenant
- Convalida di uno spazio di tabella
- Convalida di un file di controllo
- Convalida di un file SPFILE
- Convalida di un file di dati Oracle

Convalida di un database

Per convalidare tutti i file pertinenti utilizzati da un database Oracle in RDS for Oracle, utilizza la procedura Amazon RDS. `rdsadmin.rdsadmin_rman_util.validate_database`

La procedura utilizza i seguenti parametri comuni per le attività RMAN:

- `p_validation_type`
- `p_parallel`
- `p_section_size_mb`
- `p_rman_to_dbms_output`

Per ulteriori informazioni, consulta Parametri comuni per le procedure RMAN.

L'esempio seguente convalida il database utilizzando i valori predefiniti per i parametri.

```
EXEC rdsadmin.rdsadmin_rman_util.validate_database;
```

L'esempio seguente convalida il database utilizzando i valori specificati per i parametri.

```
BEGIN
    rdsadmin.rdsadmin_rman_util.validate_database(
        p_validation_type     => 'PHYSICAL+LOGICAL',
        p_parallel            => 4,
        p_section_size_mb     => 10,
        p_rman_to_dbms_output => FALSE);
END;
/
```

Quando il parametro `p_rman_to_dbms_output` è impostato su `FALSE`, l'output RMAN viene scritto in un file nella directory `BDUMP`.

Per visualizzare i file nella directory `BDUMP`, esegui la seguente istruzione `SELECT`.

```
SELECT * FROM table(rdsadmin.rds_file_util.listdir('BDUMP')) order by mtime;
```

Per visualizzare i contenuti di un file nella directory `BDUMP`, esegui la seguente istruzione `SELECT`.

```
SELECT text FROM table(rdsadmin.rds_file_util.read_text_file('BDUMP','rds-rman-
validate-nnn.txt'));
```

Sostituisci il nome del file con il nome del file che desideri visualizzare.

Convalida di un database del tenant

Per convalidare i file di dati del database del tenant in un database container (CDB), utilizza la procedura Amazon RDS `rdsadmin.rdsadmin_rman_util.validate_tenant`.

La procedura si applica solo al database del tenant corrente e utilizza i seguenti parametri comuni per le attività RMAN:

- `p_validation_type`
- `p_parallel`
- `p_section_size_mb`
- `p_rman_to_dbms_output`

Per ulteriori informazioni, consulta Parametri comuni per le procedure RMAN. Questa procedura è supportata nelle seguenti versioni del motore di database:

- Oracle Database 21c (21.0.0) CDB
- Oracle Database 19c (19.0.0) CDB

Il seguente esempio convalida il database del tenant corrente utilizzando i valori predefiniti per i parametri.

```
EXEC rdsadmin.rdsadmin_rman_util.validate_tenant;
```

Il seguente esempio convalida il database del tenant corrente utilizzando i valori predefiniti per i parametri.

```
BEGIN
    rdsadmin.rdsadmin_rman_util.validate_tenant(
        p_validation_type     => 'PHYSICAL+LOGICAL',
        p_parallel            => 4,
        p_section_size_mb     => 10,
        p_rman_to_dbms_output => FALSE);
END;
/
```

Quando il parametro `p_rman_to_dbms_output` è impostato su `FALSE`, l'output RMAN viene scritto in un file nella directory `BDUMP`.

Per visualizzare i file nella directory `BDUMP`, esegui la seguente istruzione `SELECT`.

```
SELECT * FROM table(rdsadmin.rds_file_util.listdir('BDUMP')) order by mtime;
```

Per visualizzare i contenuti di un file nella directory `BDUMP`, esegui la seguente istruzione `SELECT`.

```
SELECT text FROM table(rdsadmin.rds_file_util.read_text_file('BDUMP','rds-rman-validate-nnn.txt'));
```

Sostituisci il nome del file con il nome del file che desideri visualizzare.

Convalida di uno spazio di tabella

Puoi usare la procedura in Amazon RDS per convalidare i file associati a uno spazio tabell `rdsadmin.rdsadmin_rman_util.validate_tablespace`.

La procedura utilizza i seguenti parametri comuni per le attività RMAN:

- `p_validation_type`
- `p_parallel`
- `p_section_size_mb`
- `p_rman_to_dbms_output`

Per ulteriori informazioni, consulta Parametri comuni per le procedure RMAN.

Questa procedura utilizza anche il seguente parametro aggiuntivo.

Nome del parametro	Tipo di dati	Valori validi	Default	Campo obbligatorio	Descrizione
`p_tablespace_name`	varchar2	Un nome spazio tabella valido	—	Sì	Il nome dello spazio tabella.

Convalida di un file di controllo

Per convalidare solo il file di controllo usato da un'istanza database Oracle Amazon RDS, utilizzare la procedura Amazon RDS `rdsadmin.rdsadmin_rman_util.validate_current_controlfile`.

La procedura utilizza il seguente parametro comune per le attività RMAN:

- `p_validation_type`
- `p_rman_to_dbms_output`

Per ulteriori informazioni, consulta Parametri comuni per le procedure RMAN.

Convalida di un file SPFILE

Per convalidare solo il file dei parametri server (SPFILE) usato da un'istanza database Oracle Amazon RDS, utilizzare la procedura Amazon RDS `rdsadmin.rdsadmin_rman_util.validate_spfile`.

La procedura utilizza il seguente parametro comune per le attività RMAN:

- `p_validation_type`
- `p_rman_to_dbms_output`

Per ulteriori informazioni, consulta Parametri comuni per le procedure RMAN.

Convalida di un file di dati Oracle

Puoi usare la procedura in Amazon RDS per convalidare un file di dat `rdsadmin.rdsadmin_rman_util.validate_datafile`.

La procedura utilizza i seguenti parametri comuni per le attività RMAN:

- `p_validation_type`
- `p_parallel`
- `p_section_size_mb`
- `p_rman_to_dbms_output`

Per ulteriori informazioni, consulta Parametri comuni per le procedure RMAN.

Questa procedura utilizza anche i seguenti parametri aggiuntivi.

Nome del parametro	Tipo di dati	Valori validi	Default	Campo obbligatorio	Descrizione
`p_datafile`	varchar2	Un numero ID di file di dati valido o un nome di file di dati valido incluso il percorso completo	—	Sì	Il numero ID di file di dati (da `v$datafile.file#`) o il nome completo di file di dati incluso il percorso (da `v$datafile.name`).
`p_from_block`	numero	Intero valido	`NULL`	No	Il numero del blocco dove la convalida inizia con i file di dati. Se questo numero è `NULL`, si utilizza 1.
`p_to_block`	numero	Intero valido	`NULL`	No	Il numero del blocco dove la convalida finisce con i file di dati. Se questo numero è `NULL`, viene utilizzato il blocco massimo nel file di dati.

Abilitazione e disabilitazione del monitoraggio delle modifiche dei blocchi.

Il rilevamento delle modifiche di blocco registra i blocchi modificati in un file di monitoraggio. Questa tecnica può migliorare le prestazioni dei backup incrementali RMAN. Per ulteriori informazioni,

consulta Utilizzo del rilevamento delle modifiche di blocco per migliorare le prestazioni di backup incrementali nella documentazione di Oracle Database.

Le funzionalità RMAN non sono supportate in una replica di lettura. Tuttavia, come parte della strategia per la disponibilità elevata, puoi scegliere di abilitare il tracciamento dei blocchi in una replica di sola lettura utilizzando la procedura `rdsadmin.rdsadmin_rman_util.enable_block_change_tracking`. Se promuovi questa replica di sola lettura a un'istanza database di origine, il tracciamento delle modifiche dei blocchi viene abilitato per la nuova istanza di origine. Pertanto, la tua istanza può trarre vantaggio da backup incrementali rapidi.

Le procedure di rilevamento delle modifiche di blocco sono supportate in Enterprise Edition solo per le seguenti versioni del motore del database:

- Oracle Database 21c (21.0.0)
- Oracle Database 19c (19.0.0)
- Oracle Database 12c Release 2 (12.2) usando 12.2.0.1.ru-2019-01.rur-2019-01.r1 o versioni successive (obsoleto)
- Oracle Database 12c Release 1 (12.1) usando 12.1.0.2.v15 o versioni successive (obsoleto)

> **Note**
>
> In un CDB a tenant singolo funzionano le seguenti operazioni, ma nessun meccanismo visibile al cliente può rilevare lo stato corrente delle operazioni. Consulta anche Limitazioni per i CDB RDS per Oracle.

Per abilitare il rilevamento delle modifiche di blocco per un'istanza database, consulta la Amazon RDS procedura `rdsadmin.rdsadmin_rman_util.enable_block_change_tracking`. Per disattivare il rilevamento delle modifiche di blocco, utilizza `disable_block_change_tracking`. Queste procedure non hanno parametri.

Per determinare se il monitoraggio delle modifiche dei blocchi è abilitato per l'istanza database, eseguire la seguente query.

```
SELECT STATUS, FILENAME FROM V$BLOCK_CHANGE_TRACKING;
```

Il seguente esempio abilita il monitoraggio delle modifiche dei blocchi per un'istanza database.

```
EXEC rdsadmin.rdsadmin_rman_util.enable_block_change_tracking;
```

Il seguente esempio disabilita il monitoraggio delle modifiche dei blocchi per un'istanza database.

```
EXEC rdsadmin.rdsadmin_rman_util.disable_block_change_tracking;
```

Controllo incrociato dei log redo archiviati

Si può eseguire il controllo incrociato dei log redo archiviati utilizzando la procedura Amazon RDS `rdsadmin.rdsadmin_rman_util.crosscheck_archivelog`.

Questa procedura serve a eseguire il controllo incrociato dei log redo archiviati registrati nel file di controllo e opzionalmente per eliminare i record di log scaduti. Quando RMAN esegue un backup, crea un record nel file di controllo. Col tempo, questi registri aumentano le dimensioni del file di controllo. Si consiglia di rimuovere periodicamente i record scaduti.

> **Note**
>
> I backup Amazon RDS standard non utilizzano RMAN, quindi non creano registri nel file di controllo.

La procedura utilizza il parametro comune `p_rman_to_dbms_output` per le attività RMAN.

Per ulteriori informazioni, consulta Parametri comuni per le procedure RMAN.

Questa procedura utilizza anche il seguente parametro aggiuntivo.

Nome del parametro	Tipo di dati	Valori validi	Default	Campo obbligatorio	Descrizione
`p_delete_expired`	booleano	`TRUE`, `FALSE`	`TRUE`	No	Se `TRUE`, si eliminano dal file di controllo i record di log redo archiviati scaduti.

Nome del parametro	Tipo di dati	Valori validi	Default	Campo obbligatorio	Descrizione
					Se FALSE, si conservano nel file di controllo i record di log redo archiviati scaduti.

Questa procedura è supportata nelle seguenti versioni del motore database Amazon RDS for Oracle:

- Oracle Database 21c (21.0.0)
- Oracle Database 19c (19.0.0)
- Oracle Database 12c Release 2 (12.2) usando 12.2.0.1.ru-2019-01.rur-2019-01.r1 o versioni successive
- Oracle Database 12c Release 1 (12.1) usando 12.1.0.2.v15 o versioni successive

Nell'esempio seguente i record di log redo archiviati nel file di controllo file vengono contrassegni come scaduti, ma non vengono eliminati.

```
BEGIN
    rdsadmin.rdsadmin_rman_util.crosscheck_archivelog(
        p_delete_expired      => FALSE,
        p_rman_to_dbms_output => FALSE);
END;
/
```

Il seguente esempio elimina i record di log redo archiviati scaduti dal file di controllo.

```
BEGIN
    rdsadmin.rdsadmin_rman_util.crosscheck_archivelog(
        p_delete_expired      => TRUE,
        p_rman_to_dbms_output => FALSE);
END;
/
```

Backup dei redo log file archiviati

Si può utilizzare il pacchetto Amazon RDS `rdsadmin.rdsadmin_rman_util` per eseguire il backup dei log redo archiviati per un'istanza database Amazon RDS Oracle.

Le procedure di backup dei log redo archiviati sono supportate nelle seguenti versioni del motore database Amazon RDS for Oracle:

- Oracle Database 21c (21.0.0)
- Oracle Database 19c (19.0.0)
- Oracle Database 12c Release 2 (12.2) usando 12.2.0.1.ru-2019-01.rur-2019-01.r1 o versioni successive
- Oracle Database 12c Release 1 (12.1), versione 12.1.0.2.v15 o versioni successive

Argomenti

Backup di tutti i log redo archiviati

Per eseguire il backup di tutti i og redo archiviati per un'istanza database Oracle Amazon RDS, utilizzare la procedura in Amazon RDS `rdsadmin.rdsadmin_rman_util.backup_archivelog_all`.

La procedura utilizza i seguenti parametri comuni per le attività RMAN:

- `p_owner`
- `p_directory_name`
- `p_label`
- `p_parallel`
- `p_compress`
- `p_rman_to_dbms_output`

- p_tag

Per ulteriori informazioni, consulta Parametri comuni per le procedure RMAN.

Il seguente esempio esegue il backup di tutti i log redo archiviati per l'istanza database.

```
BEGIN
    rdsadmin.rdsadmin_rman_util.backup_archivelog_all(
        p_owner               => 'SYS',
        p_directory_name      => 'MYDIRECTORY',
        p_parallel            => 4,
        p_tag                 => 'MY_LOG_BACKUP',
        p_rman_to_dbms_output => FALSE);
END;
/
```

Backup di un log redo archiviato in base a un intervallo di date

Per eseguire il backup di log redo specifici archiviati per un'istanza database Oracle Amazon RDS specificando un intervallo di date, utilizzare la procedura in Amazon RDS `rdsadmin.rdsadmin_rman_util.backup_archivelog_date`. L'intervallo di date specifica i log redo archiviati di cui eseguire il backup.

La procedura utilizza i seguenti parametri comuni per le attività RMAN:

- p_owner
- p_directory_name
- p_label
- p_parallel
- p_compress
- p_rman_to_dbms_output
- p_tag

Per ulteriori informazioni, consulta Parametri comuni per le procedure RMAN.

Questa procedura utilizza anche i seguenti parametri aggiuntivi.

Nome del parametro	Tipo di dati	Valori validi	Default	Campo obbligatorio	Descrizione
p_from_date	date	Una data compresa tra start_date e next_date di un log redo archiviato esistente su disco. Il valore deve essere uguale o inferiore al valore specificato per p_to_date.	—	Sì	La data di inizio per i backup dei log archiviati.
p_to_date	date	Una data compresa tra start_date e next_date di un log redo archiviato esistente	—	Sì	La data di file per i backup dei log archiviati.

Nome del parametro	Tipo di dati	Valori validi	Default	Campo obbligatorio	Descrizione
		su disco. Il valore deve essere maggiore o uguale al valore specificato per `p_from_date` .			

Il seguente esempio esegue il backup dei log redo archiviati nell'intervallo di date per l'istanza database.

```
BEGIN
    rdsadmin.rdsadmin_rman_util.backup_archivelog_date(
        p_owner                => 'SYS',
        p_directory_name       => 'MYDIRECTORY',
        p_from_date            => '03/01/2019 00:00:00',
        p_to_date              => '03/02/2019 00:00:00',
        p_parallel             => 4,
        p_tag                  => 'MY_LOG_BACKUP',
        p_rman_to_dbms_output => FALSE);
END;
/
```

Backup di un log redo archiviato in base a un intervallo SCN

Per eseguire il backup di log redo specifici archiviati per un'istanza database Oracle Amazon RDS specificando un numero di modifica del sistema (SCN), utilizzare la procedura in Amazon RDS `rdsadmin.rdsadmin_rman_util.backup_archivelog_scn`. L'intervallo SCN specifica i log redo archiviati di cui eseguire il backup.

La procedura utilizza i seguenti parametri comuni per le attività RMAN:

- `p_owner`
- `p_directory_name`
- `p_label`
- `p_parallel`
- `p_compress`
- `p_rman_to_dbms_output`
- `p_tag`

Per ulteriori informazioni, consulta Parametri comuni per le procedure RMAN.

Questa procedura utilizza anche i seguenti parametri aggiuntivi.

Nome del parametro	Tipo di dati	Valori validi	Default	Campo obbligatorio	Descrizione
`p_from_scn`	numero	Un SCN di un log redo archiviato esistente su disco. Il valore deve essere uguale o inferiore al valore specificato per `p_to_scn`.	—	Sì	L'SCN di inizio per i backup dei log archiviati.

Nome del parametro	Tipo di dati	Valori validi	Default	Campo obbligatorio	Descrizione
p_to_scn	numero	Un SCN di un log redo archiviato esistente su disco. Il valore deve essere maggiore o uguale al valore specificato per p_from_scn .	—	Sì	Il SCN di fine per i backup dei log archiviati.

Il seguente esempio esegue il backup dei log redo archiviati nell'intervallo di SCN per l'istanza database.

```
BEGIN
    rdsadmin.rdsadmin_rman_util.backup_archivelog_scn(
        p_owner                => 'SYS',
        p_directory_name       => 'MYDIRECTORY',
        p_from_scn             => 1533835,
        p_to_scn               => 1892447,
        p_parallel             => 4,
        p_tag                  => 'MY_LOG_BACKUP',
        p_rman_to_dbms_output => FALSE);
END;
/
```

Backup di un log redo archiviato in base a un intervallo di numeri di sequenza

Per eseguire il backup di log redo specifici archiviati per un'istanza database Oracle Amazon RDS specificando un intervallo di numeri in sequenza, utilizzare la procedura in Amazon RDS `rdsadmin.rdsadmin_rman_util.backup_archivelog_sequence`. L'intervallo di numeri di sequenza specifica i log redo archiviati di cui eseguire il backup.

La procedura utilizza i seguenti parametri comuni per le attività RMAN:

- `p_owner`
- `p_directory_name`
- `p_label`
- `p_parallel`
- `p_compress`
- `p_rman_to_dbms_output`
- `p_tag`

Per ulteriori informazioni, consulta Parametri comuni per le procedure RMAN.

Questa procedura utilizza anche i seguenti parametri aggiuntivi.

Nome del parametro	Tipo di dati	Valori validi	Default	Campo obbligatorio	Descrizione
`p_from_sequence`	numero	Un numero di sequenza di un log redo archiviato esistente su disco. Il valore deve	—	Sì	Il numero di sequenza iniziale per i backup dei log archiviati.

Nome del parametro	Tipo di dati	Valori validi	Default	Campo obbligatorio	Descrizione
		essere uguale o inferiore al valore specificato per `p_to_sequence` .			
`p_to_sequence`	numero	Un numero di sequenza di un log redo archiviato esistente su disco. Il valore deve essere maggiore o uguale al valore specificato per `p_from_sequence` .	—	Sì	Il numero di sequenza finale per i backup dei log archiviati.

Il seguente esempio esegue il backup dei log redo archiviati nell'intervallo di numeri di sequenza per l'istanza database.

```
BEGIN
```

```
    rdsadmin.rdsadmin_rman_util.backup_archivelog_sequence(
        p_owner               => 'SYS',
        p_directory_name      => 'MYDIRECTORY',
        p_from_sequence       => 11160,
        p_to_sequence         => 11160,
        p_parallel            => 4,
        p_tag                 => 'MY_LOG_BACKUP',
        p_rman_to_dbms_output => FALSE);
END;
/
```

Esecuzione di un backup di database completo

Si può eseguire un backup di tutti i blocchi di file di dati inclusi nel backup utilizzando la procedura Amazon RDS `rdsadmin.rdsadmin_rman_util.backup_database_full`.

La procedura utilizza i seguenti parametri comuni per le attività RMAN:

- `p_owner`
- `p_directory_name`
- `p_label`
- `p_parallel`
- `p_section_size_mb`
- `p_include_archive_logs`
- `p_optimize`
- `p_compress`
- `p_rman_to_dbms_output`
- `p_tag`

Per ulteriori informazioni, consulta Parametri comuni per le procedure RMAN.

Questa procedura è supportata nelle seguenti versioni del motore database Amazon RDS for Oracle:

- Oracle Database 21c (21.0.0)
- Oracle Database 19c (19.0.0)

Il seguente esempio esegue un backup completo dell'istanza database utilizzando i valori specificati per i parametri:

```
BEGIN
    rdsadmin.rdsadmin_rman_util.backup_database_full(
        p_owner               => 'SYS',
        p_directory_name      => 'MYDIRECTORY',
        p_parallel            => 4,
        p_section_size_mb     => 10,
        p_tag                 => 'FULL_DB_BACKUP',
        p_rman_to_dbms_output => FALSE);
END;
/
```

Esecuzione di un backup completo di un database del tenant

È possibile eseguire un backup di tutti i blocchi di dati inclusi in un database del tenant in un database container (CDB). Usa la procedura Amazon RDS `rdsadmin.rdsadmin_rman_util.backup_tenant_full`. La procedura si applica solo al backup del database corrente e utilizza i seguenti parametri comuni per le operazioni RMAN:

- `p_owner`
- `p_directory_name`
- `p_label`
- `p_parallel`
- `p_section_size_mb`
- `p_include_archive_logs`
- `p_optimize`
- `p_compress`
- `p_rman_to_dbms_output`
- `p_tag`

Per ulteriori informazioni, consulta Parametri comuni per le procedure RMAN.

La procedura `rdsadmin_rman_util.backup_tenant_full` è supportata nelle seguenti versioni del motore di database RDS per Oracle:

- Oracle Database 21c (21.0.0) CDB
- Oracle Database 19c (19.0.0) CDB

Il seguente esempio esegue un backup completo del database del tenant corrente utilizzando i valori specificati per i parametri:

```
BEGIN
    rdsadmin.rdsadmin_rman_util.backup_tenant_full(
        p_owner               => 'SYS',
        p_directory_name      => 'MYDIRECTORY',
        p_parallel            => 4,
        p_section_size_mb     => 10,
        p_tag                 => 'FULL_TENANT_DB_BACKUP',
        p_rman_to_dbms_output => FALSE);
END;
/
```

Esecuzione di un backup di database incrementale

Si può eseguire un backup incrementale dell'istanza database utilizzando la procedura Amazon RDS `rdsadmin.rdsadmin_rman_util.backup_database_incremental`.

Per ulteriori informazioni sui backup incrementali, consulta Incremental Backups nella documentazione di Oracle.

La procedura utilizza i seguenti parametri comuni per le attività RMAN:

- `p_owner`
- `p_directory_name`
- `p_label`
- `p_parallel`
- `p_section_size_mb`
- `p_include_archive_logs`
- `p_include_controlfile`
- `p_optimize`
- `p_compress`

- `p_rman_to_dbms_output`
- `p_tag`

Per ulteriori informazioni, consulta Parametri comuni per le procedure RMAN.

Questa procedura è supportata nelle seguenti versioni del motore database Amazon RDS for Oracle:

- Oracle Database 21c (21.0.0)
- Oracle Database 19c (19.0.0)
- Oracle Database 12c Release 2 (12.2) usando 12.2.0.1.ru-2019-01.rur-2019-01.r1 o versioni successive
- Oracle Database 12c Release 1 (12.1) usando 12.1.0.2.v15 o versioni successive

Questa procedura utilizza anche il seguente parametro aggiuntivo.

Nome del parametro	Tipo di dati	Valori validi	Default	Campo obbligatorio	Descrizione
`p_level`	numero	`0`, `1`	`0`	No	Specificare `0` per abilitare un backup incrementale completo. Specificare 1 per abilitare un backup incrementale non cumulativo.

Il seguente esempio esegue un backup incrementale dell'istanza database utilizzando i valori specificati per i parametri:

```
BEGIN
    rdsadmin.rdsadmin_rman_util.backup_database_incremental(
        p_owner              => 'SYS',
        p_directory_name     => 'MYDIRECTORY',
```

```
        p_level               => 1,
        p_parallel            => 4,
        p_section_size_mb     => 10,
        p_tag                 => 'MY_INCREMENTAL_BACKUP',
        p_rman_to_dbms_output => FALSE);
END;
/
```

Esecuzione di un backup incrementale di un database del tenant

È possibile eseguire un backup incrementale del database del tenant corrente nel tuo CDB. Usa la procedura Amazon RDS `rdsadmin.rdsadmin_rman_util.backup_tenant_incremental`.

Per ulteriori informazioni sui backup incrementali, consulta Incremental Backups nella documentazione di Oracle Database.

La procedura si applica solo al database del tenant corrente e utilizza i seguenti parametri comuni per le attività RMAN:

- `p_owner`
- `p_directory_name`
- `p_label`
- `p_parallel`
- `p_section_size_mb`
- `p_include_archive_logs`
- `p_include_controlfile`
- `p_optimize`
- `p_compress`
- `p_rman_to_dbms_output`
- `p_tag`

Per ulteriori informazioni, consulta Parametri comuni per le procedure RMAN.

Questa procedura è supportata nelle seguenti versioni del motore database Amazon RDS for Oracle:

- Oracle Database 21c (21.0.0) CDB

- Oracle Database 19c (19.0.0) CDB

Questa procedura utilizza anche il seguente parametro aggiuntivo.

Nome del parametro	Tipo di dati	Valori validi	Default	Campo obbligatorio	Descrizione
`p_level`	numero	`0`, `1`	`0`	No	Specificare `0` per abilitare un backup incrementale completo. Specificare 1 per abilitare un backup incrementale non cumulativo.

Il seguente esempio esegue un backup incrementale del database del tenant corrente utilizzando i valori specificati per i parametri.

```
BEGIN
    rdsadmin.rdsadmin_rman_util.backup_tenant_incremental(
        p_owner               => 'SYS',
        p_directory_name      => 'MYDIRECTORY',
        p_level               => 1,
        p_parallel            => 4,
        p_section_size_mb     => 10,
        p_tag                 => 'MY_INCREMENTAL_BACKUP',
        p_rman_to_dbms_output => FALSE);
END;
/
```

Backup di uno spazio di tabella

Si può eseguire il backup di uno spazio di tabella utilizzando la procedura `rdsadmin.rdsadmin_rman_util.backup_tablespace` di Amazon RDS.

La procedura utilizza i seguenti parametri comuni per le attività RMAN:

- p_owner
- p_directory_name
- p_label
- p_parallel
- p_section_size_mb
- p_include_archive_logs
- p_include_controlfile
- p_optimize
- p_compress
- p_rman_to_dbms_output
- p_tag

Per ulteriori informazioni, consulta Parametri comuni per le procedure RMAN.

Questa procedura utilizza anche il seguente parametro aggiuntivo.

Nome del parametro	Tipo di dati	Valori validi	Default	Campo obbligatorio	Descrizione
p_tablespace_name	varchar2	Un nome spazio tabella valido.	—	Sì	Il nome dello spazio tabella di cui eseguire il backup.

Questa procedura è supportata nelle seguenti versioni del motore database Amazon RDS for Oracle:

- Oracle Database 21c (21.0.0)
- Oracle Database 19c (19.0.0)
- Oracle Database 12c Release 2 (12.2) usando 12.2.0.1.ru-2019-01.rur-2019-01.r1 o versioni successive
- Oracle Database 12c Release 1 (12.1) usando 12.1.0.2.v15 o versioni successive

Il seguente esempio esegue un backup dello spazio tabella utilizzando i valori specificati per i parametri.

```
BEGIN
    rdsadmin.rdsadmin_rman_util.backup_tablespace(
        p_owner               => 'SYS',
        p_directory_name      => 'MYDIRECTORY',
        p_tablespace_name     => 'MYTABLESPACE',
        p_parallel            => 4,
        p_section_size_mb     => 10,
        p_tag                 => 'MYTABLESPACE_BACKUP',
        p_rman_to_dbms_output => FALSE);
END;
/
```

Backup di un file di controllo

Si può eseguire il backup di un file di controllo utilizzando la procedura `rdsadmin.rdsadmin_rman_util.backup_current_controlfile` di Amazon RDS.

La procedura utilizza i seguenti parametri comuni per le attività RMAN:

- `p_owner`
- `p_directory_name`
- `p_label`
- `p_compress`
- `p_rman_to_dbms_output`
- `p_tag`

Per ulteriori informazioni, consulta Parametri comuni per le procedure RMAN.

Questa procedura è supportata nelle seguenti versioni del motore database Amazon RDS for Oracle:

- Oracle Database 21c (21.0.0)
- Oracle Database 19c (19.0.0)
- Oracle Database 12c Release 2 (12.2) usando 12.2.0.1.ru-2019-01.rur-2019-01.r1 o versioni successive
- Oracle Database 12c Release 1 (12.1) usando 12.1.0.2.v15 o versioni successive

Il seguente esempio esegue un backup di un file di controllo utilizzando i valori specificati per i parametri.

```
BEGIN
    rdsadmin.rdsadmin_rman_util.backup_current_controlfile(
        p_owner               => 'SYS',
        p_directory_name      => 'MYDIRECTORY',
        p_tag                 => 'CONTROL_FILE_BACKUP',
        p_rman_to_dbms_output => FALSE);
END;
/
```

Esecuzione del ripristino dei supporti a blocchi

Puoi ripristinare singoli blocchi di dati, noto come block media recovery, utilizzando le procedure `rdsadmin.rdsadmin_rman_util.recover_datafile_block` Amazon RDS. È possibile utilizzare questa procedura sovraccaricata per ripristinare un singolo blocco di dati o una serie di blocchi di dati.

La procedura utilizza il seguente parametro comune per le attività RMAN:

- `p_rman_to_dbms_output`

Per ulteriori informazioni, consulta Parametri comuni per le procedure RMAN.

Questa procedura utilizza i seguenti parametri aggiuntivi.

Nome del parametro	Tipo di dati	Valori validi	Default	Campo obbligatorio	Descrizione
`p_datafile`	`NUMBER`	Un numero ID valido per il file di dati.	—	Sì	Il file di dati contenente i blocchi danneggiati. Specificate il file di dati in uno dei seguenti modi: • Il numero ID del file di dati, che si trova in V`$DATAFILE.FILE#`

Nome del parametro	Tipo di dati	Valori validi	Default	Campo obbligatorio	Descrizione
					• Il nome completo del file di dati, incluso il percorso, che si trova in `V$DATAFILE.NAME`
`p_block`	`NUMBER`	Un numero intero valido.	—	Sì	Il numero di un singolo blocco da recuperare. I seguenti parametri si escludono a vicenda: • `p_block` • `p_from_block` e `p_to_block`
`p_from_block`	`NUMBER`	Un numero intero valido.	—	Sì	Il primo numero di blocco in un intervallo di blocchi da recuperare. I seguenti parametri si escludono a vicenda: • `p_block` • `p_from_block` e `p_to_block`

Nome del parametro	Tipo di dati	Valori validi	Default	Campo obbligatorio	Descrizione
p_to_block	NUMBER	Un numero intero valido.	—	Sì	L'ultimo numero di blocco in un intervallo di blocchi da recuperare. I seguenti parametri si escludono a vicenda: • p_block • p_from_block e p_to_block

Questa procedura è supportata nelle seguenti versioni del motore database Amazon RDS for Oracle:

- Oracle Database 21c (21.0.0)
- Oracle Database 19c (19.0.0)

L'esempio seguente recupera il blocco 100 nel file di dati 5.

```
BEGIN
    rdsadmin.rdsadmin_rman_util.recover_datafile_block(
        p_datafile            => 5,
        p_block               => 100,
        p_rman_to_dbms_output => TRUE);
END;
/
```

L'esempio seguente recupera i blocchi da 100 a 150 nel file di dati 5.

```
BEGIN
    rdsadmin.rdsadmin_rman_util.recover_datafile_block(
        p_datafile            => 5,
        p_from_block          => 100,
        p_to_block            => 150,
        p_rman_to_dbms_output => TRUE);
END;
```

```
/
```

Esecuzione di attività di programmazione comuni per le istanze database Oracle

Alcuni processi Scheduler di proprietà di SYS possono interferire con le normali operazioni di database. Oracle Support consiglia di disattivare questi processi o di modificare la pianificazione. Puoi utilizzare il pacchetto `rdsadmin.rdsadmin_dbms_scheduler` di Amazon RDS per eseguire le attività dai processi Oracle Scheduler di proprietà di SYS.

Le procedure `rdsadmin.rdsadmin_dbms_scheduler` sono supportate nelle seguenti versioni del motore database Amazon RDS for Oracle:

- Oracle Database 21c (21.0.0)
- Oracle Database 19c
- Oracle Database 12c Release 2 (12.2) su 12.2.0.2.ru-2019-07.rur-2019-07.r1 o versioni successive alla 12.2
- Oracle Database 12c Release 1 (12.1), versione 12.1.0.2.v17 o versioni successive alle 12.1

Parametri comuni per procedure Oracle Scheduler

Per eseguire attività con Oracle Scheduler, puoi utilizzare le procedure del pacchetto di Amazon RDS `rdsadmin.rdsadmin_dbms_scheduler`. Nelle procedure del pacchetto ci sono diversi parametri comuni. Il pacchetto presenta i seguenti parametri comuni.

Nome del parametro	Tipo di dati	Valori validi	Default	Campo obbligatorio	Descrizione
`name`	varchar2	`'SYS.BSLI _MAINTAI _STATS_JO B' ,'SYS. NUP_ONLI E_IND_BU LD'`	—	Sì	Il nome del processo da modificare. **Note** Al momento, puoi solo modificar

Nome del parametro	Tipo di dati	Valori validi	Default	Campo obbligatorio	Descrizione
					e i lavori `SYS.CLEANUP_ONLINE_IND_BUILD` e `SYS.BSLN_MAINTAIN_STATS_JOB` .
`attribute`	varchar2	`'REPEAT_INTERVAL'` `_NAME'`	–	Sì	Attributo da modificare Per modificare l'intervallo di ripetizione del lavoro, specificare `'REPEAT_INTERVAL'` . Per modificare il nome della pianificazione del lavoro, specificare `'SCHEDULE_NAME'` .

Nome del parametro	Tipo di dati	Valori validi	Default	Campo obbligatorio	Descrizione
`value`	varchar2	Un intervallo o un nome di pianificazione valido, a seconda dell'attributo utilizzato.	–	Sì	Il nuovo valore dell'attributo.

Modifica dei processi di DBMS_SCHEDULER

Puoi utilizzare la procedura Oracle `dbms_scheduler.set_attribute` per modificare determinati componenti di Oracle Scheduler. Per ulteriori informazioni, consulta DBMS_SCHEDULER e Procedura SET_ATTRIBUTE nella documentazione di Oracle.

Quando usi istanze database Amazon RDS, aggiungi il nome di schema SYS come prefisso al nome dell'oggetto. L'esempio seguente imposta l'attributo del piano delle risorse per l'oggetto finestra Lunedì.

```
BEGIN
    DBMS_SCHEDULER.SET_ATTRIBUTE(
        name      => 'SYS.MONDAY_WINDOW',
        attribute => 'RESOURCE_PLAN',
        value     => 'resource_plan_1');
END;
/
```

Modifica delle finestre di AutoTask manutenzione

Le istanze di Amazon RDS for Oracle vengono create con le impostazioni predefinite per le finestre di manutenzione. Le attività di manutenzione automatizzate come la raccolta delle statistiche

dell'ottimizzatore vengono eseguite durante queste finestre. Per impostazione predefinita, le finestre di manutenzione attivano Oracle Database Resource Manager.

Per modificare la finestra, utilizzare il pacchetto `DBMS_SCHEDULER`. Potrebbe essere necessario modificare le impostazioni della finestra di manutenzione per i motivi seguenti:

- Si desidera che i processi di manutenzione vengano eseguiti in un momento diverso, con impostazioni diverse o non del tutto. Ad esempio, è possibile modificare la durata della finestra o modificare il tempo e l'intervallo di ripetizione.
- Si desidera evitare l'impatto sulle prestazioni dell'abilitazione di Resource Manager durante la manutenzione. Ad esempio, se viene specificato il piano di manutenzione predefinito e se si apre la finestra di manutenzione mentre il database è sotto carico, è possibile che vengano visualizzati eventi di attesa come `resmgr:cpu quantum`. Questo evento di attesa è correlato a Database Resource Manager. Sono disponibili le seguenti opzioni:
  - Assicurarsi che le finestre di manutenzione siano attive durante i periodi non di punta per l'istanza del database.
  - Disabilitare il piano di manutenzione predefinito impostando l'attributo `resource_plan` ad una stringa vuota.
  - Imposta il parametro `resource_manager_plan` su `FORCE:` nel gruppo di parametri. Se l'istanza utilizza Enterprise Edition, questa impostazione impedisce l'attivazione dei piani di Database Resource Manager.

Per modificare le impostazioni della finestra di manutenzione

1. Connettersi al database utilizzando un client Oracle SQL.
2. Eseguire una query sulla configurazione corrente per una finestra di scheduler.

   Nell'esempio seguente viene eseguita una query sulla configurazione di `MONDAY_WINDOW`.

   ```
   SELECT ENABLED, RESOURCE_PLAN, DURATION, REPEAT_INTERVAL
   FROM   DBA_SCHEDULER_WINDOWS
   WHERE  WINDOW_NAME='MONDAY_WINDOW';
   ```

   Il seguente output mostra che la finestra utilizza i valori di default.

   ```
   ENABLED         RESOURCE_PLAN                  DURATION         REPEAT_INTERVAL
   ```

```
-------------- ------------------------------ ----------------
 ------------------------------
TRUE            DEFAULT_MAINTENANCE_PLAN       +000 04:00:00
 freq=daily;byday=MON;byhour=22
                                                                ;byminute=0;
 bysecond=0
```

3. Modificare la finestra utilizzando il pacchetto DBMS_SCHEDULER.

   Nell'esempio seguente il piano delle risorse viene impostato su null in modo che il Resource Manager non venga eseguito durante la finestra di manutenzione.

```
BEGIN
  -- disable the window to make changes
  DBMS_SCHEDULER.DISABLE(name=>'"SYS"."MONDAY_WINDOW"',force=>TRUE);

  -- specify the empty string to use no plan
  DBMS_SCHEDULER.SET_ATTRIBUTE(name=>'"SYS"."MONDAY_WINDOW"',
 attribute=>'RESOURCE_PLAN', value=>'');

  -- re-enable the window
  DBMS_SCHEDULER.ENABLE(name=>'"SYS"."MONDAY_WINDOW"');
END;
/
```

   L'esempio seguente imposta la durata massima della finestra su 2 ore.

```
BEGIN
  DBMS_SCHEDULER.DISABLE(name=>'"SYS"."MONDAY_WINDOW"',force=>TRUE);
  DBMS_SCHEDULER.SET_ATTRIBUTE(name=>'"SYS"."MONDAY_WINDOW"',
 attribute=>'DURATION', value=>'0 2:00:00');
  DBMS_SCHEDULER.ENABLE(name=>'"SYS"."MONDAY_WINDOW"');
END;
/
```

   Nell'esempio seguente l'intervallo di ripetizione viene impostato su ogni lunedì alle 10:00.

```
BEGIN
  DBMS_SCHEDULER.DISABLE(name=>'"SYS"."MONDAY_WINDOW"',force=>TRUE);
  DBMS_SCHEDULER.SET_ATTRIBUTE(name=>'"SYS"."MONDAY_WINDOW"',
 attribute=>'REPEAT_INTERVAL',
 value=>'freq=daily;byday=MON;byhour=10;byminute=0;bysecond=0');
```

```
  DBMS_SCHEDULER.ENABLE(name=>'"SYS"."MONDAY_WINDOW"');
END;
/
```

Impostazione del fuso orario per i job di Oracle Scheduler

Per modificare il fuso orario per Oracle Scheduler, è possibile utilizzare la procedura Oracle `dbms_scheduler.set_scheduler_attribute`. Per ulteriori informazioni sul pacchetto `dbms_scheduler`, consulta DBMS_SCHEDULER e SET_SCHEDULER_ATTRIBUTE nella documentazione di Oracle.

Per modificare l'impostazione del fuso orario corrente

1. Connettersi al database utilizzando un client come SQL Developer. Per ulteriori informazioni, consulta Connessione all'istanza database tramite Oracle SQL Developer.
2. Impostare il fuso orario predefinito come segue, sostituendo il fuso orario per *`time_zone_name`*.

```
BEGIN
  DBMS_SCHEDULER.SET_SCHEDULER_ATTRIBUTE(
    attribute => 'default_timezone',
    value => 'time_zone_name'
  );
END;
/
```

Nell'esempio seguente, si modifica il fuso orario in Asia/Shanghai.

Iniziare interrogando il fuso orario corrente, come illustrato di seguito.

```
SELECT VALUE FROM DBA_SCHEDULER_GLOBAL_ATTRIBUTE WHERE
 ATTRIBUTE_NAME='DEFAULT_TIMEZONE';
```

L'output mostra che il fuso orario corrente è ETC/UTC.

```
VALUE
-------
Etc/UTC
```

Quindi si imposta il fuso orario su Asia/Shanghai.

```
BEGIN
  DBMS_SCHEDULER.SET_SCHEDULER_ATTRIBUTE(
    attribute => 'default_timezone',
    value => 'Asia/Shanghai'
  );
END;
/
```

Per ulteriori informazioni sulla modifica del fuso orario di sistema, consulta Fuso orario Oracle.

Disattivazione dei processi Oracle Scheduler di proprietà di SYS

Per disattivare un processo Oracle Scheduler di proprietà dell'utente SYS, utilizza la procedura `rdsadmin.rdsadmin_dbms_scheduler.disable`.

Questa procedura utilizza il parametro comune `name` per le attività Oracle Scheduler. Per ulteriori informazioni, consulta Parametri comuni per procedure Oracle Scheduler.

L'esempio seguente disabilita il lavoro Oracle Scheduler `SYS.CLEANUP_ONLINE_IND_BUILD`.

```
BEGIN
   rdsadmin.rdsadmin_dbms_scheduler.disable('SYS.CLEANUP_ONLINE_IND_BUILD');
END;
/
```

Attivazione dei processi Oracle Scheduler di proprietà di SYS

Per attivare un processo Oracle Scheduler di proprietà SYS, utilizza la procedura `rdsadmin.rdsadmin_dbms_scheduler.enable`.

Questa procedura utilizza il parametro comune `name` per le attività Oracle Scheduler. Per ulteriori informazioni, consulta Parametri comuni per procedure Oracle Scheduler.

L'esempio seguente abilita il lavoro Oracle Scheduler per `SYS.CLEANUP_ONLINE_IND_BUILD`.

```
BEGIN
   rdsadmin.rdsadmin_dbms_scheduler.enable('SYS.CLEANUP_ONLINE_IND_BUILD');
END;
/
```

Modifica dell'intervallo di ripetizione di Oracle Scheduler dei processi di tipo CALENDAR

Per modificare l'intervallo di ripetizione per modificare un lavoro Oracle Scheduler di proprietà SYS di tipo `CALENDAR`, utilizzare la procedura `rdsadmin.rdsadmin_dbms_scheduler.disable`.

Questa procedura utilizza i parametri comuni seguenti per le attività Oracle Scheduler:

- `name`
- `attribute`
- `value`

Per ulteriori informazioni, consulta Parametri comuni per procedure Oracle Scheduler.

L'esempio seguente modifica l'intervallo di ripetizione del lavoro Oracle Scheduler `SYS.CLEANUP_ONLINE_IND_BUILD`.

```
BEGIN
     rdsadmin.rdsadmin_dbms_scheduler.set_attribute(
          name      => 'SYS.CLEANUP_ONLINE_IND_BUILD',
          attribute => 'repeat_interval',
          value     => 'freq=daily;byday=FRI,SAT;byhour=20;byminute=0;bysecond=0');
END;
/
```

Modifica dell'intervallo di ripetizione di Oracle Scheduler dei processi di tipo NAMED

I lavori Oracle Scheduler utilizzano il nome della pianificazione invece dell'intervallo. Per questi tipi di lavori, è necessario creare un nuovo nome di pianificazione nello schema dell'utente principale. Per questo, utilizzare la procedura standard Oracle `sys.dbms_scheduler.create_schedule`. Inoltre, utilizza `rdsadmin.rdsadmin_dbms_scheduler.set_attribute procedure` per assegnare le nuove pianificazioni denominate al processo.

Questa procedura utilizza i parametri comuni seguenti per le attività Oracle Scheduler:

- `name`
- `attribute`
- `value`

Per ulteriori informazioni, consulta Parametri comuni per procedure Oracle Scheduler.

L'esempio seguente modifica l'intervallo di ripetizione del lavoro Oracle Scheduler `SYS.BSLN_MAINTAIN_STATS_JOB`.

```
BEGIN
     DBMS_SCHEDULER.CREATE_SCHEDULE (
          schedule_name   => 'rds_master_user.new_schedule',
          start_date      => SYSTIMESTAMP,
          repeat_interval =>
 'freq=daily;byday=MON,TUE,WED,THU,FRI;byhour=0;byminute=0;bysecond=0',
          end_date        => NULL,
          comments        => 'Repeats daily forever');
END;
/

BEGIN
     rdsadmin.rdsadmin_dbms_scheduler.set_attribute (
          name      => 'SYS.BSLN_MAINTAIN_STATS_JOB',
          attribute => 'schedule_name',
          value     => 'rds_master_user.new_schedule');
END;
/
```

Disattivazione del commit automatico per la creazione di processi in Oracle Scheduler

`DBMS_SCHEDULER.CREATE_JOB` crea i processi Oracle Scheduler immediatamente e conferma le modifiche. Potrebbe essere necessario incorporare la creazione di processi Oracle Scheduler nella transazione utente per:

- Eseguire il rollback del processo Oracle Schedule quando viene eseguito il rollback della transazione dell'utente.
- Creare il processo Oracle Scheduler quando viene confermata la transazione dell'utente principale.

Puoi utilizzare la procedura `rdsadmin.rdsadmin_dbms_scheduler.set_no_commit_flag` per attivare questo comportamento. Questa procedura non richiede parametri. È possibile utilizzare questa procedura nelle seguenti versioni di RDS per Oracle:

- 21.0.0.0.ru-2022-07.rur-2022-07.r1 versioni successive
- 19.0.0.0.ru-2022-07.rur-2022-07.r1 versioni successive

L'esempio seguente disattiva il commit automatico per Oracle Scheduler, crea un processo Oracle Scheduler e quindi esegue il rollback della transazione. Poiché il commit automatico è disattivato, il database esegue il rollback anche del processo Oracle Scheduler.

```
BEGIN
  rdsadmin.rdsadmin_dbms_scheduler.set_no_commit_flag;
  DBMS_SCHEDULER.CREATE_JOB(job_name   => 'EMPTY_JOB',
                            job_type   => 'PLSQL_BLOCK',
                            job_action => 'begin null; end;',
                            auto_drop  => false);
  ROLLBACK;
END;
/

PL/SQL procedure successfully completed.

SELECT * FROM DBA_SCHEDULER_JOBS WHERE JOB_NAME='EMPTY_JOB';

no rows selected
```

Esecuzione di attività diagnostiche comuni per le istanze database Oracle

Oracle Database include un'infrastruttura di diagnosi degli errori che è possibile utilizzare per analizzare i problemi del database. Nella terminologia Oracle, un problema è un errore critico, ad esempio un bug di codice o il danneggiamento dei dati. Un incidente è il verificarsi di un problema. Se lo stesso errore si verifica tre volte, l'infrastruttura mostra tre incidenti di questo problema. Per ulteriori informazioni, consulta Diagnostica e risoluzione dei problemi nella documentazione di Oracle Database.

L'utility ADRCI (Automatic Diagnostic Repository Command Interpreter) è uno strumento a riga di comando Oracle utilizzato per gestire i dati di diagnostica. Ad esempio, è possibile utilizzare questo strumento per analizzare i problemi e creare pacchetti di dati diagnostici. Un pacchetto incidente include dati diagnostici per un incidente o tutti gli incidenti che fanno riferimento a un problema specifico. È possibile caricare un pacchetto incidente, implementato come file zip, nel supporto Oracle.

Per offrire un'esperienza di servizio gestito, Amazon RDS non fornisce accesso shell ad ADRCI. Per eseguire i task diagnostici per l'istanza Oracle, utilizzare invece il pacchetto Amazon RDS `rdsadmin.rdsadmin_adrci_util`.

Utilizzando le funzioni in `rdsadmin_adrci_util`, è possibile elencare e creare pacchetti di problemi e incidenti, nonché visualizzare i file di traccia. Tutte le funzioni restituiscono un ID attività. Questo ID fa parte del nome del file di registro che contiene l'output ADRCI, come in `dbtask-`*`task_id`*`.log`. Il file di registro si trova nella directory BDUMP. È possibile scaricare il file di registro seguendo la procedura descritta in[Download di un file di log di database](#).

Parametri comuni per le procedure diagnostiche

Per eseguire attività diagnostiche, utilizzare le funzioni nel pacchetto Amazon RDS `rdsadmin.rdsadmin_adrci_util`. Il pacchetto presenta i seguenti parametri comuni.

Nome del parametro	Tipo di dati	Valori validi	Default	Campo obbligatorio	Descrizione
`incident_id`	numero	Un ID incidente valido o null	Null	No	Se il valore è null, la funzione mostra tutti gli incidenti. Se il valore non è null e rappresenta un ID incidente valido, la funzione mostra l'incidente specificato.
`problem_id`	numero	Un ID di problema valido o null	Null	No	Se il valore è null, la funzione mostra tutti i problemi. Se il valore non è null e rappresenta un ID problema valido, la funzione mostra il problema specificato.
`last`	numero	Un numero intero valido	Null	No	Se il valore è null, la funzione visualizza al massimo 50 elementi. Se il valore non è null,

Nome del parametro	Tipo di dati	Valori validi	Default	Campo obbligatorio	Descrizione
		maggiore di 0 o null			la funzione visualizza il numero specificato.

Elenco degli incidenti

Per elencare gli incidenti diagnostici per Oracle, utilizzare la funzione Amazon RDS `rdsadmin.rdsadmin_adrci_util.list_adrci_incidents`. È possibile elencare gli incidenti in modalità base o dettagliata. Per impostazione predefinita, la funzione elenca i 50 incidenti più recenti.

Questa funzione utilizza i seguenti parametri comuni:

- `incident_id`
- `problem_id`
- `last`

Se specifichi `incident_id` e `problem_id`, `incident_id` sostituisce `problem_id`. Per ulteriori informazioni, consulta Parametri comuni per le procedure diagnostiche.

Questa funzione utilizza il seguente parametro aggiuntivo.

Nome del parametro	Tipo di dati	Valori validi	Default	Campo obbligatorio	Descrizione
`detail`	booleano	`TRUE` o `FALSE`	`FALSE`	No	Se `TRUE`, la funzione elenca gli incidenti in modalità dettaglio. Se `FALSE`, la funzione elenca gli incidenti in modalità base.

Per elencare tutti gli incidenti, esegui una query della funzione `rdsadmin.rdsadmin_adrci_util.list_adrci_incidents` senza argomenti. La query restituisce l'ID attività.

```
SQL> SELECT rdsadmin.rdsadmin_adrci_util.list_adrci_incidents AS task_id FROM DUAL;

TASK_ID
------------------
1590786706158-3126
```

Oppure chiama la funzione `rdsadmin.rdsadmin_adrci_util.list_adrci_incidents` senza argomenti e memorizza l'output in una variabile client SQL. Puoi utilizzare la variabile in altre istruzioni.

```
SQL> VAR task_id VARCHAR2(80);
SQL> EXEC :task_id := rdsadmin.rdsadmin_adrci_util.list_adrci_incidents;

PL/SQL procedure successfully completed.
```

Per leggere il file di registro, chiamare la procedura Amazon RDS `rdsadmin.rds_file_util.read_text_file`. Fornire l'ID attività come parte del nome del file. Il seguente output mostra tre incidenti: 53523, 53522 e 53521.

```
SQL> SELECT * FROM TABLE(rdsadmin.rds_file_util.read_text_file('BDUMP',
 'dbtask-'||:task_id||'.log'));

TEXT
--------------------------------------------------------------------------------
2020-05-29 21:11:46.193 UTC [INFO ] Listing ADRCI incidents.
2020-05-29 21:11:46.256 UTC [INFO ]
ADR Home = /rdsdbdata/log/diag/rdbms/orcl_a/ORCL:
*************************************************************************
INCIDENT_ID PROBLEM_KEY                                                 CREATE_TIME
----------- ----------------------------------------------------------
 ----------------------------------------
53523       ORA 700 [EVENT_CREATED_INCIDENT] [942] [SIMULATED_ERROR_003 2020-05-29
 20:15:20.928000 +00:00
53522       ORA 700 [EVENT_CREATED_INCIDENT] [942] [SIMULATED_ERROR_002 2020-05-29
 20:15:15.247000 +00:00
53521       ORA 700 [EVENT_CREATED_INCIDENT] [942] [SIMULATED_ERROR_001 2020-05-29
 20:15:06.047000 +00:00
```

```
3 rows fetched


2020-05-29 21:11:46.256 UTC [INFO ] The ADRCI incidents were successfully listed.
2020-05-29 21:11:46.256 UTC [INFO ] The task finished successfully.

14 rows selected.
```

Per elencare un particolare incidente, specificarne l'ID utilizzando il parametro `incident_id`. Nell'esempio seguente, è possibile eseguire la query del file di registro solo per incidente 53523.

```
SQL> EXEC :task_id :=
 rdsadmin.rdsadmin_adrci_util.list_adrci_incidents(incident_id=>53523);

PL/SQL procedure successfully completed.

SQL> SELECT * FROM TABLE(rdsadmin.rds_file_util.read_text_file('BDUMP',
 'dbtask-'||:task_id||'.log'));

TEXT
--------------------------------------------------------------------------------
2020-05-29 21:15:25.358 UTC [INFO ] Listing ADRCI incidents.
2020-05-29 21:15:25.426 UTC [INFO ]
ADR Home = /rdsdbdata/log/diag/rdbms/orcl_a/ORCL:
*************************************************************************
INCIDENT_ID          PROBLEM_KEY
 CREATE_TIME
-------------------- ----------------------------------------------------------
 ---------------------------------
53523                ORA 700 [EVENT_CREATED_INCIDENT] [942] [SIMULATED_ERROR_003
 2020-05-29 20:15:20.928000 +00:00
1 rows fetched


2020-05-29 21:15:25.427 UTC [INFO ] The ADRCI incidents were successfully listed.
2020-05-29 21:15:25.427 UTC [INFO ] The task finished successfully.

12 rows selected.
```

Elenco dei problemi

Per elencare i problemi di diagnostica per Oracle, utilizzare la funzione Amazon RDS `rdsadmin.rdsadmin_adrci_util.list_adrci_problems`.

Per impostazione predefinita, la funzione elenca i 50 problemi più recenti.

Questa funzione utilizza i parametri comuni `problem_id` e `last`. Per ulteriori informazioni, consulta Parametri comuni per le procedure diagnostiche.

Per ottenere l'ID attività per tutti i problemi, chiamare la funzione `rdsadmin.rdsadmin_adrci_util.list_adrci_problems` senza argomenti e memorizzare l'output in una variabile client SQL.

```
SQL> EXEC :task_id := rdsadmin.rdsadmin_adrci_util.list_adrci_problems;

PL/SQL procedure successfully completed.
```

Per leggere il file di registro, chiamare la funzione `rdsadmin.rds_file_util.read_text_file`, fornendo l'ID attività come parte del nome del file. Nell'output seguente, il file di registro mostra tre problemi: 1, 2 e 3.

```
SQL> SELECT * FROM TABLE(rdsadmin.rds_file_util.read_text_file('BDUMP',
 'dbtask-'||:task_id||'.log'));

TEXT
--------------------------------------------------------------------------------
2020-05-29 21:18:50.764 UTC [INFO ] Listing ADRCI problems.
2020-05-29 21:18:50.829 UTC [INFO ]
ADR Home = /rdsdbdata/log/diag/rdbms/orcl_a/ORCL:
*************************************************************************
PROBLEM_ID   PROBLEM_KEY                                                 LAST_INCIDENT
       LASTINC_TIME
---------- ------------------------------------------------------------ -------------
 -------------------------------
2          ORA 700 [EVENT_CREATED_INCIDENT] [942] [SIMULATED_ERROR_003 53523
 2020-05-29 20:15:20.928000 +00:00
3          ORA 700 [EVENT_CREATED_INCIDENT] [942] [SIMULATED_ERROR_002 53522
 2020-05-29 20:15:15.247000 +00:00
1          ORA 700 [EVENT_CREATED_INCIDENT] [942] [SIMULATED_ERROR_001 53521
 2020-05-29 20:15:06.047000 +00:00
3 rows fetched


2020-05-29 21:18:50.829 UTC [INFO ] The ADRCI problems were successfully listed.
2020-05-29 21:18:50.829 UTC [INFO ] The task finished successfully.
```

```
14 rows selected.
```

Nell'esempio seguente, è possibile elencare solo il problema 3.

```
SQL> EXEC :task_id := rdsadmin.rdsadmin_adrci_util.list_adrci_problems(problem_id=>3);

PL/SQL procedure successfully completed.
```

Per leggere il file di registro per il problema 3, chiamare `rdsadmin.rds_file_util.read_text_file`. Fornire l'ID attività come parte del nome del file.

```
SQL> SELECT * FROM TABLE(rdsadmin.rds_file_util.read_text_file('BDUMP',
 'dbtask-'||:task_id||'.log'));

TEXT
--------------------------------------------------------------------------------
2020-05-29 21:19:42.533 UTC [INFO ] Listing ADRCI problems.
2020-05-29 21:19:42.599 UTC [INFO ]
ADR Home = /rdsdbdata/log/diag/rdbms/orcl_a/ORCL:
*********************************************************************
PROBLEM_ID PROBLEM_KEY                                                 LAST_INCIDENT
 LASTINC_TIME
---------- ------------------------------------------------------------ -------------
 -------------------------------
3          ORA 700 [EVENT_CREATED_INCIDENT] [942] [SIMULATED_ERROR_002 53522
 2020-05-29 20:15:15.247000 +00:00
1 rows fetched


2020-05-29 21:19:42.599 UTC [INFO ] The ADRCI problems were successfully listed.
2020-05-29 21:19:42.599 UTC [INFO ] The task finished successfully.

12 rows selected.
```

Creazione di pacchetti incidenti

È possibile creare pacchetti incidenti utilizzando la funzione Amazon RDS `rdsadmin.rdsadmin_adrci_util.create_adrci_package`. L'output è un file zip che è possibile fornire al supporto Oracle.

Questa funzione utilizza i seguenti parametri comuni:

- problem_id
- incident_id

Assicurarsi di specificare uno dei parametri precedenti. Se si specificano entrambi i parametri, incident_id sovrascrive problem_id. Per ulteriori informazioni, consulta Parametri comuni per le procedure diagnostiche.

Per creare un pacchetto per un incidente specifico, chiamare la funzione rdsadmin.rdsadmin_adrci_util.create_adrci_package Amazon RDS con il parametro incident_id. Nell'esempio seguente viene creato un pacchetto per l'incidente 53523.

```
SQL> EXEC :task_id :=
 rdsadmin.rdsadmin_adrci_util.create_adrci_package(incident_id=>53523);

PL/SQL procedure successfully completed.
```

Per leggere il file di registro, chiamare rdsadmin.rds_file_util.read_text_file. È possibile fornire l'ID attività come parte del nome del file. L'output mostra che hai generato il pacchetto incidente ORA700EVE_20200529212043_COM_1.zip.

```
SQL> SELECT * FROM TABLE(rdsadmin.rds_file_util.read_text_file('BDUMP',
 'dbtask-'||:task_id||'.log'));

TEXT
--------------------------------------------------------------------------------
2020-05-29 21:20:43.031 UTC [INFO ] The ADRCI package is being created.
2020-05-29 21:20:47.641 UTC [INFO ] Generated package 1 in file /rdsdbdata/log/trace/
ORA700EVE_20200529212043_COM_1.zip, mode complete
2020-05-29 21:20:47.642 UTC [INFO ] The ADRCI package was successfully created.
2020-05-29 21:20:47.642 UTC [INFO ] The task finished successfully.
```

Per creare un pacchetto di dati diagnostici per un particolare problema, specificarne l'ID utilizzando il parametro problem_id. Nell'esempio seguente, si impacchettano i dati solo per il problema 3.

```
SQL> EXEC :task_id := rdsadmin.rdsadmin_adrci_util.create_adrci_package(problem_id=>3);

PL/SQL procedure successfully completed.
```

Per leggere l'output dell'attività, chiamare `rdsadmin.rds_file_util.read_text_file`, fornendo l'ID attività come parte del nome del file. L'output mostra che hai generato il pacchetto incidente `ORA700EVE_20200529212111_COM_1.zip`.

```
SQL> SELECT * FROM TABLE(rdsadmin.rds_file_util.read_text_file('BDUMP',
 'dbtask-'||:task_id||'.log'));

TEXT
--------------------------------------------------------------------------------
2020-05-29 21:21:11.050 UTC [INFO ] The ADRCI package is being created.
2020-05-29 21:21:15.646 UTC [INFO ] Generated package 2 in file /rdsdbdata/log/trace/
ORA700EVE_20200529212111_COM_1.zip, mode complete
2020-05-29 21:21:15.646 UTC [INFO ] The ADRCI package was successfully created.
2020-05-29 21:21:15.646 UTC [INFO ] The task finished successfully.
```

È inoltre possibile scaricare il file di registro. Per ulteriori informazioni, consulta Download di un file di log di database.

Visualizzazione di file di traccia

Puoi utilizzare la funzione Amazon RDS `rdsadmin.rdsadmin_adrci_util.show_adrci_tracefile` per elencare i file di traccia nella directory di traccia e tutte le directory degli incidenti nella home page ADR corrente. È inoltre possibile visualizzare il contenuto dei file di traccia e dei file di traccia degli incidenti.

Questa funzione utilizza il seguente parametro.

Nome del parametro	Tipo di dati	Valori validi	Default	Campo obbligatorio	Descrizione
`filename`	varchar2	Un nome di file di traccia valido	Null	No	Se il valore è null, la funzione visualizza tutti i file di traccia. Se non è null, la funzione mostra il file specificato.

Per visualizzare il file di traccia, chiama la funzione `rdsadmin.rdsadmin_adrci_util.show_adrci_tracefile` Amazon RDS.

```
SQL> EXEC :task_id := rdsadmin.rdsadmin_adrci_util.show_adrci_tracefile;

PL/SQL procedure successfully completed.
```

Per elencare i nomi dei file di traccia, chiamare la procedura `rdsadmin.rds_file_util.read_text_file` Amazon RDS fornendo l'ID attività come parte del nome file.

```
SQL> SELECT * FROM TABLE(rdsadmin.rds_file_util.read_text_file('BDUMP',
 'dbtask-'||:task_id||'.log')) WHERE TEXT LIKE '%/alert_%';

TEXT
--------------------------------------------------------------------
     diag/rdbms/orcl_a/ORCL/trace/alert_ORCL.log.2020-05-28
     diag/rdbms/orcl_a/ORCL/trace/alert_ORCL.log.2020-05-27
     diag/rdbms/orcl_a/ORCL/trace/alert_ORCL.log.2020-05-26
     diag/rdbms/orcl_a/ORCL/trace/alert_ORCL.log.2020-05-25
     diag/rdbms/orcl_a/ORCL/trace/alert_ORCL.log.2020-05-24
     diag/rdbms/orcl_a/ORCL/trace/alert_ORCL.log.2020-05-23
     diag/rdbms/orcl_a/ORCL/trace/alert_ORCL.log.2020-05-22
     diag/rdbms/orcl_a/ORCL/trace/alert_ORCL.log.2020-05-21
     diag/rdbms/orcl_a/ORCL/trace/alert_ORCL.log

9 rows selected.
```

Nell'esempio seguente, si genera un output per`alert_ORCL.log`.

```
SQL> EXEC :task_id := rdsadmin.rdsadmin_adrci_util.show_adrci_tracefile('diag/rdbms/
orcl_a/ORCL/trace/alert_ORCL.log');

PL/SQL procedure successfully completed.
```

Per leggere il file di registro, chiamare `rdsadmin.rds_file_util.read_text_file`. Fornire l'ID attività come parte del nome del file. L'output mostra le prime 10 righe di Alert_Orcl.log.

```
SQL> SELECT * FROM TABLE(rdsadmin.rds_file_util.read_text_file('BDUMP',
 'dbtask-'||:task_id||'.log')) WHERE ROWNUM <= 10;
```

```
TEXT
--------------------------------------------------------------------------------
2020-05-29 21:24:02.083 UTC [INFO ] The trace files are being displayed.
2020-05-29 21:24:02.128 UTC [INFO ] Thu May 28 23:59:10 2020
Thread 1 advanced to log sequence 2048 (LGWR switch)
  Current log# 3 seq# 2048 mem# 0: /rdsdbdata/db/ORCL_A/onlinelog/o1_mf_3_hbl2p8xs_.log
Thu May 28 23:59:10 2020
Archived Log entry 2037 added for thread 1 sequence 2047 ID 0x5d62ce43 dest 1:
Fri May 29 00:04:10 2020
Thread 1 advanced to log sequence 2049 (LGWR switch)
  Current log# 4 seq# 2049 mem# 0: /rdsdbdata/db/ORCL_A/onlinelog/o1_mf_4_hbl2qgmh_.log
Fri May 29 00:04:10 2020

10 rows selected.
```

È inoltre possibile scaricare il file di registro. Per ulteriori informazioni, consulta Download di un file di log di database.

Esecuzione di varie attività per istanze database Oracle

Di seguito viene descritto come eseguire attività DBA varie sulle istanze database Amazon RDS che eseguono Oracle. Per fornire un'esperienza di servizio gestito, Amazon RDS non fornisce accesso shell alle istanze database e limita l'accesso a certe procedure e tabelle di sistema che richiedono privilegi avanzati.

Argomenti

- Creazione ed eliminazione di directory nello spazio di archiviazione dati principale
- Generazione di un elenco dei file in una directory di istanze database
- Lettura dei file in una directory di istanze database
- Accesso ai file Opatch
- Gestione delle attività degli advisor
- Trasporto di tablespace

Creazione ed eliminazione di directory nello spazio di archiviazione dati principale

Puoi usare la procedura in Amazon RDS per creare director `rdsadmin.rdsadmin_util.create_directory`. Puoi creare fino a 10000 directory, tutte

posizionate nello spazio principale di storage dei dati. Puoi usare la procedura Amazon RDS in per eliminare le director `rdsadmin.rdsadmin_util.drop_directory`.

Le procedure `create_directory` e `drop_directory` hanno il seguente parametro obbligatorio.

Nome del parametro	Tipo di dati	Default	Campo obbligatorio	Descrizione
`p_directory_name`	varchar2	—	Sì	Il nome della directory.

L'esempio seguente crea una nuova directory denominata `PRODUCT_DESCRIPTIONS`.

```
EXEC rdsadmin.rdsadmin_util.create_directory(p_directory_name =>
 'product_descriptions');
```

Il dizionario dati memorizza il nome della directory in maiuscolo. Puoi elencare le directory eseguendo query su `DBA_DIRECTORIES`. Il sistema seleziona il percorso host effettivo automaticamente. L'esempio seguente mostra ottiene il percorso di directory per la directory denominata `PRODUCT_DESCRIPTIONS`:

```
SELECT DIRECTORY_PATH
  FROM DBA_DIRECTORIES
 WHERE DIRECTORY_NAME='PRODUCT_DESCRIPTIONS';

DIRECTORY_PATH
----------------------------------------
/rdsdbdata/userdirs/01
```

Il nome utente master per l'istanza database ha privilegi di lettura e scrittura nella nuova directory e può concedere l'accesso ad altri utenti. I privilegi `EXECUTE` non sono disponibili per le directory su una istanza database. Le directory vengono create nello spazio principale dello storage dei dati e consumeranno spazio e larghezza di banda I/O.

L'esempio seguente elimina la directory denominata `PRODUCT_DESCRIPTIONS`.

```
EXEC rdsadmin.rdsadmin_util.drop_directory(p_directory_name => 'product_descriptions');
```

> **Note**
>
> È inoltre possibile eliminare una directory utilizzando il comando Oracle SQL `DROP DIRECTORY`.

Il rilascio di una directory non rimuove i suoi contenuti. Poiché il metodo `rdsadmin.rdsadmin_util.create_directory` può riutilizzare percorsi, i file nelle directory eliminate possono apparire in una directory appena creata. Prima di eliminare una directory, si consiglia di utilizzare `UTL_FILE.FREMOVE` per rimuovere i file dalla directory. Per ulteriori informazioni, consulta la sezione relativa alla Procedura FREMOVE nella documentazione di Oracle.

Generazione di un elenco dei file in una directory di istanze database

Puoi usare la procedura in Amazon RDS per elencare i file in una director `rdsadmin.rds_file_util.listdir`. Questa procedura non è supportata su una replica Oracle. La procedura `listdir` include i seguenti parametri.

Nome del parametro	Tipo di dati	Default	Campo obbligatorio	Descrizione
`p_directory`	varchar2	—	Sì	Il nome della directory da elencare.

Nel seguente esempio vengono assegnati i privilegi di lettura/scrittura nella directory `PRODUCT_DESCRIPTIONS` all'utente `rdsadmin` e quindi elenca i file in questa directory.

```
GRANT READ,WRITE ON DIRECTORY PRODUCT_DESCRIPTIONS TO rdsadmin;
SELECT * FROM TABLE(rdsadmin.rds_file_util.listdir(p_directory =>
 'PRODUCT_DESCRIPTIONS'));
```

Lettura dei file in una directory di istanze database

Puoi usare la procedura in Amazon RDS per leggere un file di test `rdsadmin.rds_file_util.read_text_file`. La procedura `read_text_file` include i seguenti parametri.

Nome del parametro	Tipo di dati	Default	Campo obbligatorio	Descrizione
p_directory	varchar2	—	Sì	Il nome della directory che contiene il file .
p_filename	varchar2	—	Sì	Nome del file da leggere.

Nell'esempio seguente viene creato il file `rice.txt` nella directory `PRODUCT_DESCRIPTIONS`.

```
declare
  fh sys.utl_file.file_type;
begin
  fh := utl_file.fopen(location=>'PRODUCT_DESCRIPTIONS', filename=>'rice.txt',
 open_mode=>'w');
  utl_file.put(file=>fh, buffer=>'AnyCompany brown rice, 15 lbs');
  utl_file.fclose(file=>fh);
end;
/
```

L'esempio seguente legge il file `rice.txt` dalla directory `PRODUCT_DESCRIPTIONS`.

```
SELECT * FROM TABLE
    (rdsadmin.rds_file_util.read_text_file(
        p_directory => 'PRODUCT_DESCRIPTIONS',
        p_filename  => 'rice.txt'));
```

Accesso ai file Opatch

Opatch è una utility Oracle che consente l'applicazione e il rollback delle patch al software Oracle. Il meccanismo Oracle per determinare quali patch sono state applicate a un database è il comando `opatch lsinventory`. Per aprire le richieste di assistenza per i clienti BYOL (Bring Your Own Licence), il supporto Oracle richiede il file `lsinventory` e talvolta il file `lsinventory_detail` generato da Opatch.

Per offrire un'esperienza di servizio gestito, Amazon RDS non fornisce accesso shell a Opatch. Invece, l'`lsinventory-`*`dbv`*`.txt` nella directory BDUMP contiene le informazioni sulla patch

relative alla versione corrente del motore. Quando esegui un aggiornamento minore o importante, Amazon RDS aggiorna `lsinventory-dbv.txt` entro un'ora dall'applicazione della patch. Per verificare le patch applicate, leggere `lsinventory-dbv.txt`. Questa operazione è simile all'esecuzione del comando `opatch lsinventory`.

> **Note**
>
> Gli esempi riportati in questa sezione presuppongono che la directory BDUMP sia denominata BDUMP. In una replica di lettura il nome della directory BDUMP è diverso. Per informazioni su come ottenere il nome BDUMP eseguendo una query `V$DATABASE.DB_UNIQUE_NAME` su una replica di lettura, consulta Elenco di file.

I file di caricamento dati utilizzano la convenzione di denominazione Amazon RDS `lsinventory-dbv.txt` e `lsinventory_detail-dbv.txt`, dove *dbv* è il nome completo della versione DB. Il file `lsinventory-dbv.txt` è disponibile in tutte le versioni DB. Il file corrispondente `lsinventory_detail-dbv.txt` è disponibile nelle seguenti versioni DB:

- 19.0.0.0, ru-2020-01.rur-2020-01.r1 o successive
- 12.2.0.1, ru-2020-01.rur-2020-01.r1 o successive
- 12.1.0.2, v19 o versioni successive

Ad esempio, se la versione del database è 19.0.0.0.ru-2021-07.rur-2021-07.r1, i file di inventario hanno i seguenti nomi.

```
lsinventory-19.0.0.0.ru-2021-07.rur-2021-07.r1.txt
lsinventory_detail-19.0.0.0.ru-2021-07.rur-2021-07.r1.txt
```

Assicurarsi di scaricare i file corrispondenti alla versione corrente del motore DB.

Console

Per scaricare un modello di inventario utilizzando la console

1. Apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel pannello di navigazione, scegliere Databases (Database).
3. Scegliere il nome dell'istanza di database che ha il file di log che si desidera visualizzare.

4. Scegliere la scheda Logs & events (Log ed eventi).
5. Scorrere fino alla sezione Logs (Log).
6. Nella sezione Log cercare `lsinventory`.
7. Selezionare il file a cui si desidera accedere, quindi scegliere Scarica.

SQL

Per leggere `lsinventory-`*dbv*`.txt` in un client SQL, è possibile utilizzare un'istruzione `SELECT`. Per questa tecnica, utilizzare una delle seguenti funzioni `rdsadmin`: `rdsadmin.rds_file_util.read_text_file` o `rdsadmin.tracefile_listing`.

Nella query di esempio seguente sostituire *dbv* con la versione di Oracle DB. Ad esempio, la versione del DB potrebbe essere 19.0.0.ru-2020-04.rur-2020-04.r1.

```
SELECT text
FROM   TABLE(rdsadmin.rds_file_util.read_text_file('BDUMP', 'lsinventory-dbv.txt'));
```

PL/SQL

Per leggere `lsinventory-`*dbv*`.txt` in un client SQL, è possibile scrivere un programma PL/SQL. Questo programma utilizza `utl_file` per leggere il file e `dbms_output` per stamparlo. Questi sono pacchetti forniti da Oracle.

Nel programma di esempio seguente sostituire *dbv* con la versione di Oracle DB. Ad esempio, la versione del DB potrebbe essere 19.0.0.ru-2020-04.rur-2020-04.r1.

```
SET SERVEROUTPUT ON
DECLARE
  v_file              SYS.UTL_FILE.FILE_TYPE;
  v_line              VARCHAR2(1000);
  v_oracle_home_type  VARCHAR2(1000);
  c_directory         VARCHAR2(30) := 'BDUMP';
  c_output_file       VARCHAR2(30) := 'lsinventory-dbv.txt';
BEGIN
  v_file := SYS.UTL_FILE.FOPEN(c_directory, c_output_file, 'r');
  LOOP
    BEGIN
      SYS.UTL_FILE.GET_LINE(v_file, v_line,1000);
      DBMS_OUTPUT.PUT_LINE(v_line);
```

```
    EXCEPTION
      WHEN no_data_found THEN
        EXIT;
    END;
  END LOOP;
END;
/
```

Oppure interrogare `rdsadmin.tracefile_listing` ed eseguire lo spooling dell'output in un file. Nell'esempio seguente viene eseguito lo spooling dell'output in `/tmp/tracefile.txt`.

```
SPOOL /tmp/tracefile.txt
SELECT *
FROM   rdsadmin.tracefile_listing
WHERE  FILENAME LIKE 'lsinventory%';
SPOOL OFF;
```

Gestione delle attività degli advisor

Oracle Database include un certo numero di advisor. Ogni advisor supporta attività automatizzate e manuali. Puoi utilizzare le procedure nel pacchetto `rdsadmin.rdsadmin_util` per gestire alcune attività di advisor.

Le procedure delle attività di advisor sono disponibili nelle seguenti versioni del motore:

- Oracle Database 21c (21.0.0)
- Versione 19.0.0.0.ru-2021-01.rur-2021-01.r1 e versioni successive di Oracle Database 19c

  Per ulteriori informazioni, consultare Versione 19.0.0.0.ru-2021-01.rur-2021-01.r1 nelle Note di rilascio di Amazon RDS for Oracle.
- Versione 12.2.0.1.ru-2021-01.rur-2021-01.r1 e versioni successive di Oracle Database 12c (Release 2) 12.2.0.1

  Per ulteriori informazioni, consultare Versione 12.2.0.1.ru-2021-01.rur-2021-01.r1 nelle Note di rilascio di Amazon RDS for Oracle.

Argomenti

- Impostazione dei parametri per le attività di advisor
- Disattivazione di AUTO_STATS_ADVISOR_TASK

- Riattivazione di AUTO_STATS_ADVISOR_TASK

Impostazione dei parametri per le attività di advisor

Per impostare i parametri per alcune attività di advisor, utilizza la procedura Amazon RDS `rdsadmin.rdsadmin_util.advisor_task_set_parameter`. La procedura `advisor_task_set_parameter` include i seguenti parametri.

Nome del parametro	Tipo di dati	Default	Campo obbligatorio	Descrizione
`p_task_name`	varchar2	—	Sì	Il nome dell'attività di advisor di cui si desidera modificare i parametri. I valori seguenti sono validi: • `AUTO_STATS_ADVISOR_TASK` • `INDIVIDUAL_STATS_ADVISOR_TASK` • `SYS_AUTO_SPM_EVOLVE_TASK` • `SYS_AUTO_SQL_TUNING_TASK`
`p_parameter`	varchar2	—	Sì	Il nome del parametro dell'attività. Per trovare parametri validi per un'attività di advisor, esegui la query riportata. Sostituisci *`p_task_name`* con un valore valido per `p_task_name` : `COL PARAMETER_NAME FORMAT a30` `COL PARAMETER_VALUE FORMAT a30` `SELECT PARAMETER_NAME, PARAMETER_VALUE` `FROM DBA_ADVISOR_PARAMETERS` `WHERE TASK_NAME='` *`p_task_name`* `'` `AND PARAMETER_VALUE != 'UNUSED'` `ORDER BY PARAMETER_NAME;`
`p_value`	varchar2	—	Sì	Il valore di un parametro di attività. Per trovare valori validi per i parametri delle attività, esegui

Nome del parametro	Tipo di dati	Default	Campo obbligatorio	Descrizione
				la query riportata. Sostituisci *p_task_name* con un valore valido per p_task_name : `COL PARAMETER_NAME FORMAT a30` `COL PARAMETER_VALUE FORMAT a30` `SELECT PARAMETER_NAME, PARAMETER _VALUE` `FROM DBA_ADVISOR_PARAMETERS` `WHERE TASK_NAME='  p_task_name '` `AND PARAMETER_VALUE != 'UNUSED'` `ORDER BY PARAMETER_NAME;`

Il seguente programma PL/SQL imposta ACCEPT_PLANS su FALSE per SYS_AUTO_SPM_EVOLVE_TASK. L'attività automatizzata Gestione piano SQL verifica i piani e genera un report dei risultati, ma non evolve automaticamente i piani. Puoi utilizzare un report per identificare le nuove baseline del piano SQL e accettarle manualmente.

```
BEGIN
  rdsadmin.rdsadmin_util.advisor_task_set_parameter(
    p_task_name => 'SYS_AUTO_SPM_EVOLVE_TASK',
    p_parameter => 'ACCEPT_PLANS',
    p_value     => 'FALSE');
END;
```

Il seguente programma PL/SQL imposta EXECUTION_DAYS_TO_EXPIRE su 10 per AUTO_STATS_ADVISOR_TASK. L'attività predefinita AUTO_STATS_ADVISOR_TASK viene eseguita automaticamente nella finestra di manutenzione una volta al giorno. Nell'esempio viene impostato il periodo di conservazione per l'esecuzione dell'attività su 10 giorni.

```
BEGIN
  rdsadmin.rdsadmin_util.advisor_task_set_parameter(
    p_task_name => 'AUTO_STATS_ADVISOR_TASK',
    p_parameter => 'EXECUTION_DAYS_TO_EXPIRE',
    p_value     => '10');
END;
```

Disattivazione di AUTO_STATS_ADVISOR_TASK

Per disabilitare AUTO_STATS_ADVISOR_TASK, utilizza la procedura Amazon RDS `rdsadmin.rdsadmin_util.advisor_task_drop`. La procedura `advisor_task_drop` accetta il seguente parametro.

> **Note**
>
> Questa procedura è disponibile in Oracle Database 12c Release 2 (12.2.0.1) e versioni successive.

Nome del parametro	Tipo di dati	Default	Campo obbligatorio	Descrizione
`p_task_name`	varchar2	—	Sì	Il nome dell'attività di advisor da disabilitare. L'unico valore valido è `AUTO_STATS_ADVISOR_TASK` .

Il seguente comando elimina AUTO_STATS_ADVISOR_TASK.

```
EXEC rdsadmin.rdsadmin_util.advisor_task_drop('AUTO_STATS_ADVISOR_TASK')
```

Puoi riabilitare AUTO_STATS_ADVISOR_TASK utilizzando `rdsadmin.rdsadmin_util.dbms_stats_init`.

Riattivazione di AUTO_STATS_ADVISOR_TASK

Per riattivare AUTO_STATS_ADVISOR_TASK, utilizzare la procedura Amazon RDS `rdsadmin.rdsadmin_util.dbms_stats_init` . La procedura `dbms_stats_init` non richiede parametri.

Il seguente comando abilita nuovamente AUTO_STATS_ADVISOR_TASK.

```
EXEC rdsadmin.rdsadmin_util.dbms_stats_init()
```

Trasporto di tablespace

Usa il pacchetto Amazon RDS `rdsadmin.rdsadmin_transport_util` per copiare un set di tablespace da un database Oracle on-premise a un'istanza database RDS per Oracle. A livello fisico, questa funzionalità tablespace trasportabile copia in modo incrementale i file di dati e metadati nell'istanza di destinazione. È possibile trasferire i file mediante Amazon EFS o Amazon S3. Per ulteriori informazioni, consulta Migrazione utilizzando le tablespace trasportabili Oracle.

Argomenti

- Importazione di tablespace trasportate nell'istanza database
- Importazione dei metadati delle tablespace trasportabili nell'istanza database
- Elenco dei file orfani dopo un'importazione della tablespace
- Eliminazione di file di dati rimasti orfani dopo un'importazione della tablespace

Importazione di tablespace trasportate nell'istanza database

Usa la procedura `rdsadmin.rdsadmin_transport_util.import_xtts_tablespaces` per ripristinare le tablespace esportate in precedenza da un'istanza database di origine. In questa fase di trasporto, viene eseguito il backup delle tablespace di sola lettura, vengono esportati i metadati di Data Pump, vengono trasferiti questi file nell'istanza database di destinazione e quindi vengono importate le tablespace. Per ulteriori informazioni, consulta Fase 4: trasporto delle tablespace.

Sintassi

```
FUNCTION import_xtts_tablespaces(
    p_tablespace_list IN CLOB,
    p_directory_name  IN VARCHAR2,
    p_platform_id     IN NUMBER DEFAULT 13,
    p_parallel        IN INTEGER DEFAULT 0) RETURN VARCHAR2;
```

Parametri

Nome del parametro	Tipo di dati	Default	Campo obbligatorio	Descrizione
`p_tablespace_list`	`CLOB`	—	Sì	L'elenco delle tablespace da importare.

Nome del parametro	Tipo di dati	Default	Campo obbligatorio	Descrizione
p_directory_name	VARCHAR2	—	Sì	La directory che contiene i backup del tablespace.
p_platform_id	NUMBER	13	No	Fornire un ID piattaforma che corrisponde a quello specificato durante la fase di backup. Per trovare un elenco di piattaforme, esegui una query su V$TRANSPORTABLE_PLATFORM . La piattaforma predefinita è Linux x86 a 64 bit, che è in formato little-endian.
p_parallel	INTEGER	0	No	Il grado di parallelismo. Il parallelismo è disabilitato per impostazione predefinita.

Esempi

L'esempio seguente importa le tablespace *TBS1*, *TBS2* e *TBS3* dalla directory *DATA_PUMP_DIR*. La piattaforma di origine è AIX based Systems (64 bit), che ha l'ID della piattaforma di. 6 È possibile trovare gli ID della piattaforma eseguendo una query. V$TRANSPORTABLE_PLATFORM

```
VAR task_id CLOB

BEGIN
  :task_id:=rdsadmin.rdsadmin_transport_util.import_xtts_tablespaces(
        'TBS1,TBS2,TBS3',
        'DATA_PUMP_DIR',
        p_platform_id => 6);
END;
```

```
/

PRINT task_id
```

Importazione dei metadati delle tablespace trasportabili nell'istanza database

Utilizza la procedura `rdsadmin.rdsadmin_transport_util.import_xtts_metadata` per importare i metadati delle tablespace trasportabili nell'istanza database RDS per Oracle. Durante l'operazione, lo stato dell'importazione dei metadati viene visualizzato nella tabella `rdsadmin.rds_xtts_operation_info`. Per ulteriori informazioni, consulta Passaggio 5: importazione dei metadati delle tablespace nell'istanza database di destinazione.

Sintassi

```
PROCEDURE import_xtts_metadata(
    p_datapump_metadata_file IN SYS.DBA_DATA_FILES.FILE_NAME%TYPE,
    p_directory_name         IN VARCHAR2,
    p_exclude_stats          IN BOOLEAN DEFAULT FALSE,
    p_remap_tablespace_list  IN CLOB DEFAULT NULL,
    p_remap_user_list        IN CLOB DEFAULT NULL);
```

Parametri

Nome del parametro	Tipo di dati	Default	Campo obbligatorio	Descrizione
`p_datapump_metadata_file`	`SYS.DBA_DATA_FILES.FILE_NAME%TYPE`	—	Sì	Il nome del file Oracle Data Pump che contiene i metadati per le tablespace trasportabili.
`p_directory_name`	`VARCHAR2`	—	Sì	La directory che contiene il file Data Pump.

Nome del parametro	Tipo di dati	Default	Campo obbligatorio	Descrizione
`p_exclude_stats`	`BOOLEAN`	`FALSE`	No	Flag che indica se escludere le statistiche.
`p_remap_tablespace_list`	`CLOB`	NULL	No	Un elenco delle tablespace che devono essere rimappate durante l'importazione dei metadati. Utilizzare il formato *`from_tbs`*`:`*`to_tbs`*. Ad esempio, specifica `users:user_data` .
`p_remap_user_list`	`CLOB`	NULL	No	Un elenco di schemi utente che devono essere rimappati durante l'importazione dei metadati. Utilizzare il formato *`from_schema_name`* `:`*`to_schema_name`* . Ad esempio, specifica `hr:human_resources` .

Esempi

L'esempio importa i metadati della tablespace dal file *`xttdump.dmp`*, che si trova nella directory *`DATA_PUMP_DIR`*.

```
BEGIN
  rdsadmin.rdsadmin_transport_util.import_xtts_metadata('xttdump.dmp','DATA_PUMP_DIR');
END;
/
```

Elenco dei file orfani dopo un'importazione della tablespace

Usa la procedura `rdsadmin.rdsadmin_transport_util.list_xtts_orphan_files` per elencare i file di dati che sono rimasti orfani dopo un'importazione della tablespace. Dopo aver identificato i file di dati, puoi eliminarli chiamando `rdsadmin.rdsadmin_transport_util.cleanup_incomplete_xtts_import`.

Sintassi

```
FUNCTION list_xtts_orphan_files RETURN xtts_orphan_files_list_t PIPELINED;
```

Esempi

L'esempio seguente esegue la procedura `rdsadmin.rdsadmin_transport_util.list_xtts_orphan_files`. L'output mostra due file di dati che sono rimasti orfani.

```
SQL> SELECT * FROM TABLE(rdsadmin.rdsadmin_transport_util.list_xtts_orphan_files);

FILENAME       FILESIZE
-------------- ---------
datafile_7.dbf 104865792
datafile_8.dbf 104865792
```

Eliminazione di file di dati rimasti orfani dopo un'importazione della tablespace

Usa la procedura `rdsadmin.rdsadmin_transport_util.list_xtts_orphan_files` per eliminare i file di dati che sono rimasti orfani dopo un'importazione della tablespace. L'esecuzione di questo comando genera un file di log che utilizza il formato del nome `rds-xtts-delete_xtts_orphaned_files-`*`YYYY-MM-DD.HH24-MI-SS.FF`*`.log` nella directory BDUMP. Usa la procedura `rdsadmin.rdsadmin_transport_util.cleanup_incomplete_xtts_import` per trovare i file rimasti orfani. Puoi leggere il file di log chiamando la procedura `rdsadmin.rds_file_util.read_text_file`. Per ulteriori informazioni, consulta Fase 6: rimozione dei file residui.

Sintassi

```
PROCEDURE cleanup_incomplete_xtts_import(
    p_directory_name IN VARCHAR2);
```

Parametri

Nome del parametro	Tipo di dati	Default	Campo obbligatorio	Descrizione
p_directory_name	VARCHAR2	—	Sì	La directory che contiene i file di dati rimasti orfani.

Esempi

Nell'esempio seguente, i file di dati rimasti orfani in *DATA_PUMP_DIR* vengono eliminati.

```
BEGIN
  rdsadmin.rdsadmin_transport_util.cleanup_incomplete_xtts_import('DATA_PUMP_DIR');
END;
/
```

L'esempio seguente legge il file di log generato dal comando precedente.

```
SELECT *
FROM TABLE(rdsadmin.rds_file_util.read_text_file(
       p_directory => 'BDUMP',
       p_filename  => 'rds-xtts-
delete_xtts_orphaned_files-2023-06-01.09-33-11.868894000.log'));

TEXT
--------------------------------------------------------------------------------
orphan transported datafile datafile_7.dbf deleted.
orphan transported datafile datafile_8.dbf deleted.
```

Configurazione delle funzionalità avanzate di RDS per Oracle

RDS per Oracle supporta varie funzionalità avanzate, tra cui HugePages, un archivio dell'istanza e i tipi di dati estesi.

Argomenti

Archiviazione di dati temporanei in un archivio dell'istanza RDS per Oracle

Utilizza un archivio dell'istanza per gli spazi di tabella temporanei e Database Smart Flash Cache (la cache flash) su classi di istanza database RDS per Oracle supportate.

Argomenti

Panoramica dell'archivio dell'istanza RDS per Oracle

Un archivio dell'istanza fornisce uno spazio di archiviazione temporaneo a livello di blocco per un'istanza database RDS per Oracle. È possibile utilizzare un archivio dell'istanza per l'archiviazione temporanea di informazioni che cambiano di frequente.

L'archivio dell'istanza è basato su dispositivi NVMe (Non-Volatile Memory Express) fisicamente collegati al computer host. Questo archivio è ottimizzato per bassa latenza, prestazioni I/O casuali e velocità di trasmissione effettiva di lettura sequenziale.

La dimensione dell'archivio dell'istanza varia in base al tipo di istanza database. Per ulteriori informazioni sull'archivio dell'istanza, consulta Instance store Amazon EC2 nella Guida per l'utente di Amazon Elastic Compute Cloud per istanze Linux.

Argomenti

- Tipi di dati nell'archivio dell'istanza RDS per Oracle
- Vantaggi dell'archivio dell'istanza RDS per Oracle
- Classi di istanza supportate per l'archivio dell'istanza RDS per Oracle
- Versioni del motore supportate per l'archivio dell'istanza RDS per Oracle
- Regioni AWS supportate per l'archivio dell'istanza RDS per Oracle
- Costo dell'archivio dell'istanza RDS per Oracle

Tipi di dati nell'archivio dell'istanza RDS per Oracle

In un archivio dell'istanza puoi inserire i seguenti tipi di dati temporanei RDS per Oracle:

Spazio di tabella temporaneo

Oracle Database utilizza gli spazi di tabella temporanei per archiviare i risultati delle query intermedie che non rientrano nella memoria. Le query più grandi possono generare grandi quantità di dati intermedi che devono essere temporaneamente memorizzati nella cache, ma non devono essere persistenti. In particolare, uno spazio di tabella temporaneo è utile per ordinamenti, aggregazioni di hash e join. Se l'istanza database RDS per Oracle utilizza Enterprise Edition o Standard Edition 2, puoi inserire uno spazio di tabella temporaneo in un archivio dell'istanza.

Cache flash

La cache flash migliora le prestazioni delle letture casuali a blocco singolo nel percorso convenzionale. È consigliabile dimensionare la cache per adattarla alla maggior parte dei set di dati attivi. Se l'istanza database RDS per Oracle utilizza Enterprise Edition, è possibile inserire la cache flash in un archivio dell'istanza.

Per impostazione predefinita, un archivio dell'istanza è configurato per uno spazio di tabella temporaneo ma non per la cache flash. Non è possibile inserire i file di dati Oracle e i file di registro del database in un archivio dell'istanza.

Vantaggi dell'archivio dell'istanza RDS per Oracle

Potresti prendere in considerazione l'utilizzo di un archivio dell'istanza per archiviare file e cache temporanei che puoi perdere. Se desideri migliorare le prestazioni del database o se un carico di lavoro in aumento causa problemi di prestazioni per l'archivio Amazon EBS, valuta la possibilità di dimensionarlo a una classe di istanza che supporti un archivio dell'istanza.

Posizionando lo spazio di tabella temporaneo e la cache flash in un archivio dell'istanza, si ottengono i seguenti vantaggi:

- Minori latenze di lettura
- Velocità di trasmissione effettiva più alta
- Carico ridotto sui volumi Amazon EBS
- Riduzione dei costi di archivio e snapshot grazie al carico ridotto di Amazon EBS
- Minore necessità di eseguire il provisioning di IOPS elevati, con conseguente riduzione dei costi complessivi

Posizionando lo spazio di tabella temporaneo nell'archivio dell'istanza, si ottiene un immediato incremento delle prestazioni per le query che utilizzano lo spazio temporaneo. Quando posizioni la cache flash nell'archivio dell'istanza, le letture dei blocchi memorizzati nella cache hanno in genere una latenza molto inferiore rispetto alle letture di Amazon EBS. La cache flash deve essere "riscaldata" prima di offrire vantaggi in termini di prestazioni. La cache si riscalda da sola perché il database scrive i blocchi nella cache flash man mano che escono dalla cache del buffer del database.

> **Note**
>
> In alcuni casi, la cache flash provoca un sovraccarico delle prestazioni a causa della gestione della cache. Prima di attivare la cache flash in un ambiente di produzione, ti consigliamo di analizzare il carico di lavoro e testare la cache in un ambiente di test.

Classi di istanza supportate per l'archivio dell'istanza RDS per Oracle

Amazon RDS supporta l'archivio dell'istanza per le seguenti classi di istanza database:

- db.m5d
- db.r5d

- db.x2idn
- db.x2iedn

RDS per Oracle supporta le classi di istanza database precedenti solo per il modello di licenza BYOL. Per ulteriori informazioni, consulta Classi di istanza RDS per Oracle supportate e Bring Your Own License (BYOL) per EE e SE2.

Per visualizzare lo spazio di archiviazione totale dell'istanza per i tipi di istanza database supportati, esegui il comando riportato di seguito in AWS CLI.

Example

```
aws ec2 describe-instance-types \
  --filters "Name=instance-type,Values=*5d.*large*" \
  --query "InstanceTypes[?contains(InstanceType,'m5d')||contains(InstanceType,'r5d')]
[InstanceType, InstanceStorageInfo.TotalSizeInGB]" \
  --output table
```

Il comando precedente restituisce la dimensione del dispositivo raw per l'archivio dell'istanza. RDS per Oracle utilizza una piccola parte di questo spazio per la configurazione. Lo spazio nell'archivio dell'istanza disponibile per gli spazi di tabella temporanei o per la cache flash è leggermente inferiore.

Versioni del motore supportate per l'archivio dell'istanza RDS per Oracle

L'archivio dell'istanza è supportato nelle seguenti versioni del motore RDS per Oracle:

- 21.0.0.0.ru-2022-01.rur-2022-01.r1 o versioni successive di Oracle Database 21c
- 19.0.0.0.ru-2020-10.rur-2020-10.r1 o versioni successive di Oracle Database 19c

Regioni AWS supportate per l'archivio dell'istanza RDS per Oracle

L'archivio dell'istanza è disponibile in tutte le Regioni AWS in cui sono supportati uno o più tipi di istanza. Per ulteriori informazioni sulle classi di istanza db.m5d e db.r5d, consulta Classi di istanze database. Per ulteriori informazioni sulle classi di istanza supportate da Amazon RDS per Oracle, consulta Classi di istanza RDS for Oracle.

Costo dell'archivio dell'istanza RDS per Oracle

Il costo dell'archivio dell'istanza è incluso nel costo delle istanze attivate dall'archivio dell'istanza. L'attivazione di un archivio dell'istanza in un'istanza database RDS per Oracle non comporta costi

aggiuntivi. Per ulteriori informazioni sulle istanze attivate dall'archivio dell'istanza, consulta Classi di istanza supportate per l'archivio dell'istanza RDS per Oracle.

Attivazione di un archivio dell'istanza RDS per Oracle

Per attivare l'archivio dell'istanza per i dati temporanei di RDS per Oracle, procedi in uno dei seguenti modi:

- Crea un'istanza database RDS per Oracle utilizzando una classe di istanza supportata. Per ulteriori informazioni, consulta Creazione di un'istanza database Amazon RDS.
- Modifica un'istanza database RDS per Oracle esistente per utilizzare una classe di istanza supportata. Per ulteriori informazioni, consulta Modifica di un'istanza database Amazon RDS.

Configurazione di un archivio dell'istanza RDS per Oracle

Per impostazione predefinita, il 100% dello spazio dell'archivio dell'istanza viene allocato allo spazio di tabella temporaneo. Per configurare l'archivio dell'istanza per allocare spazio alla cache flash e allo spazio di tabella temporaneo, imposta i seguenti parametri nel gruppo di parametri per l'istanza:

db_flash_cache_size={DBInstanceStore*{0,2,4,6,8,10}/10}

Questo parametro specifica la quantità di spazio di archiviazione allocata per la cache flash. Questo parametro è valido solo per Oracle Database Enterprise Edition. Il valore di default è `{DBInstanceStore*0/10}`. Se imposti un valore diverso da zero per `db_flash_cache_size`, l'istanza RDS per Oracle abilita la cache flash dopo il riavvio dell'istanza.

rds.instance_store_temp_size={DBInstanceStore*{0,2,4,6,8,10}/10}

Questo parametro specifica la quantità di spazio di archiviazione allocata per lo spazio di tabella temporaneo. Il valore di default è `{DBInstanceStore*10/10}`. Questo parametro è modificabile per Oracle Database Enterprise Edition e di sola lettura per Standard Edition 2. Se imposti un valore diverso da zero per `rds.instance_store_temp_size`, Amazon RDS alloca lo spazio nell'archivio dell'istanza per lo spazio di tabella temporaneo.

È possibile impostare i parametri `db_flash_cache_size` e `rds.instance_store_temp_size` per le istanze database che non utilizzano un archivio dell'istanza. In questo caso, entrambe le impostazioni restituiscono il risultato di `0`, che disattiva la caratteristica. In questo caso, puoi utilizzare lo stesso gruppo di parametri per istanze di

dimensioni diverse e per istanze che non utilizzano un archivio dell'istanza. Se modifichi questi parametri, assicurati di riavviare le istanze associate in modo che le modifiche abbiano effetto.

> **Important**
>
> Se allochi spazio per uno spazio di tabella temporaneo, Amazon RDS non crea automaticamente lo spazio di tabella temporaneo. Per informazioni su come creare lo spazio di tabella temporaneo nell'archivio dell'istanza, consulta Creazione di un spazio di tabella temporaneo nell'archivio dell'istanza.

Il valore combinato dei parametri precedenti non deve superare 10/10 o 100%. La tabella seguente illustra le impostazioni valide e non valide dei parametri.

Impostazione db_flash_cache_size	Impostazione rds.instance_store_temp_size	Spiegazione
db_flash_cache_size={DBInstanceStore*0/10}	rds.instance_store_temp_size={DBInstanceStore*10/10}	Questa è una configurazione valida per tutte le edizioni di Oracle Database. Amazon RDS alloca il 100% dello spazio dell'archivio dell'istanza allo spazio di tabella temporaneo. Questa è l'impostazione predefinita.
db_flash_cache_size={DBInstanceStore*10/10}	rds.instance_store_temp_size={DBInstanceStore*0/10}	Questa configurazione è valida solo

Impostazione db_flash_cache_size	Impostazione rds.instance_store_temp_size	Spiegazione
		per Oracle Database Enterprise Edition. Amazon RDS alloca il 100% dello spazio dell'archivio dell'istanza alla cache flash.
db_flash_cache_size={DBInstanceStore*2/10}	rds.instance_store_temp_size={DBInstanceStore*8/10}	Questa configurazione è valida solo per Oracle Database Enterprise Edition. Amazon RDS alloca il 20% dello spazio dell'archivio dell'istanza alla cache flash e l'80% dello spazio dell'archivio dell'istanza allo spazio di tabella temporaneo.

Impostazione db_flash_cache_size	Impostazione rds.instance_store_temp_size	Spiegazione
db_flash_cache_size={DBInstanceStore*6/10}	rds.instance_store_temp_size={DBInstanceStore*4/10}	Questa configurazione è valida solo per Oracle Database Enterprise Edition. Amazon RDS alloca il 60% dello spazio dell'archivio dell'istanza alla cache flash e il 40% dello spazio dell'archivio dell'istanza allo spazio di tabella temporaneo.

Impostazione db_flash_cache_size	Impostazione rds.instance_store_temp_size	Spiegazione
db_flash_cache_size={DBInstanceStore*2/10}	rds.instance_store_temp_size={DBInstanceStore*4/10}	Questa configurazione è valida solo per Oracle Database Enterprise Edition. Amazon RDS alloca il 20% dello spazio dell'archivio dell'istanza alla cache flash e il 40% dello spazio dell'archivio dell'istanza allo spazio di tabella temporaneo.
db_flash_cache_size={DBInstanceStore*8/10}	rds.instance_store_temp_size={DBInstanceStore*8/10}	Questa configurazione non è valida perché la percentuale combinata di spazio dell'archivio dell'istanza supera il 100%. In questi casi, il tentativo di Amazon RDS non riesce.

Considerazioni sulla modifica del tipo di istanza database

La modifica del tipo di istanza database può influire sulla configurazione della cache flash o dello spazio di tabella temporaneo nell'archivio dell'istanza. Considera le seguenti modifiche e gli effetti:

Aumento o riduzione dell'istanza database che supporta l'archivio dell'istanza.

I seguenti valori aumentano o diminuiscono proporzionalmente alla nuova dimensione dell'archivio dell'istanza:

- La nuova dimensione della cache flash.
- Lo spazio allocato agli spazi di tabella temporanei che risiedono nell'archivio dell'istanza.

Ad esempio, l'impostazione `db_flash_cache_size={DBInstanceStore*6/10}` su un'istanza db.m5d.4xlarge fornisce circa 340 GB di spazio nella cache flash. Se aumenti il tipo di istanza a db.m5d.8xlarge, lo spazio della cache flash aumenta fino a circa 680 GB.

Modifica di un'istanza database che non utilizza un archivio dell'istanza in un'istanza che ne utilizza uno.

Se `db_flash_cache_size` è impostato su un valore maggiore di `0`, la cache flash è configurata. Se `rds.instance_store_temp_size` è impostato su un valore maggiore di `0`, lo spazio dell'archivio dell'istanza viene allocato per essere utilizzato da una spazio di tabella temporaneo. RDS per Oracle non sposta automaticamente i file temporanei nell'archivio dell'istanza. Per informazioni sull'utilizzo dello spazio allocato, consulta Creazione di un spazio di tabella temporaneo nell'archivio dell'istanza o Aggiunta di un file temporaneo all'archivio dell'istanza in una replica di lettura.

Modifica di un'istanza database che utilizza un archivio dell'istanza in un'istanza che non ne utilizza uno.

In questo caso, RDS per Oracle rimuove la cache flash. RDS ricrea il file temporaneo che si trova attualmente nell'archivio dell'istanza su un volume Amazon EBS. La dimensione massima del nuovo file temporaneo è la dimensione precedente del parametro `rds.instance_store_temp_size`.

Utilizzo di un archivio dell'istanza in una replica di lettura Oracle

Le repliche di lettura supportano la cache flash e gli spazi di tabella temporanei in un archivio dell'istanza. Sebbene la cache flash funzioni allo stesso modo dell'istanza database primaria, nota le seguenti differenze per gli spazi di tabella temporanei:

- Non è possibile creare uno spazio di tabella temporaneo in una replica di lettura. Se crei un nuovo spazio di tabella temporaneo sull'istanza primaria, RDS per Oracle replica le informazioni nello spazio di tabella senza file temporanei. Per aggiungere un nuovo file temporaneo, utilizza una delle seguenti tecniche:
  - Usa la procedura Amazon RDS `rdsadmin.rdsadmin_util.add_inst_store_tempfile`. RDS per Oracle crea un file temporaneo nell'archivio dell'istanza della replica di lettura e lo aggiunge alla spazio di tabella temporaneo specificato.
  - Esegui il comando `ALTER TABLESPACE … ADD TEMPFILE`. RDS per Oracle inserisce il file temporaneo nell'archivio Amazon EBS.

  > **Note**
  >
  > Le dimensioni dei file temporanei e i tipi di archivio possono essere diversi nell'istanza database primaria e nella replica di lettura.

- È possibile gestire l'impostazione predefinita della spazio di tabella temporaneo solo nell'istanza database primaria. RDS per Oracle replica l'impostazione in tutte le repliche di lettura.
- È possibile configurare i gruppi di spazi di tabella temporanei solo nell'istanza database primaria. RDS per Oracle replica l'impostazione in tutte le repliche di lettura.

Configurazione di un gruppo di spazi di tabella temporanei in un archivio dell'istanza e in Amazon EBS

Puoi configurare un gruppo di spazi di tabella temporanei per includere spazi di tabella temporanei sia in un archivio dell'istanza che in Amazon EBS. Questa tecnica è utile quando si desidera uno spazio di archiviazione temporaneo superiore a quello consentita dall'impostazione massima di `rds.instance_store_temp_size`.

Quando configuri un gruppo di spazi di tabella temporanei in un archivio dell'istanza e in Amazon EBS, i due spazi di tabella hanno caratteristiche prestazionali significativamente diverse. Oracle Database sceglie lo spazio di tabella per servire le query in base a un algoritmo interno. Pertanto, query simili possono variare in termini di prestazioni.

In genere, si crea una spazio di tabella temporaneo nell'archivio dell'istanza come segue:

1. Crea una spazio di tabella temporaneo nell'archivio dell'istanza.
2. Imposta il nuovo spazio di tabella come spazio di tabella temporaneo predefinito del database.

Se la dimensione dello spazio di tabella nell'archivio dell'istanza è insufficiente, puoi creare uno spazio di archiviazione temporaneo aggiuntivo come segue:

1. Assegna lo spazio di tabella temporaneo nell'archivio dell'istanza a un gruppo di spazi di tabella temporanei.
2. Crea un nuovo spazio di tabella temporaneo in Amazon EBS se non ne esiste uno.
3. Assegna lo spazio di tabella temporaneo in Amazon EBS allo stesso gruppo di spazi di tabella che include lo spazio di tabella dell'archivio dell'istanza.
4. Imposta il gruppo di spazi di tabella come spazio di tabella temporaneo predefinito.

L'esempio seguente presuppone che la dimensione della spazio di tabella temporaneo nell'archivio dell'istanza non soddisfi i requisiti dell'applicazione. L'esempio crea lo spazio di tabella temporaneo `temp_in_inst_store` nell'archivio dell'istanza, lo assegna al gruppo di spazi di tabella `temp_group`, aggiunge lo spazio di tabella Amazon EBS esistente denominato `temp_in_ebs` a questo gruppo e imposta il gruppo come spazio di tabella temporaneo predefinito.

```
SQL> EXEC rdsadmin.rdsadmin_util.create_inst_store_tmp_tblspace('temp_in_inst_store');

PL/SQL procedure successfully completed.

SQL> ALTER TABLESPACE temp_in_inst_store TABLESPACE GROUP temp_group;

Tablespace altered.

SQL> ALTER TABLESPACE temp_in_ebs TABLESPACE GROUP temp_group;

Tablespace altered.

SQL> EXEC rdsadmin.rdsadmin_util.alter_default_temp_tablespace('temp_group');

PL/SQL procedure successfully completed.

SQL> SELECT * FROM DBA_TABLESPACE_GROUPS;

GROUP_NAME                     TABLESPACE_NAME
------------------------------ ------------------------------
TEMP_GROUP                     TEMP_IN_EBS
TEMP_GROUP                     TEMP_IN_INST_STORE
```

```
SQL> SELECT PROPERTY_VALUE FROM DATABASE_PROPERTIES WHERE
 PROPERTY_NAME='DEFAULT_TEMP_TABLESPACE';

PROPERTY_VALUE
--------------
TEMP_GROUP
```

Rimozione di un archivio dell'istanza RDS per Oracle

Per rimuovere l'archivio dell'istanza, modifica l'istanza database RDS per Oracle per utilizzare un tipo di istanza che non supporta l'archivio dell'istanza, ad esempio db.m5 o db.r5.

Attivazione di HugePages per un'istanza RDS per Oracle

Amazon RDS for Oracle supporta Huge Pages del kernel di Linux per una maggiore scalabilità del database. HugePages restituisce tabelle di pagina più piccole e meno tempo CPU dedicato alla gestione della memoria, migliorando così le prestazioni di istanze database di grosse dimensioni. Per ulteriori informazioni, consulta Overview of HugePages nella documentazione Oracle.

Puoi utilizzare HugePages con tutte le versioni e le edizioni supportate di RDS per Oracle.

Il parametro `use_large_pages` controlla se HugePages è attivato per un'istanza database. Le impostazioni possibili per questo parametro sono `ONLY`, `FALSE` e `{DBInstanceClassHugePagesDefault}`. Il parametro `use_large_pages` è impostato su `{DBInstanceClassHugePagesDefault}` nel gruppo di parametri database predefinito per Oracle.

Per controllare se HugePages viene attivato automaticamente per un'istanza database, puoi utilizzare la variabile di formula `DBInstanceClassHugePagesDefault` nei gruppi di parametri. Il valore è determinato nel modo seguente:

- Per le classi di istanza database menzionate nella tabella seguente, `DBInstanceClassHugePagesDefault` restituisce sempre `FALSE` per impostazione predefinita e `use_large_pages` restituisce `FALSE`. Puoi attivare HugePages manualmente per queste classi di istanza database se la classe di istanza database dispone di almeno 14 GB di memoria.
- Per le classi di istanza database non menzionate nella tabella seguente, se la classe dell'istanza database dispone di meno di 14 GiB di memoria, `DBInstanceClassHugePagesDefault` restituisce sempre `FALSE`. Inoltre, `use_large_pages` restituisce `FALSE`.
- Per le classi di istanza database non menzionate nella tabella seguente, se la classe dell'istanza dispone di almeno di 14 GB e fino a 100 GB di memoria, `DBInstanceClassHugePagesDefault`

restituisce TRUE per impostazione predefinita. Inoltre, `use_large_pages` restituisce `ONLY`. Puoi disattivare HugePages manualmente impostando `use_large_pages` su `FALSE`.

- Per le classi di istanza database non menzionate nella tabella seguente, se la classe dell'istanza dispone di almeno 100 GiB di memoria, `DBInstanceClassHugePagesDefault` restituisce sempre `TRUE`. Inoltre, `use_large_pages` restituisce `ONLY` e HugePages non può essere disabilitato.

Per impostazione predefinita HugePages non è attivato per le seguenti classi di istanza database.

Famiglia di classi di istanza database	Classi di istanza database con HugePages non attivato per impostazione predefinita.
db.m5	db.m5.large
db.m4	db.m4.large, db.m4.xlarge, db.m4.2xlarge, db.m4.4xlarge, db.m4.10xlarge
db.t3	db.t3.micro, db.t3.small, db.t3.medium, db.t3.large

Per altre informazioni sulle classi di istanza database, consulta Specifiche hardware per le classi di istanza database .

Per attivare manualmente HugePages per le istanze database nuove o esistenti, imposta il parametro `use_large_pages` su `ONLY`. Non è possibile utilizzare Huge Pages con Automatic Memory Management (AMM) di Oracle. Se viene impostato il parametro `use_large_pages` su `ONLY`, è necessario anche impostare `memory_target` e `memory_max_target` su `0`. Per ulteriori informazioni sull'impostazione dei parametri database per l'istanza database, consulta Utilizzo di gruppi di parametri.

È possibile anche impostare i parametri `sga_target`, `sga_max_size` e `pga_aggregate_target`. Quando imposti i parametri di memoria della SGA (Area globale del sistema) e della PGA (Area globale del programma), aggiungi i valori insieme. Sottrai questo totale dalla memoria di istanza disponibile (`DBInstanceClassMemory`) per determinare la memoria libera al di là dell'allocazione Huge Pages. È necessario lasciare almeno 2 GiB di memoria libera o 10% del totale della memoria di istanza disponibile, scegliendo la più piccola.

Dopo aver configurato i parametri, è necessario riavviare la tua istanza database per rendere effettive le modifiche. Per ulteriori informazioni, consulta Riavvio di un'istanza database.

> **Note**
>
> L'istanza di Oracle DB rinvia le modifiche ai parametri di inizializzazione relativi a SGA fino al riavvio dell'istanza senza failover. Nella console Amazon RDS scegliere Riavvia ma non scegliere Riavvia con failover. In AWS CLI, chiamare il comando `reboot-db-instance` con il parametro `--no-force-failover`. L'istanza database non elabora i parametri relativi a SGA durante il failover o durante altre operazioni di manutenzione che causano il riavvio dell'istanza.

Di seguito trovi una configurazione di esempio di parametri per Huge Pages che abilita manualmente Huge Pages. È necessario impostare i valori per soddisfare le esigenze specifiche.

```
memory_target            = 0
memory_max_target        = 0
pga_aggregate_target     = {DBInstanceClassMemory*1/8}
sga_target               = {DBInstanceClassMemory*3/4}
sga_max_size             = {DBInstanceClassMemory*3/4}
use_large_pages          = ONLY
```

Supponi che i seguenti valori dei parametri siano impostati in un gruppo di parametri.

```
memory_target            = IF({DBInstanceClassHugePagesDefault}, 0,
 {DBInstanceClassMemory*3/4})
memory_max_target        = IF({DBInstanceClassHugePagesDefault}, 0,
 {DBInstanceClassMemory*3/4})
pga_aggregate_target     = IF({DBInstanceClassHugePagesDefault},
 {DBInstanceClassMemory*1/8}, 0)
sga_target               = IF({DBInstanceClassHugePagesDefault},
 {DBInstanceClassMemory*3/4}, 0)
sga_max_size             = IF({DBInstanceClassHugePagesDefault},
 {DBInstanceClassMemory*3/4}, 0)
use_large_pages          = {DBInstanceClassHugePagesDefault}
```

Il gruppo di parametri viene utilizzato da una classe di istanza database db.r4 con meno di 100 GiB di memoria. Con queste impostazioni di parametri e `use_large_pages` impostato su `{DBInstanceClassHugePagesDefault}`, HugePages viene attivato per l'istanza db.r4.

Valuta un altro esempio con i seguenti valori dei parametri impostati in un gruppo di parametri.

```
memory_target           = IF({DBInstanceClassHugePagesDefault}, 0,
 {DBInstanceClassMemory*3/4})
memory_max_target       = IF({DBInstanceClassHugePagesDefault}, 0,
 {DBInstanceClassMemory*3/4})
pga_aggregate_target    = IF({DBInstanceClassHugePagesDefault},
 {DBInstanceClassMemory*1/8}, 0)
sga_target              = IF({DBInstanceClassHugePagesDefault},
 {DBInstanceClassMemory*3/4}, 0)
sga_max_size            = IF({DBInstanceClassHugePagesDefault},
 {DBInstanceClassMemory*3/4}, 0)
use_large_pages         = FALSE
```

Il gruppo di parametri è utilizzato da una classe di istanze database db.r4 e una classe di istanza database db.r5 con meno di 100 GiB di memoria. Con queste impostazioni di parametri, HugePages viene disattivato nelle istanze db.r4 e db.r5.

> **Note**
>
> Se questo gruppo di parametri è utilizzato da una classe di istanza database db.r4 o una classe di istanza database db.r5 con almeno 100 GiB di memoria, l'impostazione `FALSE` per `use_large_pages` è sovrascritta e impostata su `ONLY`. In questo caso, viene inviata una notifica al cliente a proposito della sostituzione.

Dopo che Huge Pages è attivo sull'istanza database, puoi visualizzare l'informazione su Huge Pages abilitando il monitoraggio avanzato. Per ulteriori informazioni, consulta Monitoraggio dei parametri del sistema operativo con il monitoraggio avanzato.

Attivazione dei tipi di dati estesi in RDS per Oracle

Amazon RDS per Oracle supporta i tipi di dati estesi. Con i tipi di dati estesi, le dimensioni massime per i tipi di dati `VARCHAR2`, `NVARCHAR2` e `RAW` sono di 32.767 byte. Per utilizzare i tipi di dati estesi, imposta il parametro `MAX_STRING_SIZE` su `EXTENDED`. Per ulteriori informazioni, consulta la pagina sui tipi di dati estesi nella documentazione Oracle.

Se non desideri utilizzare i tipi di dati estesi, mantieni il parametro MAX_STRING_SIZE impostato su STANDARD (impostazione predefinita). In questo caso, i limiti di dimensione sono di 4.000 byte per i tipi di dati VARCHAR2 e NVARCHAR2 e 2.000 byte per il tipo di dati RAW.

Puoi attivare i tipi di dati estesi nelle istanze database nuove o esistenti. Per le nuove istanze database, il tempo di creazione di un'istanza database è in genere più lungo quando i tipi di dati estesi sono attivati. Per le istanze database esistenti, l'istanza database non può essere utilizzata durante il processo di conversione.

Considerazioni sui tipi di dati estesi

Considera quanto segue quando abiliti i tipi di dati estesi per la tua istanza database:

- Quando attivi i tipi di dati estesi, non puoi ripristinare l'utilizzo delle dimensioni standard per i tipi di dati nell'istanza database. Dopo aver convertito un'istanza database per utilizzare i tipi di dati estesi, se si imposta il parametro MAX_STRING_SIZE di nuovo su STANDARD, viene restituito lo stato `incompatible-parameters`.
- Quando si ripristina un'istanza database che utilizza i tipi di dati estesi, è necessario specificare un gruppo di parametri con il parametro MAX_STRING_SIZE impostato su EXTENDED. Durante il ripristino, se si specifica il gruppo di parametri predefinito o altri gruppi di parametri con MAX_STRING_SIZE impostato su STANDARD, viene restituito lo stato `incompatible-parameters`.
- Quando lo stato dell'istanza database è `incompatible-parameters` a causa dell'impostazione MAX_STRING_SIZE, l'istanza database resta non disponibile finché non si imposta il parametro MAX_STRING_SIZE su EXTENDED e non si riavvia l'istanza.
- Consigliamo di non attivare i tipi di dati estesi per le istanze database Oracle in esecuzione nella classe di istanze database t2.micro.

Attivazione dei tipi di dati estesi per una nuova istanza database

Per attivare i tipi di dati estesi per una nuova istanza database

1. Impostare il parametro MAX_STRING_SIZE su EXTENDED in un gruppo di parametri.

   Per impostare il parametro, creare un nuovo gruppo di parametri o modificarne uno esistente.

   Per ulteriori informazioni, consulta Utilizzo di gruppi di parametri.
2. Crea una nuova istanza database RDS per Oracle

Per ulteriori informazioni, consulta Creazione di un'istanza database Amazon RDS.

3. Associa il gruppo di parametri con `MAX_STRING_SIZE` impostato su `EXTENDED` all'istanza database.

   Per ulteriori informazioni, consulta Creazione di un'istanza database Amazon RDS.

Attivazione dei tipi di dati estesi per un'istanza database esistente

Dopo aver modificato un'istanza database in modo da attivare i tipi di dati estesi, RDS converte i dati nel database in modo che vengano utilizzate le dimensioni estese. La conversione e il tempo di inattività si verificano al successivo riavvio del database dopo la modifica del parametro. L'istanza database non è disponibile durante la conversione.

La durata dell'operazione dipende dalla classe di istanza database utilizzata dall'istanza database e dalle dimensioni del database. Per ridurre i tempi di inattività, prendi in considerazione la possibilità di creare un'istantanea immediatamente prima del riavvio. In questo modo si riduce la durata del backup durante il flusso di lavoro di conversione.

> **Note**
>
> Dopo aver attivato i tipi di dati estesi, non puoi effettuare un ripristino point-in-time durante la conversione. Questo ripristino può essere effettuato subito prima o dopo la conversione.

Per attivare i tipi di dati estesi per un'istanza database esistente

1. Acquisire uno snapshot del database

   Se nel database sono presenti oggetti non validi, Amazon RDS prova a ricompilarli. La conversione ai tipi di dati estesi può non riuscire se Amazon RDS non riesce a ricompilare un oggetto non valido. La snapshot consente di ripristinare il database in caso di problemi con la conversione. Controllare sempre la presenza di oggetti non validi prima della conversione e, se ci sono, correggerli o eliminarli. Per i database di produzione, consigliamo di provare il processo di conversione prima su una copia dell'istanza database.

   Per ulteriori informazioni, consulta Creazione di uno snapshot DB per un'istanza DB Single-AZ.

2. Impostare il parametro `MAX_STRING_SIZE` su `EXTENDED` in un gruppo di parametri.

Per impostare il parametro, creare un nuovo gruppo di parametri o modificarne uno esistente.

Per ulteriori informazioni, consulta Utilizzo di gruppi di parametri.

3. Modificare l'istanza database per associarla al gruppo di parametri con `MAX_STRING_SIZE` impostato su `EXTENDED`.

   Per ulteriori informazioni, consulta Modifica di un'istanza database Amazon RDS.

4. Riavviare l'istanza database per applicare la modifica al parametro.

   Per ulteriori informazioni, consulta Riavvio di un'istanza database.

Importazione di dati in Oracle in Amazon RDS

La modalità di importazione dei dati in un'istanza database Amazon RDS per Oracle dipende da quanto segue:

- La quantità di dati disponibili
- Il numero di oggetti di database esistenti nel database
- La varietà di oggetti di database esistenti nel database

Ad esempio, puoi usare i seguenti strumenti, a seconda delle esigenze:

- Oracle SQL Developer: importazione di un database semplice da 20 MB.
- Oracle Data Pump: importazione di database complessi o database con dimensioni di diverse centinaia di megabyte o diversi terabyte. Ad esempio, è possibile trasportare tablespace da un database on-premise all'istanza database RDS per Oracle. È possibile usare Amazon S3 o Amazon EFS per trasferire i file di dati e i metadati. Per ulteriori informazioni, consulta Migrazione utilizzando le tablespace trasportabili Oracle, Integrazione Amazon EFS e Integrazione Amazon S3.
- AWS Database Migration Service (AWS DMS) — Migrazione dei database senza tempi di inattività. Per ulteriori informazioni AWS DMS, consulta What is AWS Database Migration Service e il post di blog Migrazione dei database Oracle con tempi di inattività quasi pari a zero utilizzando DMS. AWS

> **Important**
>
> Prima di utilizzare le tecniche di migrazione precedentemente descritte, è consigliabile eseguire il backup del database. Dopo avere importato i dati, puoi eseguire il backup delle istanze database RDS per Oracle creando snapshot. Successivamente, è possibile ripristinare gli snapshot. Per ulteriori informazioni, consulta Backup, ripristino ed esportazione dei dati.

Per molti motori di database, la replica continua può continuare fino a quando non si è pronti per passare al database di destinazione. È possibile utilizzare AWS DMS per migrare a RDS for Oracle dallo stesso motore di database o da un motore diverso. Se si esegue la migrazione da un motore di database diverso, è possibile utilizzare il AWS Schema Conversion Tool per migrare gli oggetti dello schema che AWS DMS non eseguono la migrazione.

Argomenti

Importazione utilizzando Oracle SQL Developer

Oracle SQL Developer è uno strumento grafico Java distribuito gratuitamente da Oracle. SQL Developer fornisce opzioni per la migrazione dei dati tra due database Oracle oppure per la migrazione dei dati da altri database, come MySQL, a Oracle. Questo strumento è ideale per la migrazione di database di piccole dimensioni.

Puoi installare questo strumento nel computer desktop (Windows, Linux o Mac) o in uno dei server. Dopo avere installato SQL Developer, puoi utilizzarlo per connetterti ai database di origine e di destinazione. Utilizza il comando Copia del database nel menu Strumenti per copiare i dati nell'istanza DB di RDS for Oracle.

Per scaricare SQL Developer, vai alla pagina http://www.oracle.com/technetwork/developer-tools/sql-developer.

Prima di iniziare la migrazione dei dati, è consigliabile leggere la documentazione di Oracle SQL Developer. Oracle fornisce anche la documentazione relativa alla migrazione da altri database, tra cui MySQL e SQL Server. Per ulteriori informazioni, consulta http://www.oracle.com/technetwork/database/migration nella documentazione di Oracle.

Migrazione utilizzando le tablespace trasportabili Oracle

È possibile utilizzare la funzionalità Oracle Tablespace trasportabile per copiare un set di tablespace da un database Oracle on-premise a un'istanza RDS per Oracle DB. A livello fisico, trasferisci file di dati di origine e file di metadati all'istanza DB di destinazione utilizzando Amazon EFS o Amazon S3. La funzionalità tablespace trasportabili utilizza il pacchetto. `rdsadmin.rdsadmin_transport_util` Per la sintassi e la semantica di questo pacchetto, vedi. Trasporto di tablespace

Per i post di blog che spiegano come trasportare le tablespace, consulta Migrare i database Oracle verso l'utilizzo di tablespace trasportabili e Amazon RDS for Oracle Transportable Tablespaces AWS usando RMAN.

Argomenti

Panoramica delle tablespace trasportabili Oracle

Un set di tablespace trasportabili è composto da file di dati per il set di tablespace da trasportare e da un file di dump di esportazione contenente i metadati delle tablespace. In una soluzione di migrazione fisica come le tablespace trasportabili, si trasferiscono file fisici, ovvero file di dati, file di configurazione e file di dump di Data Pump.

Argomenti

Vantaggi e svantaggi delle tablespace trasportabili

Si consiglia di utilizzare tablespace trasportabili quando è necessario eseguire la migrazione di una o più tablespace di grandi dimensioni su RDS con tempi di inattività minimi. Rispetto alla migrazione logica, le tablespace trasportabili offrono i seguenti vantaggi:

- I tempi di inattività sono inferiori rispetto alla maggior parte delle altre soluzioni di migrazione Oracle.
- Poiché la funzionalità Tablespace trasportabile copia solo i file fisici, vengono evitati gli errori di integrità dei dati e il danneggiamento logico che si possono verificare nella migrazione logica.

- Non è richiesta alcuna licenza aggiuntiva.
- È possibile eseguire la migrazione di un set di tablespace su diverse piattaforme e tipi di endianità, ad esempio, da una piattaforma Oracle Solaris a Linux. Tuttavia, il trasporto di tablespace da e verso i server Windows non è supportato.

> **Note**
>
> Linux è completamente testato e supportato. Non tutte le varianti di UNIX sono state testate.

Se vengono utilizzate tablespace trasportabili, è possibile trasportare i dati utilizzando Amazon S3 o Amazon EFS:

- In caso di utilizzo di EFS, i backup rimangono nel file system EFS per tutta la durata dell'importazione. È possibile rimuovere i file in seguito. Con questa tecnica, non è necessario effettuare il provisioning dello spazio di archiviazione EBS per l'istanza database. Per questo motivo, è consigliabile utilizzare Amazon EFS anziché S3. Per ulteriori informazioni, consulta Integrazione Amazon EFS.
- In caso di utilizzo di S3, è sufficiente scaricare i backup RMAN nello spazio di archiviazione EBS associato all'istanza database. I file rimangono nello spazio di archiviazione EBS durante l'importazione. Dopo l'importazione, è possibile liberare questo spazio, che rimane allocato all'istanza database.

Lo svantaggio principale delle tablespace trasportabili è dato dal fatto che è necessaria una conoscenza relativamente avanzata del database Oracle. Per ulteriori informazioni, consulta l'argomento relativo al trasporto delle tablespace tra database nel manuale Oracle Database Administrator's Guide (Guida per gli amministratori di Oracle Database).

Limitazioni relative alle tablespace trasportabili

Le limitazioni di Oracle Database per le tablespace trasportabili si applicano se si utilizza questa funzionalità in RDS per Oracle. Per ulteriori informazioni, consulta l'argomento relativo alle limitazioni delle tablespace trasportabili e alle limitazioni generali relative al trasporto dei dati nel manuale Oracle Database Administrator's Guide. È opportuno considerare le seguenti limitazioni aggiuntive per le tablespace trasportabili in RDS per Oracle:

- Né il database di origine né quello di destinazione possono utilizzare Standard Edition 2 (SE2). È supportata solo la versione Enterprise Edition.
- Non è possibile utilizzare un database Oracle Database 11g come origine. La funzionalità di tablespace trasportabili multipiattaforma di RMAN si basa sul meccanismo di trasporto RMAN, che Oracle Database 11g non supporta.
- Non è possibile eseguire la migrazione dei dati da un'istanza database RDS per Oracle utilizzando le tablespace trasportabili. È possibile utilizzare solo tablespace trasportabili per eseguire la migrazione dei dati su un'istanza database RDS per Oracle.
- Il sistema operativo Windows non è supportato.
- Non è possibile effettuare il trasporto delle tablespace in un database di una versione inferiore. Il database di destinazione deve essere della stessa versione o di una versione successiva rispetto al database di origine. Ad esempio, non è possibile effettuare il trasporto delle tablespace da Oracle Database 21c a Oracle Database 19c.
- Non è possibile effettuare il trasporto delle tablespace amministrative, ad esempio `SYSTEM` e `SYSAUX`.
- Non è possibile trasportare oggetti non contenenti dati come pacchetti PL/SQL, classi Java, viste, trigger, sequenze, utenti, ruoli e tabelle temporanee. Per trasportare oggetti non contenenti dati, creali manualmente o utilizza l'esportazione e l'importazione dei metadati di Data Pump. Per ulteriori informazioni, vedere My Oracle Support Note 1454872.1.
- Non è possibile effettuare il trasporto di tablespace crittografate o utilizzare colonne crittografate.
- In caso di trasferimento di file utilizzando Amazon S3, la dimensione massima supportata è di 5 TiB.
- Se il database di origine utilizza opzioni Oracle come Spatial (Spaziale), non è possibile effettuare il trasporto delle tablespace a meno che le stesse opzioni non siano configurate anche nel database di destinazione.
- Non è possibile effettuare il trasporto delle tablespace in un'istanza database RDS per Oracle in una configurazione di replica Oracle. Come soluzione alternativa, è possibile eliminare tutte le repliche, effettuare il trasporto delle tablespace e quindi ricreare le repliche.

Prerequisiti per le tablespace trasportabili

Prima di iniziare, completa le seguenti attività:

- Fare riferimento ai requisiti per le tablespace trasportabili descritti nei seguenti documenti in My Oracle Support:

  - Reduce Transportable Tablespace Downtime using Cross Platform Incremental Backup (Doc ID 2471245.1) (Riduzione dei tempi di inattività delle tablespace trasportabili utilizzando il backup incrementale multipiattaforma [ID documento 2471245.1])
  - Transportable Tablespace (TTS) Restrictions and Limitations: Details, Reference, and Version Where Applicable (Doc ID 1454872.1) (Restrizioni e limitazioni relative alle tablespace trasportabili (TTS): dettagli, riferimento e versione laddove applicabile [ID documento 1454872.1])
  - Primary Note for Transportable Tablespaces (TTS) – Common Questions and Issues (Doc ID 1166564.1) (Nota principale per le tablespace trasportabili (TTS) – Domande e problemi comuni [ID documento 1166564.1])
- Piano per la conversione endianness. Se specifichi l'ID della piattaforma di origine, RDS per Oracle converte automaticamente l'endianness. Per informazioni su come trovare gli ID di piattaforma, consulta Data Guard Support for Heterogeneous Primary and Physical Standbys in Same Data Guard Configuration (Doc ID 413484.1).
- Assicurarsi che la funzionalità Tablespace trasportabile sia abilitata nell'istanza database di destinazione. La funzionalità è abilitata solo se non vengono visualizzati errori `ORA-20304` quando si esegue la seguente query:

  ```
  SELECT * FROM TABLE(rdsadmin.rdsadmin_transport_util.list_xtts_orphan_files);
  ```

  Se la funzionalità Tablespace trasportabile non è abilitata, riavviare l'istanza database. Per ulteriori informazioni, consulta Riavvio di un'istanza database.
- Se si prevede di trasferire file utilizzando Amazon S3, procedere come segue:
  - Assicurati che sia disponibile un bucket Amazon S3 per i trasferimenti di file e che il bucket Amazon S3 si trovi nella stessa AWS regione dell'istanza DB. Per istruzioni, consultare Creazione di un bucket nella Guida introduttiva di Amazon Simple Storage Service.
  - Preparare il bucket Amazon S3 per l'integrazione con Amazon RDS seguendo le istruzioni in Configurazione delle autorizzazioni IAM per l'integrazione di RDS per Oracle con Amazon S3.
- Se si prevede di trasferire file utilizzando Amazon EFS, assicurarsi di aver configurato EFS secondo le istruzioni riportate in Integrazione Amazon EFS.
- È vivamente consigliabile di attivare i backup automatici nell'istanza database di destinazione. Poiché la fase di importazione dei metadati può potenzialmente non riuscire, è importante poter ripristinare lo stato dell'istanza database precedente all'importazione, evitando così la necessità di eseguire nuovamente il backup, il trasferimento e l'importazione delle tablespace.

Fase 1: configurazione dell'host di origine

In questo passaggio, si copiano gli script delle tablespace di trasporto forniti da My Oracle Support e si impostano i file di configurazione necessari. Nei passaggi seguenti, l'host di origine esegue il database contenente le tablespace da trasportare nell'istanza di destinazione.

Configurazione dell'host di origine

1. Accedere all'host di origine come proprietario della Oracle home.
2. Assicurarsi che le variabili di ambiente `ORACLE_HOME` e `ORACLE_SID` puntino al database di origine.
3. Accedere al database come amministratore e verificare che la versione del fuso orario, il set di caratteri del database e il set di caratteri nazionali siano gli stessi del database di destinazione.

```
SELECT * FROM V$TIMEZONE_FILE;
SELECT * FROM NLS_DATABASE_PARAMETERS
  WHERE PARAMETER IN ('NLS_CHARACTERSET','NLS_NCHAR_CHARACTERSET');
```

4. Configurare l'utilità Tablespace trasportabile come descritto in Oracle Support note 2471245.1 (Nota del Supporto Oracle 2471245.1).

   La configurazione include la modifica del file `xtt.properties` sull'host di origine. Il file `xtt.properties` di esempio seguente specifica i backup delle tre tablespace nella directory `/dsk1/backups`. Si tratta delle tablespace da trasportare nell'istanza database di destinazione. Specifica inoltre l'ID della piattaforma di origine per convertire automaticamente l'endiannes.

   > **Note**
   >
   > Per gli ID di piattaforma, consulta Data Guard Support for Heterogeneous Primary and Physical Standbys in Same Data Guard Configuration (Doc ID 413484.1).

```
#linux system
platformid=13
#list of tablespaces to transport
tablespaces=TBS1,TBS2,TBS3
#location where backup will be generated
src_scratch_location=/dsk1/backups
#RMAN command for performing backup
```

```
usermantransport=1
```

Fase 2: preparazione del backup completo delle tablespace

In questa fase, si esegue il backup delle tablespace per la prima volta, si trasferiscono i backup sull'host di destinazione e quindi li si ripristina utilizzando la procedura `rdsadmin.rdsadmin_transport_util.import_xtts_tablespaces`. Dopo aver completato questa fase, i backup iniziali delle tablespace si trovano nell'istanza database di destinazione e possono essere aggiornati mediante backup incrementali.

Argomenti

- Passaggio 1: esecuzione del backup delle tablespace sull'host di origine
- Passaggio 2: trasferimento dei file di backup nell'istanza database di destinazione
- Passaggio 3: importazione delle tablespace nell'istanza database di destinazione

Passaggio 1: esecuzione del backup delle tablespace sull'host di origine

In questo passaggio, si utilizza lo script `xttdriver.pl` per eseguire un backup completo delle tablespace. L'output di `xttdriver.pl` è archiviato nella variabile di ambiente `TMPDIR`.

Esecuzione del backup delle tablespace

1. Se le tablespace sono in modalità di sola lettura, accedere al database di origine come utente con il privilegio `ALTER TABLESPACE` e impostare le tablespace sulla modalità di lettura/scrittura. Altrimenti, passare alla fase successiva.

   L'esempio seguente imposta `tbs1`, `tbs2` e `tbs3` sulla modalità lettura/scrittura.

   ```
   ALTER TABLESPACE tbs1 READ WRITE;
   ALTER TABLESPACE tbs2 READ WRITE;
   ALTER TABLESPACE tbs3 READ WRITE;
   ```

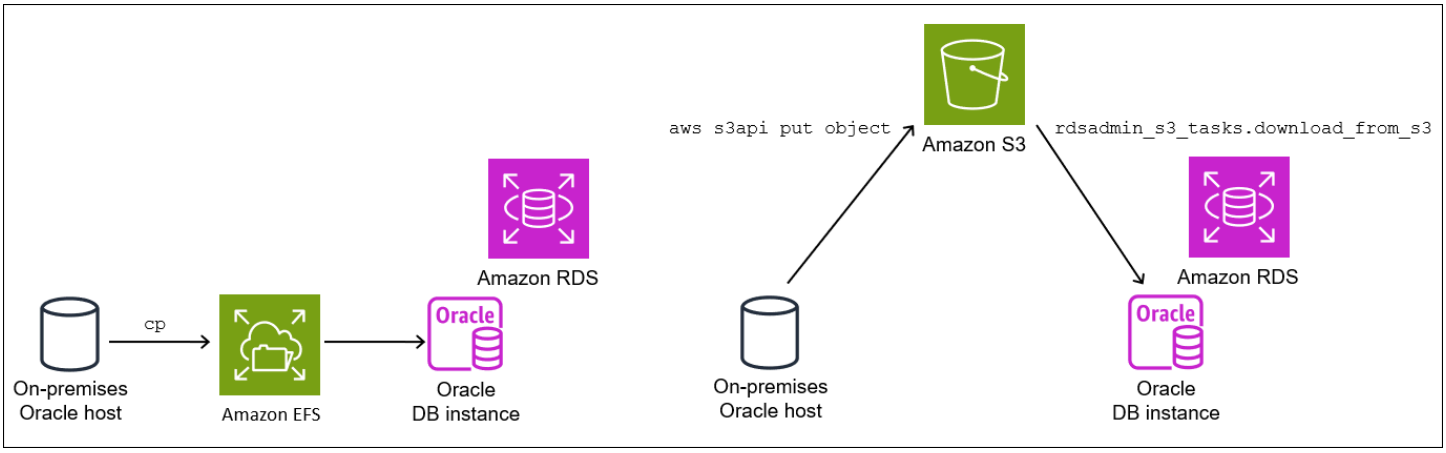
2. Eseguire il backup delle tablespace utilizzando lo script `xttdriver.pl`. Facoltativamente, è possibile specificare `--debug` per eseguire lo script in modalità di debug.

   ```
   export TMPDIR=location_of_log_files
   cd location_of_xttdriver.pl
   $ORACLE_HOME/perl/bin/perl xttdriver.pl --backup
   ```

Passaggio 2: trasferimento dei file di backup nell'istanza database di destinazione

In questo passaggio, vengono copiati i file di backup e la configurazione dalla posizione temporanea all'istanza database di destinazione. Selezionare una delle seguenti opzioni:

- Se gli host di origine e di destinazione condividono un file system Amazon EFS, utilizzare un'utilità del sistema operativo, ad esempio `cp`, per copiare i file di backup e il file `res.txt` dalla posizione temporanea in una directory condivisa. Quindi passa a Passaggio 3: importazione delle tablespace nell'istanza database di destinazione.
- Se è necessario preparare i backup in un bucket Amazon S3, completare i seguenti passaggi.



Passaggio 2.2: caricamento dei backup nel bucket Amazon S3

Caricare i backup e il file `res.txt` dalla cartella temporanea al bucket Amazon S3. Per ulteriori informazioni, consulta Caricamento degli oggetti nella Guida per l'utente di Amazon Simple Storage Service.

Passaggio 2.3: scaricamento dei backup dal bucket Amazon S3 all'istanza database di destinazione

In questo passaggio, si utilizza la procedura `rdsadmin.rdsadmin_s3_tasks.download_from_s3` per scaricare i backup nell'istanza database RDS per Oracle.

Scaricamento dei backup dal bucket Amazon S3

1. Avviare SQL*Plus o Oracle SQL Developer e accedere all'istanza database RDS per Oracle.
2. Scaricare i backup dal bucket Amazon S3 nell'istanza database di destinazione utilizzando la procedura Amazon RDS `rdsadmin.rdsadmin_s3_tasks.download_from_s3` su d.

L'esempio seguente illustra come scaricare tutti i file da un bucket Amazon S3 denominato *mys3bucket* nella directory *DATA_PUMP_DIR*.

```
EXEC UTL_FILE.FREMOVE ('DATA_PUMP_DIR', 'res.txt');
SELECT rdsadmin.rdsadmin_s3_tasks.download_from_s3(
  p_bucket_name    =>  'mys3bucket',
  p_directory_name =>  'DATA_PUMP_DIR')
AS TASK_ID FROM DUAL;
```

L'istruzione SELECT restituisce l'ID dell'attività in un tipo di dati VARCHAR2. Per ulteriori informazioni, consulta Download di file da un bucket Amazon S3 a un'istanza database Oracle.

Passaggio 3: importazione delle tablespace nell'istanza database di destinazione

Per ripristinare i tablespace nell'istanza DB di destinazione, utilizza la procedura. `rdsadmin.rdsadmin_transport_util.import_xtts_tablespaces` Questa procedura converte automaticamente i file di dati nel formato endian corretto.

Se importate da una piattaforma diversa da Linux, specificate la piattaforma di origine utilizzando il parametro `p_platform_id` quando chiamate. `import_xtts_tablespaces` Assicuratevi che l'ID della piattaforma specificato corrisponda a quello specificato nel `xtt.properties` file inPassaggio 2: esportazione dei metadati delle tablespace nell'host di origine.

Importazione delle tablespace nell'istanza database di destinazione

1. Avviare il client Oracle SQL e accedere come utente master all'istanza database RDS per Oracle di destinazione.
2. Eseguire la procedura `rdsadmin.rdsadmin_transport_util.import_xtts_tablespaces`, specificando le tablespace da importare e la directory contenente i backup.

   L'esempio seguente importa le tablespace *TBS1*, *TBS2* e *TBS3* dalla directory *DATA_PUMP_DIR*. La piattaforma di origine è AIX based Systems (64 bit), che ha l'ID di piattaforma di. 6 È possibile trovare gli ID della piattaforma eseguendo una query. `V$TRANSPORTABLE_PLATFORM`

   ```
   VAR task_id CLOB

   BEGIN
     :task_id:=rdsadmin.rdsadmin_transport_util.import_xtts_tablespaces(
   ```

```
        'TBS1,TBS2,TBS3',
        'DATA_PUMP_DIR',
        p_platform_id => 6);
END;
/

PRINT task_id
```

3. (Facoltativo) Monitorare l'avanzamento eseguendo una query sulla tabella `rdsadmin.rds_xtts_operation_info`. La colonna `xtts_operation_state` mostra il valore `EXECUTING`, `COMPLETED` o`FAILED`.

```
SELECT * FROM rdsadmin.rds_xtts_operation_info;
```

> **Note**
>
> Per operazioni con tempi di esecuzione lunghi, è anche possibile eseguire una query su `V$SESSION_LONGOPS` `V$RMAN_STATUS` e `V$RMAN_OUTPUT`.

4. Visualizzare il log dell'importazione completata utilizzando l'ID attività del passaggio precedente.

```
SELECT * FROM TABLE(rdsadmin.rds_file_util.read_text_file('BDUMP',
 'dbtask-'||'&task_id'||'.log'));
```

Assicurarsi che l'importazione sia stata completata correttamente prima di passare alla fase successiva.

Fase 3: creazione e trasferimento dei backup incrementali

In questa fase, si effettuano e si trasferiscono periodicamente backup incrementali mentre il database di origine è attivo. Questa tecnica riduce le dimensioni del backup finale delle tablespace. Se vengono eseguiti più backup incrementali, è necessario copiare il file `res.txt` dopo l'ultimo backup incrementale prima di poterlo applicare all'istanza di destinazione.

I passaggi sono gli stessi di quelli indicati in Fase 2: preparazione del backup completo delle tablespace, tranne per il fatto che il passaggio di importazione è facoltativo.

Fase 4: trasporto delle tablespace

In questa fase, si esegue il backup delle tablespace di sola lettura e si esportano i metadati di Data Pump, questi file vengono quindi trasferiti nell'host di destinazione e infine vengono importati sia le tablespace che i metadati.

Argomenti

Passaggio 1: esecuzione del backup delle tablespace di sola lettura

Questo passaggio è identico a Passaggio 1: esecuzione del backup delle tablespace sull'host di origine, con una differenza fondamentale: le tablespace vengono impostate sulla modalità di sola lettura prima di eseguirne il backup per l'ultima volta.

L'esempio seguente imposta `tbs1`, `tbs2` e `tbs3` sulla modalità di sola lettura.

```
ALTER TABLESPACE tbs1 READ ONLY;
ALTER TABLESPACE tbs2 READ ONLY;
ALTER TABLESPACE tbs3 READ ONLY;
```

Passaggio 2: esportazione dei metadati delle tablespace nell'host di origine

Esportare i metadati delle tablespace eseguendo l'utilità `expdb` nell'host di origine. L'esempio seguente esporta le tablespace *`TBS1`*, *`TBS2`* e *`TBS3`* nel file di dump *`xttdump.dmp`* nella directory *`DATA_PUMP_DIR`*.

```
expdp username/pwd \
dumpfile=xttdump.dmp \
directory=DATA_PUMP_DIR \
statistics=NONE \
transport_tablespaces=TBS1,TBS2,TBS3 \
transport_full_check=y \
logfile=tts_export.log
```

Se *DATA_PUMP_DIR* è una directory condivisa in Amazon EFS, passare a Passaggio 4: importazione delle tablespace nell'istanza database di destinazione.

Passaggio 3: (solo Amazon S3) trasferimento dei file di backup ed esportazione nell'istanza database di destinazione

Se si utilizza Amazon S3 per preparare i backup delle tablespace e il file di esportazione di Data Pump, completare i seguenti passaggi.

Passaggio 3.1: caricamento dei backup e del file di dump dall'host di origine al bucket Amazon S3

Caricare i file di backup e i file di dump dall'host di origine al bucket Amazon S3. Per ulteriori informazioni, consulta Caricamento degli oggetti nella Guida per l'utente di Amazon Simple Storage Service.

Passaggio 3.2: scaricamento dei backup e del file di dump dal bucket Amazon S3 all'istanza database di destinazione

In questo passaggio, si utilizza la procedura `rdsadmin.rdsadmin_s3_tasks.download_from_s3` per scaricare i backup e il file di dump nell'istanza database RDS per Oracle. Seguire la procedura riportata in Passaggio 2.3: scaricamento dei backup dal bucket Amazon S3 all'istanza database di destinazione.

Passaggio 4: importazione delle tablespace nell'istanza database di destinazione

Utilizzare la procedura `rdsadmin.rdsadmin_transport_util.import_xtts_tablespaces` per ripristinare le tablespace. Per la sintassi e la semantica di questa procedura, consulta Importazione di tablespace trasportate nell'istanza database

> **Important**
>
> Dopo aver completato l'importazione finale delle tablespace, il passaggio successivo prevede l'importazione dei metadati di Oracle Data Pump. Se l'importazione non riesce, è importante ripristinare lo stato dell'istanza database precedente all'errore. Pertanto, è consigliabile creare uno snapshot DB dell'istanza database seguendo le istruzioni riportate in Creazione di uno snapshot DB per un'istanza DB Single-AZ. Lo snapshot conterrà tutte le tablespace importate. Pertanto, se l'importazione non riesce, non sarà necessario ripetere il processo di backup e importazione.
>
> Se per l'istanza database di destinazione sono stati abilitati i backup automatici e Amazon RDS non rileva che è stato eseguito uno snapshot valido prima dell'importazione dei metadati, RDS tenta di creare uno snapshot. A seconda dell'attività dell'istanza, questo

snapshot potrebbe riuscire o meno. Se non viene rilevato uno snapshot valido o non è possibile avviarne uno, l'importazione dei metadati viene terminata con errori.

Importazione delle tablespace nell'istanza database di destinazione

1. Avviare il client Oracle SQL e accedere come utente master all'istanza database RDS per Oracle di destinazione.
2. Eseguire la procedura `rdsadmin.rdsadmin_transport_util.import_xtts_tablespaces`, specificando le tablespace da importare e la directory contenente i backup.

   L'esempio seguente importa le tablespace *`TBS1`*, *`TBS2`* e *`TBS3`* dalla directory *`DATA_PUMP_DIR`*.

   ```
   BEGIN

    :task_id:=rdsadmin.rdsadmin_transport_util.import_xtts_tablespaces('TBS1,TBS2,TBS3','DATA_
   END;
   /
   PRINT task_id
   ```

3. (Facoltativo) Monitorare l'avanzamento eseguendo una query sulla tabella `rdsadmin.rds_xtts_operation_info`. La colonna `xtts_operation_state` mostra il valore `EXECUTING`, `COMPLETED` o`FAILED`.

   ```
   SELECT * FROM rdsadmin.rds_xtts_operation_info;
   ```

   > **Note**
   >
   > Per operazioni con tempi di esecuzione lunghi, è anche possibile eseguire una query su `V$SESSION_LONGOPS` `V$RMAN_STATUS` e `V$RMAN_OUTPUT`.

4. Visualizzare il log dell'importazione completata utilizzando l'ID attività del passaggio precedente.

   ```
   SELECT * FROM TABLE(rdsadmin.rds_file_util.read_text_file('BDUMP',
    'dbtask-'||'&task_id'||'.log'));
   ```

   Assicurarsi che l'importazione sia stata completata correttamente prima di passare alla fase successiva.

5. Eseguire uno snapshot DB manuale seguendo le istruzioni riportate in Creazione di uno snapshot DB per un'istanza DB Single-AZ.

Passaggio 5: importazione dei metadati delle tablespace nell'istanza database di destinazione

In questo passaggio, vengono importati i metadati delle tablespace trasportabili nell'istanza database RDS per Oracle utilizzando la procedura `rdsadmin.rdsadmin_transport_util.import_xtts_metadata`. Per la sintassi e la semantica di questa procedura, consulta Importazione dei metadati delle tablespace trasportabili nell'istanza database. Durante l'operazione, lo stato dell'importazione viene visualizzato nella tabella `rdsadmin.rds_xtts_operation_info`.

> **Important**
>
> Prima di importare i metadati, è vivamente consigliabile di verificare che sia stato creato correttamente uno snapshot DB dopo aver importato le tablespace. Se la fase di importazione ha esito negativo, ripristinare l'istanza database, correggere gli errori di importazione e riprovare l'importazione.

Importazione dei metadati di Data Pump nell'istanza database RDS per Oracle

1. Avviare il client Oracle SQL e accedere come utente master all'istanza database RDS per Oracle di destinazione.
2. Se non esistono già, creare gli utenti proprietari degli schemi nelle tablespace trasportate.

```
CREATE USER tbs_owner IDENTIFIED BY password;
```

3. Importare i metadati, specificando il nome del file di dump e la sua posizione nella directory.

```
BEGIN

 rdsadmin.rdsadmin_transport_util.import_xtts_metadata('xttdump.dmp','DATA_PUMP_DIR');
END;
/
```

4. (Facoltativo) Eseguire una query sulla tabella della cronologia delle tablespace trasportabili per visualizzare lo stato dell'importazione dei metadati.

```
SELECT * FROM rdsadmin.rds_xtts_operation_info;
```

Al termine dell'operazione, le tablespace sono in modalità di sola lettura.

5. (Facoltativo) Visualizzare il file di log.

   L'esempio seguente elenca il contenuto della directory BDUMP e quindi esegue una query sul log di importazione.

```
SELECT * FROM TABLE(rdsadmin.rds_file_util.listdir(p_directory => 'BDUMP'));

SELECT * FROM TABLE(rdsadmin.rds_file_util.read_text_file(
  p_directory => 'BDUMP',
  p_filename => 'rds-xtts-
import_xtts_metadata-2023-05-22.01-52-35.560858000.log'));
```

Fase 5: convalida delle tablespace trasportate

In questo passaggio facoltativo, le tablespace trasportate vengono convalidate utilizzando la procedura`rdsadmin.rdsadmin_rman_util.validate_tablespace`, quindi vengono impostate sulla modalità di lettura/scrittura.

Convalida dei dati trasportati

1. Avviare SQL*Plus o SQL Developer e accedere come utente master all'istanza database RDS per Oracle.
2. Convalidare le tablespace utilizzando la procedura `rdsadmin.rdsadmin_rman_util.validate_tablespace`.

```
SET SERVEROUTPUT ON
BEGIN
    rdsadmin.rdsadmin_rman_util.validate_tablespace(
        p_tablespace_name     => 'TBS1',
        p_validation_type     => 'PHYSICAL+LOGICAL',
        p_rman_to_dbms_output => TRUE);
    rdsadmin.rdsadmin_rman_util.validate_tablespace(
        p_tablespace_name     => 'TBS2',
        p_validation_type     => 'PHYSICAL+LOGICAL',
        p_rman_to_dbms_output => TRUE);
```

```
    rdsadmin.rdsadmin_rman_util.validate_tablespace(
        p_tablespace_name     => 'TBS3',
        p_validation_type     => 'PHYSICAL+LOGICAL',
        p_rman_to_dbms_output => TRUE);
END;
/
```

3. Impostare le tablespace sulla modalità lettura/scrittura.

```
ALTER TABLESPACE TBS1 READ WRITE;
ALTER TABLESPACE TBS2 READ WRITE;
ALTER TABLESPACE TBS3 READ WRITE;
```

Fase 6: rimozione dei file residui

In questo passaggio facoltativo, tutti i file non necessari vengono rimossi. Usa la procedura `rdsadmin.rdsadmin_transport_util.list_xtts_orphan_files` per elencare i file di dati che sono rimasti orfani dopo un'importazione tablespace, quindi utilizza la procedura `rdsadmin.rdsadmin_transport_util.list_xtts_orphan_files` per eliminarli. Per la sintassi e la semantica di queste procedure, consulta Elenco dei file orfani dopo un'importazione della tablespace e Eliminazione di file di dati rimasti orfani dopo un'importazione della tablespace.

Rimozione dei file residui

1. Rimuovere i vecchi backup in *DATA_PUMP_DIR* nel seguente modo:

   a. Elencare i file di backup eseguendo `rdsadmin.rdsadmin_file_util.listdir`.

   ```
   SELECT * FROM TABLE(rdsadmin.rds_file_util.listdir(p_directory =>
    'DATA_PUMP_DIR'));
   ```

   b. Rimuovere i backup uno a uno chiamando `UTL_FILE.FREMOVE`.

   ```
   EXEC UTL_FILE.FREMOVE ('DATA_PUMP_DIR', 'backup_filename');
   ```

2. Se è stata eseguita l'importazione delle tablespace, ma non dei relativi metadati, è possibile eliminare i file di dati orfani nel seguente modo:

a. Elencare i file di dati orfani da eliminare. L'esempio seguente esegue la procedura `rdsadmin.rdsadmin_transport_util.list_xtts_orphan_files`.

```
SQL> SELECT * FROM
 TABLE(rdsadmin.rdsadmin_transport_util.list_xtts_orphan_files);

FILENAME       FILESIZE
-------------- ---------
datafile_7.dbf 104865792
datafile_8.dbf 104865792
```

b. Eliminare i file orfani eseguendo la procedura `rdsadmin.rdsadmin_transport_util.cleanup_incomplete_xtts_import`.

```
BEGIN

 rdsadmin.rdsadmin_transport_util.cleanup_incomplete_xtts_import('DATA_PUMP_DIR');
END;
/
```

L'operazione di rimozione genera un file di log che utilizza il formato di nome `rds-xtts-delete_xtts_orphaned_files-`*`YYYY-MM-DD.HH24-MI-SS.FF`*`.log` nella directory BDUMP.

c. Leggere il file di log generato nel passaggio precedente. Il seguente esempio legge il log `rds-xtts-delete_xtts_orphaned_files-`*`2023-06-01.09-33-11.868894000`*`.log`.

```
SELECT *
FROM TABLE(rdsadmin.rds_file_util.read_text_file(
       p_directory => 'BDUMP',
       p_filename  => 'rds-xtts-
delete_xtts_orphaned_files-2023-06-01.09-33-11.868894000.log'));

TEXT
--------------------------------------------------------------------------------
orphan transported datafile datafile_7.dbf deleted.
orphan transported datafile datafile_8.dbf deleted.
```

3. Se è stata eseguita l'importazione sia delle tablespace che dei relativi metadati, ma sono stati restituiti errori di compatibilità o si sono verificati problemi di altro tipo con Oracle Data Pump, rimuovere i file di dati parzialmente trasportati nel seguente modo:

   a. Elencare le tablespace contenente i file di dati parzialmente trasportati mediante una query su `DBA_TABLESPACES`.

   ```
   SQL> SELECT TABLESPACE_NAME FROM DBA_TABLESPACES WHERE PLUGGED_IN='YES';

   TABLESPACE_NAME
   ------------------------------------------------------------------------------
   TBS_3
   ```

   b. Rimuovere le tablespace e i file di dati parzialmente trasportati.

   ```
   DROP TABLESPACE TBS_3 INCLUDING CONTENTS AND DATAFILES;
   ```

Importazione utilizzando Oracle Data Pump

Oracle Data Pump è un'utilità che consente di esportare i dati Oracle in un file di dump e importarli in un altro database Oracle. È un sostituto a lungo termine per le utilità di esportazione/importazione di Oracle e rappresenta lo strumento consigliato per spostare grandi quantità di dati da un'installazione di Oracle a un'istanza database Amazon RDS.

Gli esempi riportati in questa sezione mostrano un modo per importare i dati in un database Oracle, ma Oracle Data Pump supporta anche altre procedure. Per ulteriori informazioni, consulta la documentazione relativa a Oracle Database.

Gli esempi in questa sezione utilizzano il pacchetto `DBMS_DATAPUMP`. Puoi eseguire le stesse attività utilizzando le utilità della riga di comando di Oracle Data Pump `impdp` e `expdp`. È possibile installare queste utilità in un host remoto come parte di un'installazione Oracle Client, incluso Oracle Instant Client. Per ulteriori informazioni, consulta la pagina relativa alla procedura di Oracle Instant Client da usare per eseguire l'importazione o l'esportazione di Data Pump per l'istanza database Amazon RDS per Oracle.

Argomenti

- Panoramica su Oracle Data Pump
- Importazione di dati con Oracle Data Pump e un bucket Amazon S3

- Importazione di dati con Oracle Data Pump e un collegamento di database

Panoramica su Oracle Data Pump

Oracle Data Pump è composto dai seguenti componenti:

- Client della riga di comando `expdp` e `impdp`
- Pacchetto PL/SQL `DBMS_DATAPUMP`
- Pacchetto PL/SQL `DBMS_METADATA`

Puoi utilizzare Oracle Data Pump per gli scenari seguenti:

- Importazione dei dati da un database Oracle (on-premise o in un'istanza Amazon EC2) in un'istanza database Amazon RDS per Oracle.
- Importazione dei dati da un'istanza database RDS per Oracle in un database Oracle (locale o su un'istanza Amazon EC2).
- Importazione dei dati tra istanze database RDS per Oracle (ad esempio, per la migrazione dei dati da EC2-Classic a VPC).

Per scaricare le utilità di Oracle Data Pump, consultare Download di software per database Oracle sul sito web di Oracle Technology Network. Per le considerazioni sulla compatibilità durante la migrazione tra versioni di Oracle Database, consulta la documentazione di Oracle Database.

Flusso di lavoro di Oracle Data Pump

In genere, si utilizza Oracle Data Pump nelle seguenti fasi:

1. Esportazione dei dati in un file di dump nel database di origine.
2. Caricamento del file di dump nell'istanza database RDS per Oracle di destinazione. Puoi eseguire il trasferimento utilizzando un bucket Amazon S3 o un collegamento di database tra i due database.
3. Importazione i dati dal file di dump nell'istanza database RDS per Oracle.

Best practice di Oracle Data Pump

Quando si utilizza Oracle Data Pump per importare dati in un'istanza RDS per Oracle, si consiglia di attenersi alle seguenti best practice:

- Esegui le importazioni in modalità `schema` o `table` per importare schemi e oggetti specifici.
- Limita gli schemi che importi a quelli richiesti dalla tua applicazione.
- Non eseguire l'importazione in modalità `full`, né importare schemi per i componenti gestiti dal sistema.

  Perché RDS per Oracle non consente l'accesso a utenti con privilegi di amministratore `SYS` o `SYSDBA`, queste azioni potrebbero danneggiare il dizionario dei dati Oracle e pregiudicare la stabilità del database.
- Quando si caricano grandi quantità di dati, procedere nel seguente modo:

  1. Trasferire il file di dump nell'istanza database RDS per Oracle di destinazione.
  2. Acquisire una snapshot DB dell'istanza.
  3. Verificare che l'importazione ha esito positivo.

  Se i componenti del database sono invalidati, puoi eliminare l'istanza database e ricrearla dallo snapshot del database. L'istanza database ripristinata include i file di dump archiviati sull'istanza database al momento della creazione dello snapshot del database.
- Non importare file di dump creati utilizzando i parametri di esportazione di Oracle Data Pump `TRANSPORT_TABLESPACES`, `TRANSPORTABLE` oppure `TRANSPORT_FULL_CHECK`. Le istanze database RDS per Oracle non supportano l'importazione di questi file di dump.
- Non importare file di dump che contengono oggetti Oracle Scheduler in `SYS`, `SYSTEM`, `RDSADMIN`, `RDSSEC` e `RDS_DATAGUARD`, e che appartengono alle seguenti categorie:
  - Processi
  - Programmi
  - Piani
  - Chain
  - Regolamento
  - Contesti di valutazione
  - Set di regole

  Le istanze database RDS per Oracle non supportano l'importazione di questi file di dump.
- Per escludere gli oggetti di Oracle Scheduler non supportati, utilizza direttive aggiuntive durante l'esportazione Data Pump. Se utilizzi `DBMS_DATAPUMP`, aggiungi un'altra direttiva `METADATA_FILTER` prima di `DBMS_METADATA.START_JOB`:

```
DBMS_DATAPUMP.METADATA_FILTER(
  v_hdnl,
  'EXCLUDE_NAME_EXPR',
  q'[IN (SELECT NAME FROM SYS.OBJ$
         WHERE TYPE# IN (66,67,74,79,59,62,46)
         AND OWNER# IN
           (SELECT USER# FROM SYS.USER$
            WHERE NAME IN ('RDSADMIN','SYS','SYSTEM','RDS_DATAGUARD','RDSSEC')
            )
        )
  ]',
  'PROCOBJ'
);
```

Se utilizzi `expdp`, crea un file di parametri contenente la direttiva `exclude` illustrata nell'esempio seguente. Quindi usa `PARFILE=`*`parameter_file`* con il comando `expdp`.

```
exclude=procobj:"IN
  (SELECT NAME FROM sys.OBJ$
   WHERE TYPE# IN (66,67,74,79,59,62,46)
   AND OWNER# IN
     (SELECT USER# FROM SYS.USER$
      WHERE NAME IN ('RDSADMIN','SYS','SYSTEM','RDS_DATAGUARD','RDSSEC')
     )
  )"
```

Importazione di dati con Oracle Data Pump e un bucket Amazon S3

Il seguente processo di importazione utilizza Oracle Data Pump e un bucket Amazon S3. I passaggi sono i seguenti:

1. Esporta i dati nel database di origine utilizzando il pacchetto Oracle DBMS_DATAPUMP.
2. Inserisci il file di dump in un bucket Amazon S3.
3. Scarica il file di dump dal bucket Amazon S3 nella directory `DATA_PUMP_DIR` sull'istanza database RDS per Oracle di destinazione.
4. Importa i dati del file di dump copiato nell'istanza database RDS per Oracle utilizzando il pacchetto `DBMS_DATAPUMP`.

Argomenti

Requisiti di importazione dei dati con Oracle Data Pump e un bucket Amazon S3

Il processo ha i requisiti seguenti:

- Assicurati che sia disponibile un bucket Amazon S3 per i trasferimenti di file e che il bucket Amazon S3 sia nella stessa istanza DB. Regione AWS Per istruzioni, consultare Creazione di un bucket nella Guida introduttiva di Amazon Simple Storage Service.
- L'oggetto caricato nel bucket Amazon S3 deve essere pari o inferiore a 5 TB. Per ulteriori informazioni sull'utilizzo di oggetti nel Amazon S3, consulta Guida per l'utente di Amazon Simple Storage Service.

> **Note**
>
> Se il file di dump supera i 5 TB, è possibile eseguire l'esportazione Oracle Data Pump con l'opzione parallela. Questa operazione diffonde i dati in più file di dump in modo da non superare il limite di 5 TB per i singoli file.

- È necessario preparare il bucket Amazon S3 per l'integrazione con Amazon RDS seguendo le istruzioni in Configurazione delle autorizzazioni IAM per l'integrazione di RDS per Oracle con Amazon S3.
- È necessario disporre di spazio di storage sufficiente per archiviare il file dump nell'istanza di origine e nell'istanza database di destinazione.

> **Note**
>
> Questo processo importa un file dump nella directory `DATA_PUMP_DIR`, una directory preconfigurata in tutte le istanze database Oracle. La directory si trova nello stesso volume di storage dei file di dati. Quando importi il file di dump, i file di dati di Oracle esistenti occupano più spazio. Pertanto, devi assicurarti che l'istanza database possa accomodare tale utilizzo dello spazio aggiuntivo. Il file dump importato non viene eliminato o ripulito automaticamente dalla directory `DATA_PUMP_DIR`. Per rimuovere il file di dump importato, utilizzare UTL_FILE.FREMOVE, disponibile sul sito web di Oracle.

Fase 1: concessione dei privilegi all'utente del database sull'istanza database RDS per Oracle

In questa fase, si creano gli schemi in cui si intende importare i dati e si concedono agli utenti i privilegi necessari.

Per creare utenti e concedere i privilegi necessari sull'istanza RDS per Oracle di destinazione

1. Utilizza SQL*Plus o Oracle SQL Developer per la connessione come utente master all'istanza database RDS per Oracle in cui verranno importati i dati. Per informazioni sulla connessione a un'istanza database, consulta Connessione all'istanza database RDS per Oracle.
2. Prima di importare i dati, crea gli spazi di tabella necessari. Per ulteriori informazioni, consulta Creazione e dimensionamento di spazi tabelle.
3. Se l'account utente in cui vengono importati i dati non esiste, crealo e concedigli le autorizzazioni e i ruoli necessari. Se intendi importare i dati su più schemi utente, crea tutti gli account utente e concedi loro i privilegi e i ruoli necessari.

   Ad esempio, le istruzioni SQL seguenti permettono di creare un nuovo utente e concedergli le autorizzazioni e i ruoli necessari per importare i dati nel suo schema. In questo passaggio e in quelli successivi, sostituisci *schema_1* con il nome del tuo schema.

   ```
   CREATE USER schema_1 IDENTIFIED BY my_password;
   GRANT CREATE SESSION, RESOURCE TO schema_1;
   ALTER USER schema_1 QUOTA 100M ON users;
   ```

> **Note**
>
> Specifica una password diversa dal prompt mostrato qui come best practice per la sicurezza.

Le istruzioni precedenti concedono al nuovo utente il privilegio `CREATE SESSION` e il ruolo `RESOURCE`. Potrebbero essere necessari privilegi e ruoli aggiuntivi, a seconda degli oggetti del database da importare.

Fase 2: esportazione dei dati in un file di dump utilizzando DBMS_DATAPUMP

Per creare un file di dump, utilizza il pacchetto `DBMS_DATAPUMP`.

Per esportare i dati Oracle in un file di dump

1. Utilizza SQL Plus o Oracle SQL Developer per connetterti all'istanza database RDS per Oracle di origine come utente amministratore. Se il database di origine è un'istanza database RDS per Oracle, esegui la connessione con l'utente master Amazon RDS.
2. Esporta i dati richiamando le procedure `DBMS_DATAPUMP`.

   Il seguente script esporta lo schema *`SCHEMA_1`* in un file di dump denominato `sample.dmp`nella directory `DATA_PUMP_DIR`. Sostituisci *`SCHEMA_1`* con il nome dello schema che si desidera esportare.

```
DECLARE
  v_hdnl NUMBER;
BEGIN
  v_hdnl := DBMS_DATAPUMP.OPEN(
    operation => 'EXPORT',
    job_mode  => 'SCHEMA',
    job_name  => null
  );
  DBMS_DATAPUMP.ADD_FILE(
    handle    => v_hdnl         ,
    filename  => 'sample.dmp'   ,
    directory => 'DATA_PUMP_DIR',
    filetype  => dbms_datapump.ku$_file_type_dump_file
  );
```

```
  DBMS_DATAPUMP.ADD_FILE(
    handle    => v_hdnl,
    filename  => 'sample_exp.log',
    directory => 'DATA_PUMP_DIR' ,
    filetype  => dbms_datapump.ku$_file_type_log_file
  );
  DBMS_DATAPUMP.METADATA_FILTER(v_hdnl,'SCHEMA_EXPR','IN (''SCHEMA_1'')');
  DBMS_DATAPUMP.METADATA_FILTER(
    v_hdnl,
    'EXCLUDE_NAME_EXPR',
    q'[IN (SELECT NAME FROM SYS.OBJ$
           WHERE TYPE# IN (66,67,74,79,59,62,46)
           AND OWNER# IN
             (SELECT USER# FROM SYS.USER$
              WHERE NAME IN ('RDSADMIN','SYS','SYSTEM','RDS_DATAGUARD','RDSSEC')
             )
          )
    ]',
    'PROCOBJ'
  );
  DBMS_DATAPUMP.START_JOB(v_hdnl);
END;
/
```

> **Note**
>
> Data Pump avvia i processi in modo asincrono. Per informazioni sul monitoraggio di un processo di Data Pump, consulta la pagina relativa al monitoraggio dello stato dei processi nella documentazione di Oracle.

3. (Facoltativo) Visualizza il contenuto del log di esportazione utilizzando la procedura `rdsadmin.rds_file_util.read_text_file`. Per ulteriori informazioni, consulta Lettura dei file in una directory di istanze database.

Passaggio 3: Caricamento del file di dump sul bucket Amazon S3

Utilizzare la procedura `rdsadmin.rdsadmin_s3_tasks.upload_to_s3` di Amazon RDS per copiare il file dump sul bucket Amazon S3. L'esempio seguente carica tutti i file dalla directory `DATA_PUMP_DIR` su un bucket Amazon S3 denominato *`myS3bucket`*.

```
SELECT rdsadmin.rdsadmin_s3_tasks.upload_to_s3(
```

```
  p_bucket_name    =>  'myS3bucket',
  p_directory_name =>  'DATA_PUMP_DIR')
AS TASK_ID FROM DUAL;
```

L'istruzione `SELECT` restituisce l'ID dell'attività in un tipo di dati `VARCHAR2`. Per ulteriori informazioni, consulta Caricamento di file da un'istanza database Oracle a un bucket Amazon S3.

Fase 4: scaricamento del file di dump dal bucket Amazon S3 all'istanza database di destinazione

Esegui questo passaggio utilizzando la procedura Amazon RDS `rdsadmin.rdsadmin_s3_tasks.download_from_s3`. Quando si scarica un file in una directory, la procedura `download_from_s3` salta il processo di scaricamento se nella directory esiste già un file con lo stesso nome. Per rimuovere il file di dump importato, utilizza UTL_FILE.FREMOVE, disponibile sul sito Web di Oracle.

Per scaricare il file di dump

1. Avvia SQL*Plus o Oracle SQL Developer e accedi come utente principale nell'istanza database Amazon RDS per Oracle di destinazione.
2. Scarica il file di dump utilizzando la procedura Amazon RDS `rdsadmin.rdsadmin_s3_tasks.download_from_s3`.

   L'esempio seguente illustra come scaricare tutti i file da un bucket Amazon S3 denominato *`myS3bucket`* nella directory `DATA_PUMP_DIR`.

   ```
   SELECT rdsadmin.rdsadmin_s3_tasks.download_from_s3(
     p_bucket_name    =>  'myS3bucket',
     p_directory_name =>  'DATA_PUMP_DIR')
   AS TASK_ID FROM DUAL;
   ```

   L'istruzione `SELECT` restituisce l'ID dell'attività in un tipo di dati `VARCHAR2`. Per ulteriori informazioni, consulta Download di file da un bucket Amazon S3 a un'istanza database Oracle.

Fase 5: importazione del file di dump nell'istanza DB di destinazione utilizzando DBMS_DATAPUMP

Usa `DBMS_DATAPUMP` per importare lo schema nell'istanza database RDS per Oracle. Potrebbero essere necessarie opzioni aggiuntive, come `METADATA_REMAP`.

Per importare dati nell'istanza database di destinazione

1. Avvia SQL*Plus o SQL Developer e accedi come utente master all'istanza database RDS per Oracle.
2. Importa i dati chiamando le procedure. `DBMS_DATAPUMP`

   L'esempio seguente importa i dati *`SCHEMA_1`* da `sample_copied.dmp` nell'istanza database di destinazione.

```
DECLARE
  v_hdnl NUMBER;
BEGIN
  v_hdnl := DBMS_DATAPUMP.OPEN(
    operation => 'IMPORT',
    job_mode  => 'SCHEMA',
    job_name  => null);
  DBMS_DATAPUMP.ADD_FILE(
    handle    => v_hdnl,
    filename  => 'sample_copied.dmp',
    directory => 'DATA_PUMP_DIR',
    filetype  => dbms_datapump.ku$_file_type_dump_file);
  DBMS_DATAPUMP.ADD_FILE(
    handle    => v_hdnl,
    filename  => 'sample_imp.log',
    directory => 'DATA_PUMP_DIR',
    filetype  => dbms_datapump.ku$_file_type_log_file);
  DBMS_DATAPUMP.METADATA_FILTER(v_hdnl,'SCHEMA_EXPR','IN (''SCHEMA_1'')');
  DBMS_DATAPUMP.START_JOB(v_hdnl);
END;
/
```

> **Note**
>
> I processi di Data Pump vengono avviati in modo asincrono. Per informazioni sul monitoraggio di un processo di Data Pump, consulta la pagina relativa al monitoraggio dello stato dei processi nella documentazione di Oracle. È possibile visualizzare il contenuto del log di importazione utilizzando la procedura `rdsadmin.rds_file_util.read_text_file`. Per ulteriori informazioni, consulta Lettura dei file in una directory di istanze database.

3. Verifica l'importazione dei dati elencando le tabelle dello schema nell'istanza database di destinazione.

   Ad esempio, la query seguente restituisce il numero di tabelle per *SCHEMA_1*.

   ```
   SELECT COUNT(*) FROM DBA_TABLES WHERE OWNER='SCHEMA_1';
   ```

Fase 6: eseguire la pulizia

Dopo che i dati sono stati importati, puoi eliminare i file che non intendi conservare.

Per rimuovere i file non necessari

1. Avvia SQL*Plus o SQL Developer e accedi come utente master all'istanza database RDS per Oracle.
2. Elenca i file contenuti in `DATA_PUMP_DIR` utilizzando il seguente comando.

   ```
   SELECT * FROM TABLE(rdsadmin.rds_file_util.listdir('DATA_PUMP_DIR')) ORDER BY
    MTIME;
   ```

3. Elimina i file non più necessari da `DATA_PUMP_DIR` utilizzando il seguente comando.

   ```
   EXEC UTL_FILE.FREMOVE('DATA_PUMP_DIR','filename');
   ```

   Ad esempio, il comando seguente elimina il file denominato `sample_copied.dmp`.

   ```
   EXEC UTL_FILE.FREMOVE('DATA_PUMP_DIR','sample_copied.dmp');
   ```

Importazione di dati con Oracle Data Pump e un collegamento di database

I processi di importazione seguenti utilizzano Oracle Data Pump e il pacchetto Oracle DBMS_FILE_TRANSFER. I passaggi sono i seguenti:

1. Stabilisci una connessione a un database Oracle di origine che può essere on-premise, un'istanza Amazon EC2 o un'istanza database RDS per Oracle.
2. Esporta i dati utilizzando il pacchetto DBMS_DATAPUMP.

3. Utilizza `DBMS_FILE_TRANSFER.PUT_FILE` per copiare il file di dump dal database Oracle alla directory `DATA_PUMP_DIR` sull'istanza database RDS per Oracle di destinazione connessa tramite un collegamento di database.
4. Importa i dati del file di dump copiato nell'istanza database RDS per Oracle utilizzando il pacchetto `DBMS_DATAPUMP`.

Il processo di importazione che utilizza Oracle Data Pump e il pacchetto `DBMS_FILE_TRANSFER` è costituito dalle seguenti fasi.

Argomenti

Requisiti di importazione dei dati con Oracle Data Pump e un collegamento di database

Il processo ha i requisiti seguenti:

- Sono necessari i privilegi di esecuzione per i pacchetti `DBMS_FILE_TRANSFER` e `DBMS_DATAPUMP`.
- Sono necessari i privilegi di scrittura nella directory `DATA_PUMP_DIR` nell'istanza database di origine.
- È necessario disporre di spazio di storage sufficiente per archiviare il file dump nell'istanza di origine e nell'istanza database di destinazione.

> **Note**
>
> Questo processo importa un file dump nella directory `DATA_PUMP_DIR`, una directory preconfigurata in tutte le istanze database Oracle. La directory si trova nello stesso volume di storage dei file di dati. Quando importi il file di dump, i file di dati di Oracle esistenti occupano più spazio. Pertanto, devi assicurarti che l'istanza database possa accomodare tale utilizzo dello spazio aggiuntivo. Il file dump importato non viene eliminato o ripulito automaticamente dalla directory `DATA_PUMP_DIR`. Per rimuovere il file di dump importato, utilizzare UTL_FILE.FREMOVE, disponibile sul sito web di Oracle.

Fase 1: concessione dei privilegi all'utente sull'istanza database RDS per Oracle di destinazione

Per concedere i privilegi all'utente sull'istanza database RDS per Oracle di destinazione, esegui i seguenti passaggi:

1. Utilizza SQL Plus o Oracle SQL Developer per la connessione all'istanza database RDS per Oracle di destinazione in cui verranno importati i dati. Esegui la connessione come utente master Amazon RDS. Per informazioni sulla connessione all'istanza database, consulta Connessione all'istanza database RDS per Oracle.
2. Prima di importare i dati, crea gli spazi di tabella necessari. Per ulteriori informazioni, consulta Creazione e dimensionamento di spazi tabelle.
3. Se l'account utente in cui vengono importati i dati non esiste, crealo e concedigli le autorizzazioni e i ruoli necessari. Se intendi importare i dati su più schemi utente, crea tutti gli account utente e concedi loro i privilegi e i ruoli necessari.

   Ad esempio, i comandi seguenti permettono di creare un nuovo utente denominato *schema_1* e di concedergli le autorizzazioni e i ruoli necessari per importare i dati nel relativo schema.

   ```
   CREATE USER schema_1 IDENTIFIED BY my-password;
   GRANT CREATE SESSION, RESOURCE TO schema_1;
   ALTER USER schema_1 QUOTA 100M ON users;
   ```

   > **Note**
   >
   > Specifica una password diversa dal prompt mostrato qui come best practice per la sicurezza.

L'esempio precedente concede al nuovo utente il privilegio `CREATE SESSION` e il ruolo `RESOURCE`. Potrebbero essere necessari privilegi e ruoli aggiuntivi, a seconda degli oggetti del database da importare.

> **Note**
>
> In questo passaggio e in quelli successivi, sostituisci *schema_1* con il nome del tuo schema.

Passaggio 2: Concessione dei privilegi all'utente nel database di origine

Utilizza SQL*Plus o Oracle SQL Developer per la connessione all'istanza database RDS per Oracle contenente i dati da importare. Se necessario, crea un account utente e concedi le autorizzazioni necessarie.

> **Note**
>
> Se il database di origine è un'istanza Amazon RDS, puoi ignorare questa fase. Per eseguire l'esportazione, utilizzi l'account utente master Amazon RDS.

I comandi seguenti creano un nuovo utente e gli concedono le autorizzazioni necessarie.

```
CREATE USER export_user IDENTIFIED BY my-password;
GRANT CREATE SESSION, CREATE TABLE, CREATE DATABASE LINK TO export_user;
ALTER USER export_user QUOTA 100M ON users;
GRANT READ, WRITE ON DIRECTORY data_pump_dir TO export_user;
GRANT SELECT_CATALOG_ROLE TO export_user;
GRANT EXECUTE ON DBMS_DATAPUMP TO export_user;
GRANT EXECUTE ON DBMS_FILE_TRANSFER TO export_user;
```

> **Note**
>
> Specifica una password diversa dal prompt mostrato qui come best practice per la sicurezza.

Fase 3: creazione di un file di dump mediante DBMS_DATAPUM

Per creare il file batch, procedi come segue:

1. Utilizza SQL*Plus o Oracle SQL Developer per la connessione all'istanza Oracle di origine come utente amministratore o come l'utente creato nel passaggio 2. Se il database di origine è un'istanza database Amazon RDS for Oracle, esegui la connessione con l'utente master Amazon RDS.
2. Crea un file di dump utilizzando l'utilità Oracle Data Pump.

   Lo script seguente crea un file dump denominato sample.dmp nella directory `DATA_PUMP_DIR`.

```
DECLARE
  v_hdnl NUMBER;
BEGIN
  v_hdnl := DBMS_DATAPUMP.OPEN(
    operation => 'EXPORT' ,
    job_mode  => 'SCHEMA' ,
    job_name  => null
  );
  DBMS_DATAPUMP.ADD_FILE(
    handle    => v_hdnl,
    filename  => 'sample.dmp'    ,
    directory => 'DATA_PUMP_DIR' ,
    filetype  => dbms_datapump.ku$_file_type_dump_file
  );
  DBMS_DATAPUMP.ADD_FILE(
    handle    => v_hdnl           ,
    filename  => 'sample_exp.log' ,
    directory => 'DATA_PUMP_DIR'  ,
    filetype  => dbms_datapump.ku$_file_type_log_file
  );
  DBMS_DATAPUMP.METADATA_FILTER(
    v_hdnl              ,
    'SCHEMA_EXPR'       ,
    'IN (''SCHEMA_1'')'
  );
  DBMS_DATAPUMP.METADATA_FILTER(
    v_hdnl,
    'EXCLUDE_NAME_EXPR',
    q'[IN (SELECT NAME FROM sys.OBJ$
           WHERE TYPE# IN (66,67,74,79,59,62,46)
           AND OWNER# IN
```

```
            (SELECT USER# FROM SYS.USER$
             WHERE NAME IN ('RDSADMIN','SYS','SYSTEM','RDS_DATAGUARD','RDSSEC')
            )
          )
    ]',
    'PROCOBJ'
  );
  DBMS_DATAPUMP.START_JOB(v_hdnl);
END;
/
```

> **Note**
>
> I processi di Data Pump vengono avviati in modo asincrono. Per informazioni sul monitoraggio di un processo di Data Pump, consulta la pagina relativa al monitoraggio dello stato dei processi nella documentazione di Oracle. È possibile visualizzare il contenuto del log di esportazione utilizzando la procedura `rdsadmin.rds_file_util.read_text_file`. Per ulteriori informazioni, consulta Lettura dei file in una directory di istanze database.

Fase 4: Creazione di un collegamento di database all'istanza database di destinazione

Crea un collegamento di database tra le istanze database di origine e destinazione. Tieni presente che, per creare un collegamento di database e trasferire il file dump esportato, l'istanza Oracle locale deve disporre di connettività di rete verso l'istanza database di destinazione.

Esegui questa fase connettendoti con lo stesso account utente utilizzato nella fase precedente.

Se crei un collegamento di database tra due istanze database nello stesso VPC o in VPC in peering, le due istanze database devono avere un instradamento valido tra loro. Il gruppo di sicurezza di ogni istanza database deve permettere l'ingresso e l'uscita dall'altra istanza database. Le regole per il traffico in entrata e in uscita del gruppo di sicurezza possono fare riferimento a gruppi di sicurezza dello stesso VPC o di un VPC in peering. Per ulteriori informazioni, consulta Modifica dei collegamenti di database per l'utilizzo con le istanze database in un VPC.

Il comando seguente crea un collegamento di database denominato `to_rds` che si connette all'utente master Amazon RDS sull'istanza database di destinazione:

```
CREATE DATABASE LINK to_rds
```

```
  CONNECT TO <master_user_account> IDENTIFIED BY <password>
  USING '(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=<dns or ip address of remote db>)
         (PORT=<listener port>))(CONNECT_DATA=(SID=<remote SID>)))';
```

Fase 5: copia del file di dump esportato nell'istanza database di destinazione mediante DBMS_FILE_TRANSFER

Utilizza DBMS_FILE_TRANSFER per copiare il file dump dall'istanza database di origine all'istanza database di destinazione. Il seguente script copia un file di dump denominato sample.dmp dall'istanza di origine su un collegamento di database di destinazione denominato to_rds (creato nel passaggio precedente).

```
BEGIN
  DBMS_FILE_TRANSFER.PUT_FILE(
    source_directory_object       => 'DATA_PUMP_DIR',
    source_file_name              => 'sample.dmp',
    destination_directory_object  => 'DATA_PUMP_DIR',
    destination_file_name         => 'sample_copied.dmp',
    destination_database          => 'to_rds' );
END;
/
```

Fase 6: importazione del file di dati nell'istanza database di destinazione mediante DBMS_DATAPUMP

Utilizza Oracle Data Pump per importare lo schema nell'istanza database. Potrebbero essere necessarie opzioni aggiuntive, come METADATA_REMAP.

Stabilisci la connessione all'istanza database con l'account utente master Amazon RDS per eseguire l'importazione.

```
DECLARE
  v_hdnl NUMBER;
BEGIN
  v_hdnl := DBMS_DATAPUMP.OPEN(
    operation => 'IMPORT',
    job_mode  => 'SCHEMA',
    job_name  => null);
  DBMS_DATAPUMP.ADD_FILE(
    handle    => v_hdnl,
    filename  => 'sample_copied.dmp',
```

```
    directory => 'DATA_PUMP_DIR',
    filetype  => dbms_datapump.ku$_file_type_dump_file );
  DBMS_DATAPUMP.ADD_FILE(
    handle    => v_hdnl,
    filename  => 'sample_imp.log',
    directory => 'DATA_PUMP_DIR',
    filetype  => dbms_datapump.ku$_file_type_log_file);
  DBMS_DATAPUMP.METADATA_FILTER(v_hdnl,'SCHEMA_EXPR','IN (''SCHEMA_1'')');
  DBMS_DATAPUMP.START_JOB(v_hdnl);
END;
/
```

> **Note**
>
> I processi di Data Pump vengono avviati in modo asincrono. Per informazioni sul monitoraggio di un processo di Data Pump, consulta la pagina relativa al monitoraggio dello stato dei processi nella documentazione di Oracle. È possibile visualizzare il contenuto del log di importazione utilizzando la procedura `rdsadmin.rds_file_util.read_text_file`. Per ulteriori informazioni, consulta Lettura dei file in una directory di istanze database.

Puoi verificare l'importazione dei dati visualizzando le tabelle dell'utente nell'istanza database. Ad esempio, la query seguente restituisce il numero di tabelle per *schema_1*.

```
SELECT COUNT(*) FROM DBA_TABLES WHERE OWNER='SCHEMA_1';
```

Fase 7: pulizia

Dopo che i dati sono stati importati, puoi eliminare i file che non intendi conservare. Puoi elencare i file contenuti in `DATA_PUMP_DIR` utilizzando il seguente comando.

```
SELECT * FROM TABLE(rdsadmin.rds_file_util.listdir('DATA_PUMP_DIR')) ORDER BY MTIME;
```

Per eliminare i file non più necessari da `DATA_PUMP_DIR`, utilizza il seguente comando.

```
EXEC UTL_FILE.FREMOVE('DATA_PUMP_DIR','<file name>');
```

Ad esempio, il comando seguente elimina il file denominato `"sample_copied.dmp"`.

```
EXEC UTL_FILE.FREMOVE('DATA_PUMP_DIR','sample_copied.dmp');
```

Importazione con le utilità Oracle di esportazione/importazione

È possibile valutare la possibilità di utilizzare le utilità Oracle di esportazione/importazione per le migrazioni nelle seguenti condizioni:

- Le dimensioni dei dati sono ridotte.
- Non sono richiesti tipi di dati come binary float e double.

Il processo di importazione crea gli oggetti dello schema necessari. Pertanto, non devi eseguire prima uno script per creare gli oggetti.

Il modo più semplice per installare le utilità Oracle di esportazione e importazione è installare Oracle Instant Client. Per scaricare il software, vai su https://www.oracle.com/database/technologies/instant-client.html. Per la documentazione, consulta Instant Client for SQL*Loader, Export and Import nel manuale Oracle Database Utilities.

Per esportare le tabelle e quindi importarle

1. Esporta le tabelle dal database di origine utilizzano il comando `exp`.

   Il comando seguente esporta le tabelle denominate `tab1`, `tab2` e `tab3`. Il file di dump è `exp_file.dmp`.

   ```
   exp cust_dba@ORCL FILE=exp_file.dmp TABLES=(tab1,tab2,tab3) LOG=exp_file.log
   ```

   L'esportazione crea un file dump binario contenente sia lo schema che i dati delle tabelle specificate.
2. Importa lo schema e i dati in un database di destinazione utilizzando il comando `imp`.

   Il comando seguente importa le tabelle `tab1`, `tab2` e `tab3` dal file di dump `exp_file.dmp`.

   ```
   imp cust_dba@targetdb FROMUSER=cust_schema TOUSER=cust_schema \
   TABLES=(tab1,tab2,tab3) FILE=exp_file.dmp LOG=imp_file.log
   ```

Le esportazioni e importazione hanno altre varianti che potrebbero essere più adatte alle esigenze specifiche. Per dettagli completi, consulta la documentazione di Oracle Database.

Importazione utilizzando Oracle SQL*Loader

Potresti valutare l'opportunità di utilizzare Oracle SQL*Loader per database di grandi dimensioni contenenti un numero limitato di oggetti. Poiché il processo di esportazione da un database di origine e di caricamento in un database di destinazione è specifico dello schema, l'esempio seguente crea gli oggetti dello schema di esempio, li esporta da un'origine e quindi carica i dati in un database di destinazione.

Il modo più semplice per installare Oracle SQL*Loader è installare Oracle Instant Client. Per scaricare il software, vai su https://www.oracle.com/database/technologies/instant-client.html. Per la documentazione, consulta Instant Client for SQL*Loader, Export and Import nel manuale Oracle Database Utilities.

Per importare dati utilizzando Oracle SQL*Loader

1. Crea una tabella di origine di esempio utilizzando la seguente istruzione SQL.

```
CREATE TABLE customer_0 TABLESPACE users
   AS (SELECT ROWNUM id, o.*
       FROM   ALL_OBJECTS o, ALL_OBJECTS x
       WHERE  ROWNUM <= 1000000);
```

2. Nell'istanza database RDS per Oracle di destinazione, crea una tabella di destinazione per caricare i dati. La clausola `WHERE 1=2` garantisce la copia della struttura di `ALL_OBJECTS`, ma non delle righe.

```
CREATE TABLE customer_1 TABLESPACE users
  AS (SELECT 0 AS ID, OWNER, OBJECT_NAME, CREATED
      FROM   ALL_OBJECTS
      WHERE  1=2);
```

3. Esporta i dati dal database di origine a un file di testo. L'esempio seguente utilizza SQL*Plus. Per i propri dati, è probabilmente necessario generare uno script che esegue l'esportazione per tutti gli oggetti nel database.

```
ALTER SESSION SET NLS_DATE_FORMAT = 'YYYY/MM/DD HH24:MI:SS'

SET LINESIZE 800 HEADING OFF FEEDBACK OFF ARRAY 5000 PAGESIZE 0
```

```
SPOOL customer_0.out
SET MARKUP HTML PREFORMAT ON
SET COLSEP ','

SELECT id, owner, object_name, created
FROM   customer_0;

SPOOL OFF
```

4. Crea un file di controllo per descrivere i dati. Potrebbe essere necessario scrivere uno script per eseguire questa operazione.

```
cat << EOF > sqlldr_1.ctl
load data
infile customer_0.out
into table customer_1
APPEND
fields terminated by "," optionally enclosed by '"'
(
  id           POSITION(01:10)    INTEGER EXTERNAL,
  owner        POSITION(12:41)    CHAR,
  object_name  POSITION(43:72)    CHAR,
  created      POSITION(74:92)    date "YYYY/MM/DD HH24:MI:SS"
)
```

   Se necessario, copiare i file generati dal codice precedente in un'area di staging, ad esempio un'istanza Amazon EC2.

5. Importa i dati utilizzando SQL*Loader con il nome utente e la password appropriati per il database di destinazione.

```
sqlldr cust_dba@targetdb CONTROL=sqlldr_1.ctl BINDSIZE=10485760 READSIZE=10485760
 ROWS=1000
```

Migrazione con le viste materializzate Oracle

Per eseguire la migrazione di set di dati di grandi dimensioni in modo efficiente, è anche possibile utilizzare la replica delle viste materializzate Oracle. Una replica consente di mantenere la sincronizzazione tra le tabelle di destinazione e le tabelle di origine. Pertanto, puoi passare ad Amazon RDS in un secondo momento, se necessario.

Prima di poter migrare utilizzando le viste materializzate, verifica che siano soddisfatti i seguenti requisiti:

- Configurazione dell'accesso dal database di destinazione al database di origine. Nell'esempio seguente sono state abilitate regole di accesso nel database di origine per permettere la connessione del database RDS per Oracle di destinazione all'origine tramite SQL*Net.
- Crea un collegamento di database tra l'istanza database RDS per Oracle e il database di origine.

Per eseguire la migrazione dei dati utilizzando viste materializzate

1. Nelle istanze RDS per Oracle di origine e di destinazione crea un account utente per il quale sia possibile eseguire l'autenticazione con la stessa password. L'esempio seguente crea un utente denominato `dblink_user`.

```
CREATE USER dblink_user IDENTIFIED BY my-password
  DEFAULT TABLESPACE users
  TEMPORARY TABLESPACE temp;

GRANT CREATE SESSION TO dblink_user;

GRANT SELECT ANY TABLE TO dblink_user;

GRANT SELECT ANY DICTIONARY TO dblink_user;
```

> **Note**
>
> Specifica una password diversa dal prompt mostrato qui come best practice per la sicurezza.

2. Crea un collegamento di database dall'istanza RDS per Oracle di destinazione all'istanza di origine utilizzando il nuovo utente creato.

```
CREATE DATABASE LINK remote_site
  CONNECT TO dblink_user IDENTIFIED BY my-password
  USING '(description=(address=(protocol=tcp) (host=my-host)
    (port=my-listener-port)) (connect_data=(sid=my-source-db-sid)))';
```

> **Note**
>
> Specifica una password diversa dal prompt mostrato qui come best practice per la sicurezza.

3. Testare il collegamento:

```
SELECT * FROM V$INSTANCE@remote_site;
```

4. Creare una tabella di esempio con una chiave primaria e un log della vista materializzata nell'istanza di origine.

```
CREATE TABLE customer_0 TABLESPACE users
  AS (SELECT ROWNUM id, o.*
      FROM   ALL_OBJECTS o, ALL_OBJECTS x
      WHERE  ROWNUM <= 1000000);

ALTER TABLE customer_0 ADD CONSTRAINT pk_customer_0 PRIMARY KEY (id) USING INDEX;

CREATE MATERIALIZED VIEW LOG ON customer_0;
```

5. Nell'istanza database RDS per Oracle di destinazione, crea una vista materializzata.

```
CREATE MATERIALIZED VIEW customer_0
  BUILD IMMEDIATE REFRESH FAST
  AS (SELECT *
      FROM   cust_dba.customer_0@remote_site);
```

6. Nell'istanza database RDS per Oracle di destinazione, aggiorna la vista materializzata.

```
EXEC DBMS_MV.REFRESH('CUSTOMER_0', 'f');
```

7. Elimina la vista materializzata e includi la clausola `PRESERVE TABLE` per mantenere la tabella container della vista materializzata e il relativo contenuto.

```
DROP MATERIALIZED VIEW customer_0 PRESERVE TABLE;
```

   La tabella conservata ha lo stesso nome della vista materializzata eliminata.

Utilizzo di repliche di lettura per Amazon RDS per Oracle

Per configurare la replica tra istanze database di Oracle, è possibile creare database di replica. Per una panoramica delle repliche di lettura di Amazon RDS, consulta Panoramica delle repliche di lettura di Amazon RDS. Per un riepilogo delle differenze tra repliche Oracle e altri motori di database, vedere Differenze tra repliche di lettura per i motori DB.

Argomenti

Panoramica delle repliche RDS per Oracle

Un database replica di Oracle è una copia fisica del database primario. Una replica Oracle in modalità di sola lettura è denominata replica di lettura. Una replica Oracle in modalità montata è denominata replica montata. Oracle Database non consente la scrittura in una replica, ma è possibile promuovere una replica per renderla scrivibile. La replica di lettura promossa contiene i dati replicati fino al momento in cui è stata effettuata la richiesta di promozione.

Nel seguente video viene fornita una panoramica utile del ripristino di emergenza di RDS per Oracle.

Per ulteriori informazioni, leggi il post del blog Ripristino di emergenza gestito con backup automatizzati tra regioni di Amazon RDS per Oracle - Parte 1 e Ripristino di emergenza gestito con backup automatizzati tra regioni di Amazon RDS per Oracle - Parte 2.

Argomenti

- Interruzioni durante la replica Oracle

Repliche di sola lettura e montate

Quando si crea o si modifica una replica Oracle, è possibile inserirla in una delle seguenti modalità:

Sola lettura

Questa è l'impostazione predefinita. Active Data Guard trasmette e applica le modifiche dal database di origine a tutti i database di replica di lettura.

È possibile creare fino a cinque repliche di lettura da un'istanza database di origine. Per informazioni generali sulle repliche di lettura applicabili a tutti i motori DB, consulta Uso delle repliche di lettura dell'istanza database. Per informazioni su Oracle Data Guard, consulta Oracle Data Guard concepts and administration (Concetti e amministrazione di Oracle Data Guard) nella documentazione di Oracle.

Montata

In questo caso, la replica utilizza Oracle Data Guard, ma il database di replica non accetta connessioni utente. L'uso principale per le repliche montate è il disaster recovery tra regioni.

Una replica montata non può gestire un carico di lavoro di sola lettura. La replica montata elimina i file di log redo archiviati dopo averli applicati, indipendentemente dalla policy di conservazione dei log archiviati.

È possibile creare una combinazione di repliche database montate e di sola lettura per la stessa istanza database di origine. È possibile modificare una replica di sola lettura in modalità montata oppure modificare una replica montata in modalità di sola lettura. In entrambi i casi, il database Oracle mantiene l'impostazione di conservazione dei log archiviati.

Repliche di lettura dei CDB

RDS per Oracle supporta le repliche di lettura Data Guard per i CDB Oracle Database 19c e 21c solo nella configurazione a tenant singolo. È possibile creare, gestire e promuovere repliche di lettura in un CDB, proprio come in un non CDB. Sono supportate anche le repliche montate. Si ottengono i seguenti vantaggi:

- Ripristino di emergenza gestito, alta disponibilità e accesso in sola lettura alle repliche
- La possibilità di creare repliche di lettura in un altro modo Regione AWS.

- Integrazione con le API di replica di lettura RDS esistenti: CreateDB e InstanceReadReplica PromoteReadReplicaSwitchoverReadReplica

Per utilizzare questa funzionalità, è necessaria una licenza Active Data Guard e una licenza Oracle Database Enterprise Edition per la replica e per le istanze database primarie. Non ci sono costi aggiuntivi correlati all'utilizzo dell'architettura CDB. I prezzi sono calcolati in base alle istanze database.

Per ulteriori informazioni sulle configurazioni a tenant singolo e multi-tenant dell'architettura CDB, consulta Panoramica dei database CDB RDS per Oracle.

Conservazione dei log di ripristino archiviati

Se un'istanza database primaria non dispone di repliche di lettura tra Regioni, Amazon RDS per Oracle mantiene per un minimo di due ore i registri redo archiviati nell'istanza database di origine. Questo è vero indipendentemente dall'impostazione per `archivelog retention hours` in `rdsadmin.rdsadmin_util.set_configuration`.

RDS elimina i log dall'istanza database sorgente dopo due ore o dopo che il tempo impostato per il periodo di conservazione dell'archivio dei log è passato, a seconda di quale risulta maggiore. RDS elimina i log dalla replica di lettura dopo che il tempo impostato per il periodo di conservazione dei log archiviati è passato, solo se ciò è stato applicato correttamente al database.

In alcuni casi, un'istanza database primaria potrebbe avere una o più repliche di lettura tra regioni. In questa evenienza, Amazon RDS for Oracle mantiene i log delle transazioni sull'istanza database di origine finché non vengono trasmessi e applicati a tutte le repliche di lettura tra regioni. Per informazioni su `rdsadmin.rdsadmin_util.set_configuration`, consultare Conservazione dei log redo archiviati.

Interruzioni durante la replica Oracle

Quando crei una replica di lettura, Amazon RDS acquisisce uno snapshot DB dell'istanza database di origine e avvia la replica. L'istanza DB di origine subisce una sospensione di I/O molto breve quando inizia l'operazione di snapshot DB. La sospensione I/O dura in genere circa un secondo. Puoi evitare l'interruzione delle operazioni di I/O se l'istanza database di origine è un'implementazione Multi-AZ, perché in questo caso lo snapshot viene acquisito dall'istanza database secondaria.

Lo snapshot del DB diventa la replica Oracle. Amazon RDS imposta i parametri e le autorizzazioni necessari per il database di origine e la replica senza interruzioni del servizio. Analogamente, se si elimina una replica, non si verifica alcuna interruzione.

Requisiti e considerazioni sulle repliche RDS per Oracle

Prima di creare una replica Oracle, esamina i requisiti e le considerazioni riportati di seguito.

Argomenti

Requisiti di versione e licenza per le repliche RDS per Oracle

Prima di creare una replica RDS per Oracle, considera i seguenti requisiti:

- Se la replica è in modalità di sola lettura, assicurarsi di disporre di una licenza Active Data Guard. Se si posiziona la replica in modalità montata, non è necessaria una licenza Active Data Guard. Solo il motore database Oracle supporta le repliche montate.
- Le repliche Oracle sono supportate solo sul motore Oracle Enterprise Edition (EE).
- Le repliche Oracle dei non CDB sono supportate solo per le istanze database create utilizzando la versione di Oracle Database 12c Release 1 (12.1.0.2.v10) e release successive a 12c e per le istanze non CDB di Oracle Database 19c.
- Le repliche Oracle dei CDB sono supportate solo per le istanze CDB create utilizzando Oracle Database versione 19c e successive.
- Le repliche di lettura Oracle sono disponibili solo per le istanze database in esecuzione su classi di istanze database con due o più vCPU. Un'istanza database di origine non può utilizzare la classe di istanze db.t3.micro o db.t3.small.
- La versione del motore del database Oracle dell'istanza database di origine e tutte le relative repliche di lettura devono essere uguali. Amazon RDS aggiorna la primaria immediatamente dopo l'aggiornamento dell’istanza database di origine, a prescindere dalla finestra di manutenzione della replica. Per gli aggiornamenti delle versioni principali delle repliche tra regioni, Amazon RDS esegue automaticamente le operazioni seguenti:
  - Genera automaticamente un gruppo di opzioni per la versione di destinazione
  - Copia tutte le opzioni e le impostazioni delle opzioni dal gruppo di opzioni originale al nuovo gruppo di opzioni

- Associa la replica aggiornata tra regioni al nuovo gruppo di opzioni

Per ulteriori informazioni sull'aggiornamento della versione del motore del database, consultare Aggiornamento del motore di database RDS per Oracle.

Considerazioni sul gruppo di opzioni per le repliche RDS per Oracle

Prima di creare una replica RDS per Oracle, considera i seguenti requisiti:

- Se la replica Oracle si trova nella stessa regione AWS dell'istanza database di origine, assicurarsi che appartenga allo stesso gruppo di opzioni dell'istanza database di origine. Le modifiche al gruppo di opzioni di origine o all'appartenenza al gruppo di opzioni di origine si propagano alle repliche. Queste modifiche vengono applicate alle repliche immediatamente dopo l'applicazione all'istanza database di origine, indipendentemente dalla finestra di manutenzione delle repliche.

  Per ulteriori informazioni sui gruppi di opzioni, consulta Uso di gruppi di opzioni.
- Quando si crea una replica tra regioni RDS per Oracle, Amazon RDS crea un gruppo di opzioni dedicato.

  Non è possibile rimuovere una replica tra regioni RDS per Oracle dal suo gruppo di opzioni dedicato. Nessun'altra istanza database può usare il gruppo di opzioni dedicato per una replica tra regioni RDS per Oracle.

  È possibile aggiungere o rimuovere solo le seguenti opzioni non replicate da un gruppo di opzioni dedicato:
  - `NATIVE_NETWORK_ENCRYPTION`
  - `OEM`
  - `OEM_AGENT`
  - `SSL`

  Per aggiungere altre opzioni a una replica tra regioni RDS per Oracle, aggiungerle al gruppo di opzioni dell'istanza database di origine. L'opzione è installata anche su tutte le repliche dell'istanza database di origine. Per le opzioni con licenza, assicurarsi che siano disponibili licenze sufficienti per le repliche.

  Quando promuovi una replica tra regioni RDS per Oracle, tale replica si comporta come qualsiasi altra istanza database di Oracle, compresa la gestione delle opzioni. Puoi promuovere una replica esplicitamente o implicitamente eliminando la sua istanza database di origine.

Per ulteriori informazioni sui gruppi di opzioni, consulta Uso di gruppi di opzioni.

Considerazioni su backup e ripristino per le repliche RDS per Oracle

Prima di creare una replica RDS per Oracle, considera i seguenti requisiti:

- Per creare snapshot delle repliche RDS per Oracle o attivare i backup automatici, assicurati di impostare manualmente il periodo di conservazione dei backup. Per impostazione predefinita, i backup automatici non sono attivati.
- Quando si ripristina un backup di repliche, si esegue il ripristino corrispondente all'ora del database e non al momento in cui il backup è stato eseguito. L'ora del database si riferisce all'ora dell'ultima transazione applicata ai dati nel backup. La differenza è significativa perché una replica può fare riferimento a un'ora più o meno precedente all'ora del database primario.

  Per trovare la differenza, usa il comando `describe-db-snapshots`. Confronta `snapshotDatabaseTime`, ovvero l'ora del database del backup di repliche e il campo `OriginalSnapshotCreateTime`, che è l'ultima transazione applicata al database primario.

Requisiti e limitazioni di Oracle Data Guard per le repliche RDS per Oracle

Prima di creare una replica di RDS per Oracle, prendi nota dei seguenti requisiti e limitazioni:

- Se l'istanza database primaria utilizza la configurazione a tenant singolo dell'architettura multi-tenant, considera quanto segue:
  - Devi utilizzare Oracle Database versione 19c o successive con Enterprise Edition.
  - L'istanza CDB primaria deve trovarsi in un ciclo di vita `ACTIVE`.
  - Non puoi convertire un'istanza primaria non CDB in un'istanza CDB e convertire le rispettive repliche nella stessa operazione. Elimina, invece, le repliche non CDB, converti l'istanza database primaria in CDB e quindi crea nuove repliche
- Assicurati che un trigger di accesso su un'istanza database primaria consenta l'accesso all'utente `RDS_DATAGUARD` e a qualsiasi utente il cui valore `AUTHENTICATED_IDENTITY` sia `RDS_DATAGUARD` o `rdsdb`. Inoltre, il trigger non deve impostare lo schema corrente per l'utente `RDS_DATAGUARD`.

- Per evitare di bloccare le connessioni dal processo di broker Data Guard, non abilitare le sessioni con restrizioni. Per ulteriori informazioni sulle sessioni con restrizioni, consulta Abilitazione e disabilitazione delle sessioni limitate.

Considerazioni varie sulle repliche RDS per Oracle

Prima di creare una replica RDS per Oracle, considera i seguenti requisiti:

- Se l'istanza database è un'origine di una o più repliche tra regioni, il database di origine mantiene i suoi log redo archiviati finché non vengono applicati a tutte le repliche tra regioni. I log redo archiviati potrebbero causare un aumento del consumo di storage.
- Per evitare di interrompere l'automazione RDS, i trigger di sistema devono consentire a utenti specifici di accedere al database primario e di replica. I trigger di sistema includono trigger DDL, di accesso e ruolo database. Si consiglia di aggiungere codice ai trigger per escludere gli utenti elencati nel codice di esempio riportato di seguito:

```
-- Determine who the user is
SELECT SYS_CONTEXT('USERENV','AUTHENTICATED_IDENTITY') INTO CURRENT_USER FROM DUAL;
-- The following users should always be able to login to either the Primary or
 Replica
IF CURRENT_USER IN ('master_user', 'SYS', 'SYSTEM', 'RDS_DATAGUARD', 'rdsdb') THEN
RETURN;
END IF;
```

- Il rilevamento delle modifiche di blocco è supportato per le repliche di sola lettura, ma non per le repliche montate. Puoi modificare una replica montata in una replica di sola lettura e quindi attivare il rilevamento delle modifiche di blocco. Per ulteriori informazioni, consulta Abilitazione e disabilitazione del monitoraggio delle modifiche dei blocchi..

Preparazione alla creazione di una replica Oracle

Prima di iniziare a utilizzare la replica, eseguire le operazioni seguenti.

Argomenti

- Abilitazione di backup automatici
- Attivazione della modalità di registrazione forzata
- Modifica della configurazione di registrazione

- Impostazione del parametro MAX_STRING_SIZE
- Ridimensionare le risorse di calcolo e storage.

Abilitazione di backup automatici

Prima di poter utilizzare un'istanza database come istanza database di origine, devi abilitare i backup automatici sull'istanza database di origine. Per informazioni su come eseguire questa procedura, consulta Abilitazione dei backup automatici.

Attivazione della modalità di registrazione forzata

Si consiglia di attivare la modalità di registrazione forzata. In modalità di registrazione forzata, il database Oracle scrive i record redo anche quando `NOLOGGING` viene utilizzato con istruzioni DDL (Data Definition Language).

Per attivare la modalità di registrazione forzata

1. Accedere al database Oracle utilizzando uno strumento client, ad esempio SQL Developer.
2. Attivare la modalità di registrazione forzata eseguendo la procedura seguente.

```
exec rdsadmin.rdsadmin_util.force_logging(p_enable => true);
```

Per ulteriori informazioni su questa procedura, consultare Impostazione accesso forzato.

Modifica della configurazione di registrazione

Per n redo log online di dimensione m, RDS crea automaticamente n +1 log di standby di dimensione m sull'istanza DB principale e su tutte le repliche. Ogni volta che si modifica la configurazione di registrazione sul primario, le modifiche si propagano automaticamente alle repliche.

Se modificate la configurazione di registrazione, tenete conto delle seguenti linee guida:

- Si consiglia di completare le modifiche prima di rendere un'istanza DB l'origine delle repliche. RDS for Oracle supporta anche l'aggiornamento dell'istanza dopo che è diventata una fonte.
- Prima di modificare la configurazione di registrazione sull'istanza DB principale, verifica che ogni replica disponga di spazio di archiviazione sufficiente per ospitare la nuova configurazione.

Puoi modificare la configurazione di registrazione per un'istanza DB utilizzando le procedure `rdsadmin.rdsadmin_util.add_logfile` Amazon RDS e. `rdsadmin.rdsadmin_util.drop_logfile` Per ulteriori informazioni, consulta Aggiunta di log redo online e Eliminazione di log redo online.

Impostazione del parametro MAX_STRING_SIZE

Prima di creare una replica di lettura, assicurarsi che l'impostazione del parametro `MAX_STRING_SIZE` sia la stessa sull'istanza database di origine e sulla replica di lettura. Puoi ottenere tale risultato associando entrambi gli oggetti allo stesso gruppo di parametri. Se disponi di gruppi di parametri diversi per l'origine e la replica di lettura, puoi impostare `MAX_STRING_SIZE` sullo stesso valore. Per ulteriori informazioni su questo parametro, consulta Attivazione dei tipi di dati estesi per una nuova istanza database.

Ridimensionare le risorse di calcolo e storage.

Assicurati che l'istanza database di origine e le sue repliche siano dimensionate correttamente, in termini di capacità di calcolo e storage, per adattarsi al loro carico operativo. Se una replica di lettura raggiunge la massima capacità di risorse calcolo, rete o archiviazione, smette di ricevere o applicare modifiche dalla sua origine. Amazon RDS for Oracle non interviene per attenuare un elevato ritardo di replica tra un'istanza database di origine e le sue repliche di lettura. Puoi modificare le risorse di storage e CPU di una replica in modo indipendente dalla sua origine e dalle altre repliche.

Creazione di una replica RDS per Oracle in modalità montata

Per impostazione predefinita, le repliche Oracle sono di sola lettura. Per creare una replica in modalità montata, utilizzare la console, AWS CLI o l'API RDS.

Console

Per creare una replica montata da un'istanza Oracle DB di origine

1. Accedi alla AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, scegliere Databases (Database).
3. Scegliere l'istanza Oracle DB che si desidera utilizzare come origine per una replica montata.
4. Per Actions (Operazioni), scegliere Create replica (Crea replica).
5. Per la modalità Replica, scegliere Montato.

6. Scegliere le impostazioni che si desiderano usare. Per DB instance identifier (Identificatore istanze DB) inserire un nome per la replica di lettura. Modificare le altre impostazioni nel modo necessario.
7. Per Regioni, scegliere la regione in cui verrà avviata la replica montata.
8. Scegli la dimensione e il tipo di archiviazione dell'istanza. Consigliamo di usare la stessa classe di istanza database e lo stesso tipo di storage dell'istanza database di origine per la replica di lettura.
9. Per Multi-AZ deployment (Implementazione Multi-AZ) scegliere Creare un'istanza di standby per creare una versione di standby delle replica in un'altra zona di disponibilità per il supporto del failover per la replica montata. La creazione della replica montata come un'istanza database Multi-AZ non dipende dal fatto che il database di origine sia un'istanza database Multi-AZ.
10. Scegliere le altre impostazioni che si desiderano usare.
11. Scegliere Crea replica.

Nella pagina Database la replica montata ha il ruolo Replica.

AWS CLI

Per creare una replica Oracle in modalità montata, `--replica-mode` impostare su `mounted` nel AWS CLI comando create-db-instance-read-replica.

Example

PerLinux, o: macOS Unix

```
aws rds create-db-instance-read-replica \
    --db-instance-identifier myreadreplica \
    --source-db-instance-identifier mydbinstance \
    --replica-mode mounted
```

Per Windows:

```
aws rds create-db-instance-read-replica ^
    --db-instance-identifier myreadreplica ^
    --source-db-instance-identifier mydbinstance ^
    --replica-mode mounted
```

Per modificare una replica di sola lettura in uno stato montato, imposta su `mounted` nel `--replica-mode` comando. AWS CLI modify-db-instance Per posizionare una replica montata in modalità di sola lettura, imposta `--replica-mode` su `open-read-only`.

API RDS

Per creare una replica Oracle in modalità montata, specificare `ReplicaMode=mounted` nell'API RDS l'operazione CreateDB. InstanceReadReplica

Modifica della modalità di replica RDS per Oracle

Per modificare la modalità di replica di una replica esistente, utilizzare la console, AWS CLI o l'API RDS. Quando si passa alla modalità montata, la replica disconnette tutte le connessioni attive. Quando si passa alla modalità di sola lettura, Amazon RDS inizializza Active Data Guard.

L'operazione di modifica può richiedere alcuni minuti. Durante l'operazione, lo stato dell'istanza database cambia in modifying (modifica). Per ulteriori informazioni sulle modifiche di stato, consulta Visualizzazione dello stato dell'istanza database di Amazon RDS .

Console

Per modificare la modalità di replica di una replica Oracle da montata a sola lettura

1. Accedi alla AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, scegliere Databases (Database).
3. Scegliere il database di replica montato.
4. Scegliere Modify (Modifica).
5. In Modalità replica, scegliere Sola lettura.
6. Scegliere le altre impostazioni che si desiderano usare.
7. Scegli Continue (Continua).
8. In Scheduling of Modifications (Pianificazione delle modifiche), scegliere Apply immediately (Applica immediatamente).
9. Scegliere Modify DB Instance (Modifica istanza database).

AWS CLI

Per modificare una replica di lettura in modalità montata, `--replica-mode` impostate su `mounted` nel AWS CLI comando modify-db-instance. Per modificare una replica montata in modalità di sola lettura, imposta `--replica-mode` su `open-read-only`.

Example

Per LinuxmacOS, oUnix:

```
aws rds modify-db-instance \
    --db-instance-identifier myreadreplica \
    --replica-mode mode
```

Per Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier myreadreplica ^
    --replica-mode mode
```

API RDS

Per modificare una replica di sola lettura in modalità montata, impostare `ReplicaMode=mounted` su ModifyDBInstance. Per modificare una replica montata in modalità di sola lettura, impostare `ReplicaMode=read-only`.

Utilizzo dei backup di repliche RDS per Oracle

È possibile creare e ripristinare i backup di una replica RDS per Oracle. Sono supportati sia i backup automatici che gli snapshot manuali. Per ulteriori informazioni, consulta Backup, ripristino ed esportazione dei dati. Le sezioni seguenti descrivono le principali differenze tra la gestione dei backup di una replica primaria e di una replica RDS per Oracle.

Attivazione di RDS per i backup di repliche Oracle

Per impostazione predefinita, i backup automatici non sono attivati per una replica Oracle. Per attivare i backup automatici, imposta il periodo di conservazione dei backup su un valore diverso da zero positivo.

Console

Per abilitare immediatamente i backup automatici

1. Accedi alla AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel pannello di navigazione, scegli Database, quindi scegli l'istanza database o il cluster di database multi-AZ che vuoi modificare.
3. Scegli Modifica.
4. In Periodo di retention dei backup, scegli un valore positivo diverso da zero, ad esempio 3 giorni.
5. Scegli Continue (Continua).
6. Scegliere Apply immediately (Applica immediatamente).
7. Scegli Modifica istanza database o Modifica cluster per salvare le modifiche e abilitare i backup automatici.

AWS CLI

Per abilitare i backup automatici, utilizza il comando AWS CLI `modify-db-instance` o `modify-db-cluster`.

Includere i seguenti parametri:

- `--db-instance-identifier` (o `--db-cluster-identifier` per un cluster di database multi-AZ)
- `--backup-retention-period`
- `--apply-immediately` o `--no-apply-immediately`

In questo esempio vengono abilitati i backup automatici impostando il periodo di conservazione dei backup su tre giorni. Le modifiche vengono applicate immediatamente.

Example

Per LinuxmacOS, oUnix:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance  \
```

```
    --backup-retention-period 3 \
    --apply-immediately
```

Per Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance  ^
    --backup-retention-period 3 ^
    --apply-immediately
```

API RDS

Per abilitare i backup automatici, utilizza l'operazione API RDS `ModifyDBInstance` o `ModifyDBCluster` con i seguenti parametri obbligatori:

- `DBInstanceIdentifier` o `DBClusterIdentifier`
- `BackupRetentionPeriod`

Ripristino di un backup di repliche RDS per Oracle

È possibile ripristinare un backup di repliche Oracle così come è possibile ripristinare un backup dell'istanza primaria. Per ulteriori informazioni, consulta gli argomenti seguenti:

- Ripristino da uno snapshot database
- Ripristino a un'ora specifica per un'istanza database

Quando si esegue il ripristino di un backup di repliche, è importante determinare il momento in cui eseguire il ripristino. L'ora del database si riferisce all'ora dell'ultima transazione applicata ai dati nel backup. Quando si ripristina un backup di repliche, si esegue il ripristino corrispondente all'ora del database e non al momento in cui il backup è stato completato. La differenza è significativa perché una replica RDS per Oracle può fare riferimento a un'ora più o meno precedente all'ora del database primario. Pertanto, l'ora del database di un backup di repliche, e quindi il momento in cui viene ripristinato, potrebbe essere molto precedente al momento della creazione del backup.

Per trovare la differenza tra l'ora del database e l'ora di creazione, utilizza il comando `describe-db-snapshots`. Confronta `SnapshotDatabaseTime`, ovvero l'ora del database del backup di repliche e il campo `OriginalSnapshotCreateTime`, che è l'ultima transazione applicata al database primario. L'esempio seguente visualizza la differenza tra due ore.

```
aws rds describe-db-snapshots \
    --db-instance-identifier my-oracle-replica
    --db-snapshot-identifier my-replica-snapshot

{
    "DBSnapshots": [
        {
            "DBSnapshotIdentifier": "my-replica-snapshot",
            "DBInstanceIdentifier": "my-oracle-replica",
            "SnapshotDatabaseTime": "2022-07-26T17:49:44Z",
            ...
            "OriginalSnapshotCreateTime": "2021-07-26T19:49:44Z"
        }
    ]
}
```

Esecuzione di uno switchover Oracle Data Guard

Uno switchover è un'inversione di ruolo tra un database primario e un database in standby. Durante uno switchover, il database primario originale passa a un ruolo di standby, mentre il database in standby originale passa al ruolo primario.

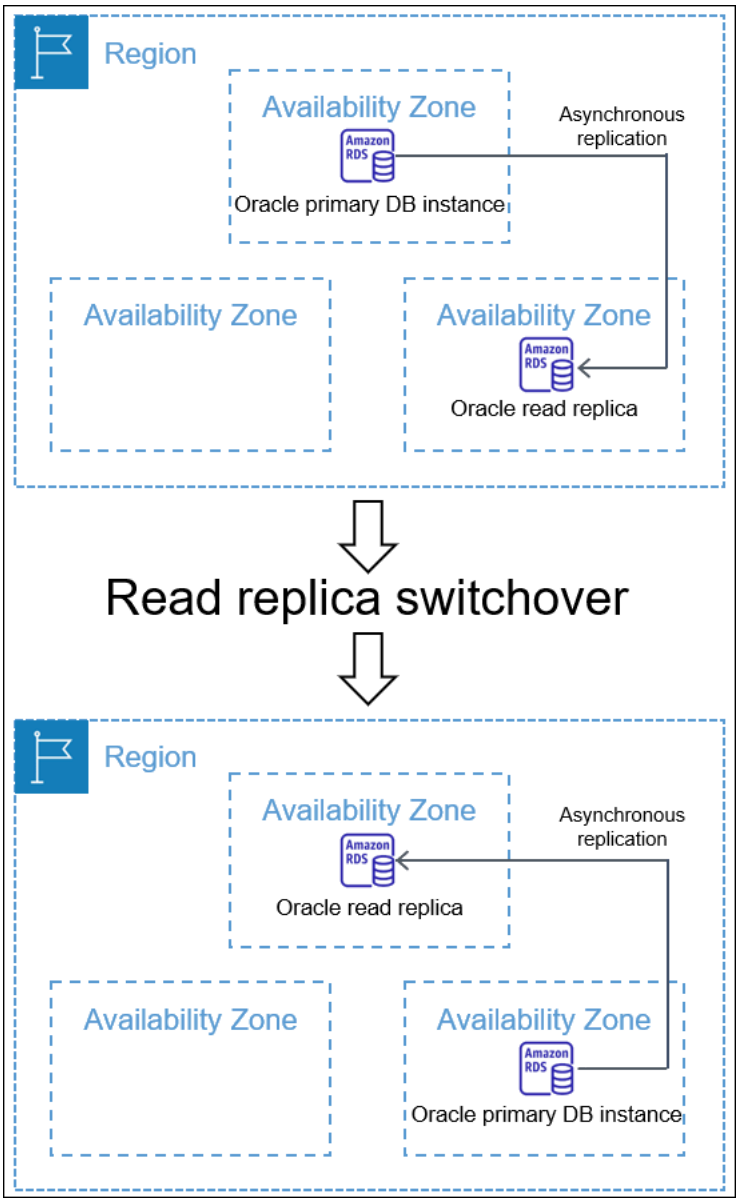
In un ambiente Oracle Data Guard, un database primario supporta uno o più database in standby. È possibile eseguire una transizione di ruolo gestita e basata sullo switchover da un database primario a un database in standby. Uno switchover è un'inversione di ruolo tra un database primario e un database in standby. Durante uno switchover, il database primario originale passa a un ruolo di standby, mentre il database in standby originale passa al ruolo primario.

Argomenti

Panoramica sullo switchover Oracle Data Guard

Amazon RDS supporta una transizione di ruolo completamente gestita e basata sullo switchover per le repliche Oracle Database. È possibile avviare uno switchover solo a un database in standby montato o aperto in modalità di sola lettura.

Le repliche possono risiedere in zone di disponibilità (AZ) separate Regioni AWS o diverse di una singola regione. Sono tutte Regioni AWS supportate.



Un passaggio al digitale è diverso da una promozione di repliche di lettura. In uno switchover, le istanze DB di origine e di replica cambiano ruolo. In una promozione, una replica di lettura diventa un'istanza DB di origine, ma l'istanza DB di origine non diventa una replica. Per ulteriori informazioni, consulta Promozione di una replica di lettura a istanza database standalone.

Argomenti

- Vantaggi dello switchover Oracle Data Guard
- Versioni di Oracle Database supportate

Vantaggi dello switchover Oracle Data Guard

Proprio come per le repliche di lettura RDS per Oracle, uno switchover gestito utilizza Oracle Data Guard. L'operazione è stata ideata in modo da annullare il rischio di perdite di dati. Amazon RDS automatizza i seguenti passaggi dello switchover:

- Inverte i ruoli del database primario e del database in standby specificato, impostando il nuovo database in standby sullo stesso stato (montato o di sola lettura) del database in standby originale
- Garantisce la consistenza dei dati
- Conserva la configurazione di replica dopo la transizione
- Supporta inversioni ripetute, consentendo al nuovo database in standby di tornare al ruolo primario originale

Versioni di Oracle Database supportate

Lo switchover Oracle Data Guard è supportato nelle seguenti versioni:

- Oracle Database 19c
- Oracle Database 12c Release 2 (12.2)
- Oracle Database 12c Release 1 (12.1) usando PSU 12.1.0.2.v10 o versioni successive

Costo dello switchover Oracle Data Guard

La funzionalità di switchover di Oracle Data Guard non comporta costi aggiuntivi. Oracle Database Enterprise Edition include il supporto per i database in standby in modalità montata. Per aprire i database in standby in modalità di sola lettura, è necessaria l'opzione Oracle Active Data Guard.

Come funziona lo switchover Oracle Data Guard

Lo switchover di Oracle Data Guard è un'operazione completamente gestita. Per avviare lo switchover per un database in standby, esegui il comando CLI `switchover-read-replica`. Amazon RDS modifica quindi i ruoli di database primario e in standby nella configurazione di replica.

I ruoli di database in standby originale e primario originale esistono già prima dello switchover. I ruoli di database nuovo in standby e nuovo primario esistono dopo lo switchover. Una replica bystander è

un database di replica che funge da database in standby nell'ambiente Oracle Data Guard e che non cambia ruolo.

Argomenti

Fasi dello switchover Oracle Data Guard

Per effettuare lo switchover, Amazon RDS deve eseguire le seguenti operazioni:

1. Bloccare le nuove transazioni sul database primario originale. Durante lo switchover, Amazon RDS interrompe la replica per tutti i database nella configurazione di Oracle Data Guard in uso. Durante lo switchover, il database primario originale non è in grado di elaborare richieste di scrittura.
2. Inviare e applicare le transazioni non applicate al database di standby originale.
3. Riavviare il nuovo database in standby in modalità di sola lettura o montata. La modalità dipende dallo stato di apertura del database in standby originale prima dello switchover.
4. Aprire il nuovo database primario in modalità lettura-scrittura.

Dopo lo switchover Oracle Data Guard

Amazon RDS scambia i ruoli dei database primario e in standby. Sarà tua la responsabilità di riconnettere l'applicazione e di eseguire qualsiasi altra configurazione desiderata.

Argomenti

Criteri di successo

Lo switchover Oracle Data Guard ha esito positivo quando il database in standby originale esegue le seguenti operazioni:

- Passaggio al suo ruolo di nuovo database primario
- Completamento della riconfigurazione

Per ridurre i tempi di inattività, il nuovo database primario diventa attivo il prima possibile. Poiché Amazon RDS configura le repliche bystander in modo asincrono, queste repliche potrebbero diventare attive dopo il database primario originale.

Connessione al nuovo database primario

Amazon RDS non propagherà le attuali connessioni di database al nuovo database primario dopo lo switchover. Una volta completato lo switchover Oracle Data Guard, ricollega l'applicazione al nuovo database primario.

Configurazione del nuovo database primario

Per effettuare lo switchover al nuovo database primario, Amazon RDS modifica la modalità di apertura del database in standby originale. Il cambio di ruolo è l'unica modifica apportata al database. Amazon RDS non configura le caratteristiche, come la replica Multi-AZ.

Se si esegue lo switchover a una replica tra più regioni con opzioni diverse, il nuovo database primario conserva le proprie opzioni. Amazon RDS non eseguirà la migrazione delle opzioni sul database principale originale. Se il database primario originale aveva opzioni come SSL, NNE, OEM e OEM_AGENT, Amazon RDS non le propaga al nuovo database primario.

Preparazione per lo switchover Oracle Data Guard

Prima di iniziare lo switchover Oracle Data Guard, verifica che il tuo ambiente di replica soddisfi i seguenti requisiti:

- Il database in standby originale è montato o aperto in modalità di sola lettura.
- I backup automatici sono abilitati sul database di standby originale.
- Il database primario originale e il database in standby originale sono in uno stato disponibile.
- Il database primario originale e il database di standby originale non hanno azioni di manutenzione in sospeso.
- Il database in standby originale è nello stato di replica.
- Non si sta tentando di avviare uno switchover durante un ciclo di vita dello switchover per il database primario o il database in standby. Se un database di replica viene riconfigurato dopo uno switchover, Amazon RDS impedisce di avviare un altro switchover.

> **Note**
>
> Una replica bystander è una replica nella configurazione di Oracle Data Guard che non è la destinazione dello switchover. Le repliche bystander possono avere qualsiasi stato durante il passaggio.

- Il database in standby originale ha una configurazione il più vicino possibile al database primario originale. Si supponga uno scenario in cui i database in standby primario e originale abbiano opzioni diverse. Una volta completato lo switchover, Amazon RDS non riconfigura automaticamente il nuovo database primario in modo che abbia le stesse opzioni del database primario originale.
- È necessario configurare l'implementazione multi-AZ desiderata prima di avviare uno switchover. Amazon RDS non gestisce multi-AZ come parte dello switchover. L'implementazione multi-AZ rimane così com'è.

  Supponi che db_maz sia il database principale in un'implementazione multi-AZ e db_saz sia una replica single-AZ. Avvii uno switchover da db_maz a db_saz. Al termine, db_maz è un database di replica multi-AZ e db_saz è un database primario Single-AZ. Il nuovo database principale ora non è protetto da un'implementazione multi-AZ.
- In preparazione allo switchover tra regioni, il database principale non utilizza lo stesso gruppo di opzioni di un'istanza database al di fuori della configurazione di replica. Affinché lo switchover tra regioni venga eseguito, il database principale corrente e le relative repliche di lettura devono essere le uniche istanze database a utilizzare il gruppo di opzioni del database principale corrente. In caso contrario, Amazon RDS impedisce lo switchover.

Avvio dello switchover Oracle Data Guard

È possibile eseguire lo switchover di una replica di lettura RDS per Oracle al ruolo primario e della precedente istanza database primaria a un ruolo di replica.

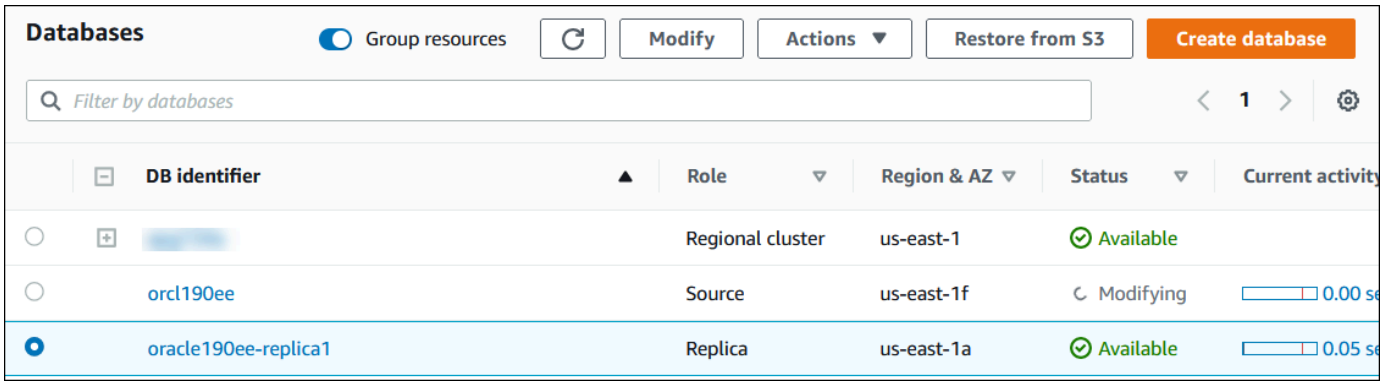
Console

Per eseguire lo switchover di una replica di lettura Oracle al ruolo DB primario

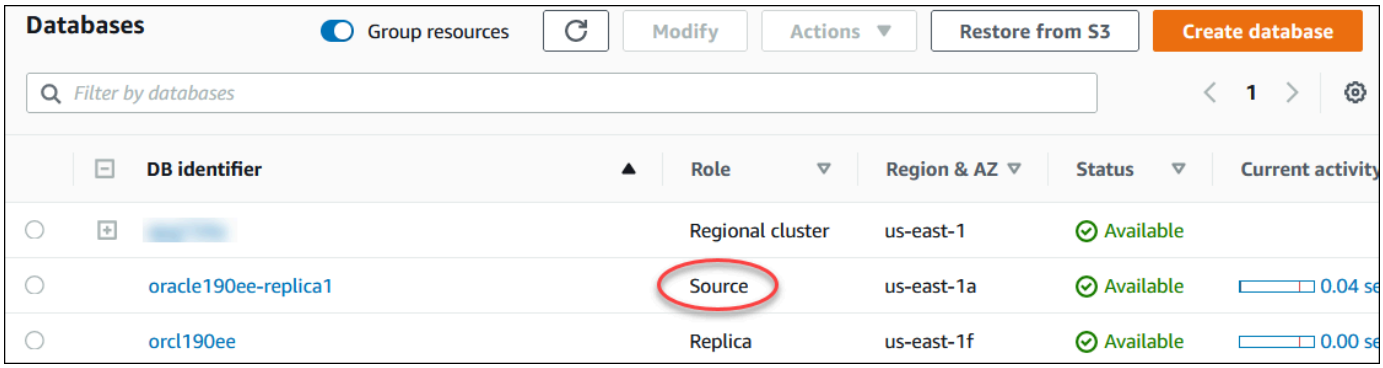
1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nella console Amazon RDS scegliere Databases (Database).

Verrà visualizzato il riquadro Databases (Database). Ogni replica di lettura mostra la Replica nella colonna Role (Ruolo).

3. Scegli la replica di lettura per cui eseguire lo switchover al ruolo primario.
4. In Actions (Operazioni), scegli Switch over replica (Esegui switchover replica).
5. Scegli I acknowledge (Accetto). Scegli quindi Switch over replica (Esegui switchover replica).
6. Nella pagina Databases (Database), monitora lo stato di avanzamento dello switchover.



Una volta completato lo switchover, il ruolo di destinazione dello switchover cambia da Replica a Source (Origine).



AWS CLI

Per passare da una replica Oracle al ruolo DB primario, usa il AWS CLI `switchover-read-replica`comando. Gli esempi seguenti inseriscono la replica Oracle denominata *`replica-to-be-made-primary`* nel nuovo database primario.

Example

PerLinux, omacOS: Unix

```
aws rds switchover-read-replica \
    --db-instance-identifier replica-to-be-made-primary
```

Per Windows:

```
aws rds switchover-read-replica ^
    --db-instance-identifier replica-to-be-made-primary
```

API RDS

Per eseguire lo switchover di una replica Oracle al ruolo database primario, richiama l'operazione `SwitchoverReadReplica` dell'API RDS con il parametro obbligatorio `DBInstanceIdentifier`. Questo parametro specifica il nome della replica Oracle a cui si desidera assegnare il ruolo di DB primario.

Monitoraggio dello switchover Oracle Data Guard

Per controllare lo stato delle tue istanze, usa il comando AWS `describe-db-instances` CLI. Il seguente comando verifica lo stato dell'istanza database *orcl2*. Questo database, che era un database in standby prima dello switchover, è il nuovo database primario dopo lo switchover.

```
aws rds describe-db-instances \
    --db-instance-identifier orcl2
```

Per confermare il corretto completamento dello switchover, esegui una query su V `$DATABASE.OPEN_MODE`. Verifica che il valore del il nuovo database primario sia `READ WRITE`.

```
SELECT OPEN_MODE FROM V$DATABASE;
```

Per cercare gli eventi relativi allo switchover, utilizzate il comando CLI. AWS `describe-events` L'esempio seguente cerca gli eventi nell'istanza *ocl2*.

```
aws rds describe-events \
    --source-identifier orcl2 \
    --source-type db-instance
```

Risoluzione dei problemi relativi alle repliche Oracle

In questa sezione vengono descritti i possibili problemi di replica e le soluzioni.

Argomenti

Monitoraggio del ritardo della replica Oracle

Per monitorare il ritardo della replica Amazon CloudWatch, visualizzare il parametro `ReplicaLag` Amazon RDS. Per ulteriori informazioni sul ritardo della replica, consulta Monitoraggio della replica di lettura e CloudWatch Parametri Amazon per Amazon RDS.

Per una replica di lettura, se il ritardo è troppo lungo, esegui una query nelle visualizzazioni seguenti:

- `V$ARCHIVED_LOG` – Mostra quali commit sono stati applicati alla replica di lettura.
- `V$DATAGUARD_STATS` – Mostra un'analisi dettagliata dei componenti che costituiscono il parametro `ReplicaLag`.
- `V$DATAGUARD_STATUS` – Mostra l'output del log dei processi di interni di replica di Oracle.

Per una replica montata, se il ritardo è troppo lungo, non è possibile eseguire query nelle visualizzazioni `V$`. Effettua invece le seguenti operazioni:

- Controlla il parametro `ReplicaLag` in CloudWatch.
- Controlla il file di log degli avvisi per la replica nella console. Cerca gli errori nei messaggi di ripristino. I messaggi includono il numero di sequenza di registro, che è possibile confrontare con il numero di sequenza principale. Per ulteriori informazioni, consulta File di log del database Oracle.

Risoluzione dei problemi di replica Oracle dopo l'aggiunta o la modifica dei trigger

Se si aggiungono o si modificano trigger e se la replica non riesce in seguito, i trigger potrebbero essere il problema. Assicurarsi che il trigger escluda i seguenti utenti, richiesti da RDS per la replica:

- Account utente con privilegi di amministratore
- `SYS`
- `SYSTEM`
- `RDS_DATAGUARD`
- `rdsdb`

Per ulteriori informazioni, consulta Considerazioni varie sulle repliche RDS per Oracle.

Aggiunta di opzioni alle istanze database Oracle

In Amazon RDS, una opzione è una funzionalità aggiuntiva. Di seguito, è possibile trovare una descrizione delle opzioni che è possibile aggiungere a istanze Amazon RDS che eseguono il motore database di Oracle.

Argomenti

Panoramica sulle opzioni database Oracle

Per abilitare queste opzioni per database Oracle, dovrai aggiungerle a un gruppo di opzioni e quindi associare il gruppo di opzioni all'istanza database. Per ulteriori informazioni, consulta Uso di gruppi di opzioni.

Argomenti

Riepilogo delle opzioni database Oracle

Puoi aggiungere le seguenti opzioni per le istanze database Oracle.

Opzione	ID opzione
Integrazione Amazon S3	`S3_INTEGRATION`
Oracle Application Express (APEX)	`APEX` `APEX-DEV`
Oracle Enterprise Manager	`OEM` `OEM_AGENT`
Oracle Java Virtual Machine	`JVM`
Oracle Label Security	`OLS`
Oracle Locator	`LOCATOR`
Oracle Multimedia	`MULTIMEDIA`
Oracle native network encryption	`NATIVE_NETWORK_ENCRYPTION`
Oracle OLAP	`OLAP`
Oracle Secure Sockets Layer	`SSL`
Oracle Spatial	`SPATIAL`
Oracle SQLT	`SQLT`

Opzione	ID opzione
Oracle Statspack	`STATSPACK`
Fuso orario Oracle	`Timezone`
Aggiornamento automatico dei file di fuso orario Oracle	`TIMEZONE_FILE_AUTOUPGRADE`
Oracle Transparent Data Encryption	`TDE`
UTL_MAIL di Oracle	`UTL_MAIL`
Oracle XML DB	`XMLDB`

Opzioni supportate per diverse edizioni

RDS per Oracle impedisce di aggiungere opzioni a un'edizione se non sono supportate. Per scoprire quali opzioni RDS sono supportate in diverse edizioni del database Oracle, utilizzare il comando `aws rds describe-option-group-options`. Nell'esempio seguente vengono elencate le opzioni supportate per il database Oracle 19c Enterprise Edition.

```
aws rds describe-option-group-options \
    --engine-name oracle-ee \
    --major-engine-version 19
```

Per ulteriori informazioni, consulta describe-option-group-optionsla AWSCLI Command Reference.

Requisiti di memoria per opzioni specifiche

L'esecuzione di alcune opzioni nella tua istanza database richiede memoria aggiuntiva. Ad esempio, Oracle Enterprise Manager Database Control utilizza circa 300 MB di RAM. Se abiliti questa opzione per una istanza database di piccole dimensioni, potresti riscontrare problemi di prestazioni dovuti ai limiti di memoria. Puoi regolare i parametri Oracle in modo che il database richieda meno RAM. In alternativa, puoi dimensionare l'istanza database aumentandone le dimensioni.

Integrazione Amazon S3

Puoi trasferire i file tra un'istanza database Amazon RDS per Oracle e un bucket Amazon S3. Puoi utilizzare l'integrazione Amazon S3 con le funzionalità Oracle Database, ad esempio Oracle Data Pump. Ad esempio, è possibile scaricare i file di Data Pump da Amazon S3 sull'istanza database RDS per Oracle. Per ulteriori informazioni, consulta Importazione di dati in Oracle in Amazon RDS.

> **Note**
>
> L'istanza database e il bucket Amazon S3 devono trovarsi nella stessa Regione AWS.

Argomenti

Configurazione delle autorizzazioni IAM per l'integrazione di RDS per Oracle con Amazon S3

Affinché RDS per Oracle si integri con Amazon S3, l'istanza database avere accesso a un bucket Amazon S3. Il Amazon VPC utilizzato dall'istanza database non deve fornire accesso agli endpoint Amazon S3.

RDS per Oracle supporta il caricamento di file da un'istanza database in un account su un bucket Amazon S3 in un account diverso. Se sono necessari ulteriori passaggi, vengono annotati nelle sezioni seguenti.

Argomenti

Fase 3: creazione di una policy IAM per il ruolo di Amazon RDS

In questo passaggio, crei una policy AWS Identity and Access Management (IAM) con le autorizzazioni necessarie per trasferire file dal bucket Amazon S3 all'istanza DB RDS. Questo passaggio presuppone che tu abbia già creato un bucket S3.

Prima di creare la policy, prendi nota delle seguenti informazioni:

- L'Amazon Resource Name (ARN) del bucket
- L'ARN per la tua AWS KMS chiave, se il bucket utilizza la crittografia SSE-KMS o SSE-S3

> **Note**
>
> Un'istanza database RDS per Oracle non può accedere ai bucket Amazon S3 crittografati con SSE-C.

Per ulteriori informazioni, consulta Protezione dei dati con la crittografia lato server nella Guida per l'utente di Amazon Simple Storage Service.

Console

Per creare una policy IAM per consentire ad Amazon RDS l'accesso a un bucket Amazon S3

1. Aprire la console di gestione IAM.
2. In Gestione accessi scegli Policy.
3. Scegliere Create Policy (Crea policy).
4. Nella scheda Visual editor (Editor visivo) selezionare Choose a service (Scegli un servizio) e quindi S3.
5. Per Operazioni, seleziona Espandi tutto, quindi sceglie le autorizzazioni del bucket e dell'oggetto necessarie per trasferire i file da un bucket Amazon S3 a Amazon RDS. Ad esempio, completa le seguenti operazioni:
   - Espandi Elenco, ListBucketquindi seleziona.
   - Espandi Leggi, quindi seleziona GetObject.
   - Espandi Scrittura, quindi seleziona PutObjecte DeleteObject.

- Espandi Gestione delle autorizzazioni, quindi seleziona PutObjectAcl. Questa autorizzazione è necessaria se si prevede di caricare file in un bucket di proprietà di un account diverso e questo account ha bisogno del pieno controllo del contenuto del bucket.

Le autorizzazioni oggetto sono autorizzazioni per operazioni sugli oggetti in Amazon S3. Devi concedere queste autorizzazioni agli oggetti presenti nel bucket e non al bucket stesso. Per ulteriori informazioni, consulta la pagina Autorizzazioni per le operazioni sugli oggetti.

6. Scegli Aggiungi regola, quindi effettua le seguenti operazioni:

   a. Scegli Specifiche.

   b. In Bucket, scegli Aggiungi ARN. Inserisci l'ARN del bucket. Il nome del bucket viene inserito automaticamente. Quindi scegliere Add (Aggiungi).

   c. Se viene visualizzata la risorsa oggetto, scegli Aggiungi ARN per aggiungere risorse manualmente o scegli Qualsiasi.

   > **Note**
   >
   > È possibile impostare Amazon Resource Name (ARN) su un valore dell'ARN più specifico in modo da consentire ad Amazon RDS di accedere solo a specifici file o cartelle presenti in un bucket Amazon S3. Per ulteriori informazioni su come definire una policy di accesso per Amazon S3, consulta Gestione delle autorizzazioni di accesso alle risorse Amazon S3.

7. (Facoltativo) Scegli Aggiungi autorizzazioni aggiuntive per aggiungere risorse alla policy. Ad esempio, completa le seguenti operazioni:

   a. Se il bucket è crittografato con una chiave KMS personalizzata, seleziona KMS per il servizio.

   b. In Operazioni manuali, seleziona quanto segue:

      - Encrypt
      - ReEncrypt da e per ReEncrypt
      - Decrypt
      - DescribeKey
      - GenerateDataKey

   c. In Risorse, scegli Specifiche.

d. In Chiave, scegli Aggiungi ARN. Specifica l'ARN della chiave personalizzata come risorsa, quindi scegli Aggiungi.

   Per ulteriori informazioni, consulta Protezione dei dati utilizzando la crittografia lato server con chiavi KMS archiviate in AWS Key Management Service (SSE-KMS) nella Guida per l'utente di Amazon Simple Storage Service.

e. Se desideri che Amazon RDS acceda ad altri bucket, aggiungi gli ARN per questi bucket. Facoltativamente, è anche possibile concedere l'accesso a tutti i bucket e gli oggetti in Amazon S3.

8. Scegliere Next: Tags (Successivo: Tag) e Next: Review (Successivo: Verifica).
9. Per Name (Nome), immettere un nome per la policy IAM, ad esempio `rds-s3-integration-policy`. Questo nome viene utilizzato quando si crea un ruolo IAM e lo si associa all'istanza database. È anche possibile aggiungere un valore Description (Descrizione) facoltativo.
10. Seleziona Create Policy (Crea policy).

AWS CLI

Crea una policy AWS Identity and Access Management (IAM) che conceda ad Amazon RDS l'accesso a un bucket Amazon S3. Dopo aver creato la policy, prendere nota del relativo ARN. L'ARN servirà in una fase successiva.

Includi le azioni appropriate nella policy in base al tipo di accesso richiesto:

- `GetObject` – È necessario trasferire i file da un bucket Amazon S3 a Amazon RDS.
- `ListBucket` – È necessario trasferire i file da un bucket Amazon S3 a Amazon RDS.
- `PutObject` – È necessario trasferire i file da un bucket Amazon RDS a un bucket Amazon S3.

Il AWS CLI comando seguente crea una policy IAM denominata *`rds-s3-integration-policy`* con queste opzioni. Concede l'accesso a un bucket denominato *`your-s3-bucket-arn`*.

Example

Per LinuxmacOS, oUnix:

```
aws iam create-policy \
   --policy-name rds-s3-integration-policy \
   --policy-document '{
```

```
    "Version": "2012-10-17",
    "Statement": [
      {
        "Sid": "s3integration",
        "Action": [
          "s3:GetObject",
          "s3:ListBucket",
          "s3:PutObject"
        ],
        "Effect": "Allow",
        "Resource": [
          "arn:aws:s3:::your-s3-bucket-arn",
          "arn:aws:s3:::your-s3-bucket-arn/*"
        ]
      }
    ]
  }'
```

Nell'esempio seguente sono incluse le autorizzazioni per le chiavi KMS personalizzate.

```
aws iam create-policy \
   --policy-name rds-s3-integration-policy \
   --policy-document '{
     "Version": "2012-10-17",
     "Statement": [
       {
         "Sid": "s3integration",
         "Action": [
           "s3:GetObject",
           "s3:ListBucket",
           "s3:PutObject",
           "kms:Decrypt",
           "kms:Encrypt",
           "kms:ReEncrypt*",
           "kms:GenerateDataKey",
           "kms:DescribeKey",
         ],
         "Effect": "Allow",
         "Resource": [
           "arn:aws:s3:::your-s3-bucket-arn",
           "arn:aws:s3:::your-s3-bucket-arn/*",
           "arn:aws:kms:::your-kms-arn"
         ]
```

```
      }
    ]
  }'
```

Per Windows:

```
aws iam create-policy ^
   --policy-name rds-s3-integration-policy ^
   --policy-document '{
     "Version": "2012-10-17",
     "Statement": [
       {
         "Sid": "s3integration",
         "Action": [
           "s3:GetObject",
           "s3:ListBucket",
           "s3:PutObject"
         ],
         "Effect": "Allow",
         "Resource": [
           "arn:aws:s3:::your-s3-bucket-arn",
           "arn:aws:s3:::your-s3-bucket-arn/*"
         ]
       }
     ]
   }'
```

Nell'esempio seguente sono incluse le autorizzazioni per le chiavi KMS personalizzate.

```
aws iam create-policy ^
   --policy-name rds-s3-integration-policy ^
   --policy-document '{
     "Version": "2012-10-17",
     "Statement": [
       {
         "Sid": "s3integration",
         "Action": [
           "s3:GetObject",
           "s3:ListBucket",
           "s3:PutObject",
           "kms:Decrypt",
           "kms:Encrypt",
           "kms:ReEncrypt",
```

```
            "kms:GenerateDataKey",
            "kms:DescribeKey",
          ],
          "Effect": "Allow",
          "Resource": [
            "arn:aws:s3:::your-s3-bucket-arn",
            "arn:aws:s3:::your-s3-bucket-arn/*",
            "arn:aws:kms:::your-kms-arn"
          ]
        }
      ]
    }'
```

Fase 2: (facoltativo) creazione di una policy IAM per il bucket Amazon S3

Questo passaggio è necessario solo nelle seguenti condizioni:

- Prevedi di caricare i file su un bucket Amazon S3 da un account (account A) e di accedervi da un altro account (account B).
- L'account B possiede il bucket.
- L'account B necessita del pieno controllo degli oggetti caricati nel bucket.

Se le condizioni precedenti non si applicano al tuo scenario, passa a Fase 3: creazione di un ruolo IAM per l'istanza database e collegamento della policy.

Per creare la policy per il bucket, assicurati di disporre di quanto segue:

- L'ID account dell'account A.
- Il nome utente dell'account A
- Il valore ARN per il bucket Amazon S3 nell'account B

Console

Per creare o modificare una policy di bucket

1. Accedi AWS Management Console e apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/.
2. Nell'elenco Buckets (Bucket) scegliere il nome del bucket per il quale si desidera creare o modificare una policy di bucket.

3. Seleziona Autorizzazioni.
4. In Bucket Policy (Policy del bucket) scegliere Edit (Modifica). Verrà visualizzata la pagina Modifica policy del bucket.
5. Nella pagina Edit bucket policy (Modifica policy del bucket) vai negli Esempi di policy della Guida per l'utente di Amazon S3 e scegli Policy generator (Generatore di policy) per generare automaticamente una policy o modificare il JSON nella sezione Policy.

   Se scegli Policy generator, il AWS Policy Generator si apre in una nuova finestra:

   a. Nella pagina Generatore di policy di AWS , all'opzione Seleziona tipo di Policy, scegli Policy del bucket S3.
   b. Aggiungi un'istruzione inserendo le informazioni nei campi forniti, quindi scegli Aggiungi istruzione. Ripetere per tutte le istruzioni che si desidera aggiungere. Per ulteriori informazioni su questi campi, consulta Riferimento agli elementi delle policy IAM JSON nella Guida per l'utente IAM.

      > **Note**
      >
      > Per comodità, la pagina Modifica policy del bucket mostra l'ARN (Amazon Resource Name) del bucket corrente sopra il campo di testo della Policy. Puoi copiare questo ARN per utilizzarlo nelle istruzioni alla pagina Generatore di policy di AWS .

   c. Dopo aver aggiunto le istruzioni, scegli Genera policy.
   d. Copia il testo della policy generata, scegli Chiudi e torna alla pagina Modifica policy del bucket nella console di Amazon S3.
6. Nella casella Policy, modificare la policy esistente o incollare la policy del bucket dal generatore di policy. Assicurati di risolvere gli avvisi di sicurezza, gli errori, gli avvisi generali e i suggerimenti prima di salvare la tua policy.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Example permissions",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::account-A-ID:account-A-user"
      },
```

```
      "Action": [
        "s3:PutObject",
        "s3:PutObjectAcl"
      ],
      "Resource": [
        "arn:aws:s3:::account-B-bucket-arn",
        "arn:aws:s3:::account-B-bucket-arn/*"
      ]
    }
  ]
}
```

7. Scegli Salva le modifiche, che ti riporterà alla pagina Autorizzazioni bucket.

Fase 3: creazione di un ruolo IAM per l'istanza database e collegamento della policy

Questo passaggio presuppone che tu abbia creato la policy IAM in Fase 3: creazione di una policy IAM per il ruolo di Amazon RDS. In questo passaggio, si crea un ruolo per l'istanza database RDS per Oracle e quindi si collega la policy al ruolo.

Console

Per creare un ruolo IAM per consentire ad Amazon RDS l'accesso a un bucket Amazon S3

1. Aprire la console di gestione IAM.
2. Nel pannello di navigazione, seleziona Roles (Ruoli).
3. Selezionare Create role (Crea ruolo).
4. Scegli un servizio in Servizio AWS .
5. Per i casi d'uso per altri AWS servizi:, scegli RDS e poi RDS — Aggiungi ruolo al database. Quindi scegli Successivo.
6. In Cerca, in Policy di autorizzazione, inserisci il nome della policy IAM creata in Fase 3: creazione di una policy IAM per il ruolo di Amazon RDS e seleziona la policy quando viene visualizzata nell'elenco. Quindi scegli Successivo.
7. In Nome ruolo, specifica un nome per il ruolo IAM, ad esempio `rds-s3-integration-role`. È anche possibile aggiungere un valore Description (Descrizione) facoltativo.
8. Scegli Crea ruolo.

AWS CLI

Per creare un ruolo e collegarvi una policy

1. Creare un ruolo IAM che Amazon RDS può assumere per conto dell'utente per accedere ai bucket Amazon S3.

   Si consiglia di utilizzare le chiavi di contesto delle condizioni globali `aws:SourceArn` e `aws:SourceAccount` nelle relazioni di trust basate sulle risorse per limitare le autorizzazioni del servizio relative a una risorsa specifica. Questo è il modo più efficace per proteggersi dal problema di deputy confused.

   Puoi usare le chiavi di contesto delle condizioni globali e avere il valore `aws:SourceArn` che contiene l'ID dell'account. In questo caso, il valore `aws:SourceAccount` e l'account nel valore `aws:SourceArn` deve utilizzare lo stesso ID account quando viene utilizzato nella stessa istruzione.

   - Utilizzare `aws:SourceArn` se si desidera un accesso cross-service per una singola risorsa.
   - Utilizzare `aws:SourceAccount` se si desidera consentire l'associazione di qualsiasi risorsa in tale account all'uso cross-service.

   Nella relazione di trust, assicurati di utilizzare la chiave di contesto della condizione globale `aws:SourceArn` con l'Amazon Resource Name (ARN) completo delle risorse che accedono al ruolo.

   Il AWS CLI comando seguente crea il ruolo denominato *`rds-s3-integration-role`* per questo scopo.

   Example

   Per LinuxmacOS, oUnix:

```
aws iam create-role \
   --role-name rds-s3-integration-role \
   --assume-role-policy-document '{
     "Version": "2012-10-17",
     "Statement": [
       {
         "Effect": "Allow",
         "Principal": {
```

```
            "Service": "rds.amazonaws.com"
          },
          "Action": "sts:AssumeRole",
          "Condition": {
              "StringEquals": {
                  "aws:SourceAccount": my_account_ID,
                  "aws:SourceArn": "arn:aws:rds:Region:my_account_ID:db:dbname"
              }
          }
        }
      ]
    }'
```

Per Windows:

```
aws iam create-role ^
   --role-name rds-s3-integration-role ^
   --assume-role-policy-document '{
     "Version": "2012-10-17",
     "Statement": [
       {
         "Effect": "Allow",
         "Principal": {
            "Service": "rds.amazonaws.com"
          },
         "Action": "sts:AssumeRole",
         "Condition": {
             "StringEquals": {
                 "aws:SourceAccount": my_account_ID,
                 "aws:SourceArn": "arn:aws:rds:Region:my_account_ID:db:dbname"
             }
         }
       }
     ]
   }'
```

Per ulteriori informazioni, consulta la pagina relativa alla creazione di un ruolo per delegare le autorizzazioni a un utente IAM nella Guida per l'utente IAM.

2. Una volta creato il ruolo, annota l'ARN del ruolo. L'ARN servirà in una fase successiva.
3. Collega la policy creata al ruolo creato.

Il AWS CLI comando seguente associa la policy al ruolo denominato*rds-s3-integration-role*.

Example

Per LinuxmacOS, oUnix:

```
aws iam attach-role-policy \
   --policy-arn your-policy-arn \
   --role-name rds-s3-integration-role
```

Per Windows:

```
aws iam attach-role-policy ^
   --policy-arn your-policy-arn ^
   --role-name rds-s3-integration-role
```

Sostituire *your-policy-arn* con l'ARN della policy annotato nel passaggio precedente.

Fase 4: associazione del ruolo IAM all'istanza database RDS per Oracle

L'ultimo passaggio per configurare le autorizzazioni per l'integrazione di Amazon S3 prevede l'associazione del ruolo IAM all'istanza database. Si notino i requisiti seguenti:

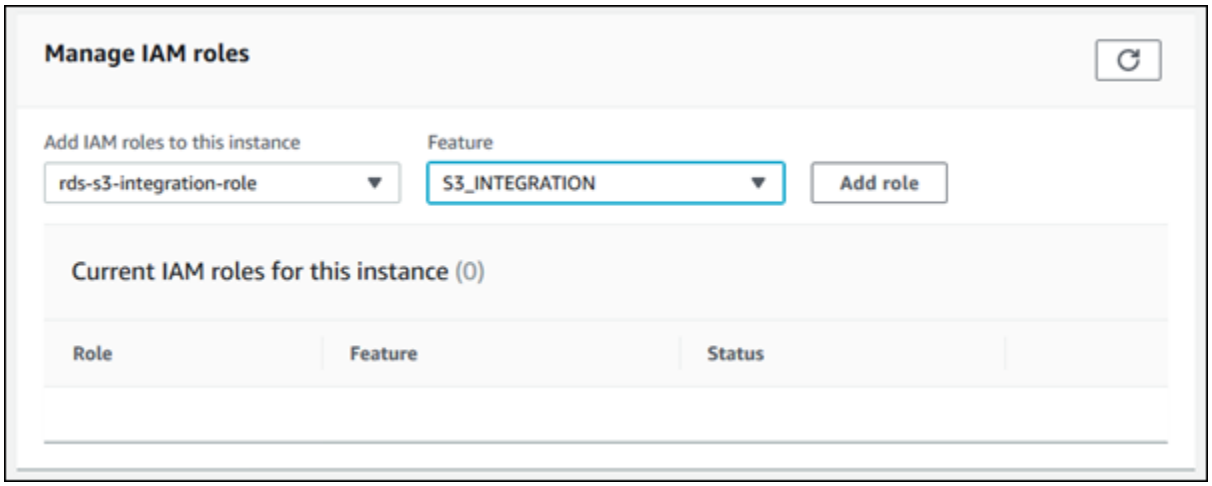
- Devi disporre dell'accesso a un ruolo a cui sono collegate le policy di autorizzazione di Amazon S3.
- È possibile associare un solo ruolo IAM alla volta all'istanza RDS per Oracle.
- Lo stato dell'istanza database deve essere Disponibile.

Console

Per associare il ruolo IAM all'istanza database RDS per Oracle

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel pannello di navigazione seleziona Database.
3. Scegli il nome dell'istanza database RDS per Oracle per visualizzarne i dettagli.

4. Sulla scheda Connettività e sicurezza, scorri verso il basso fino alla sezione Gestisci i ruoli IAM in fondo alla pagina.
5. Per Aggiungi i ruoli IAM a questa istanza, scegli il ruolo creato in Fase 3: creazione di un ruolo IAM per l'istanza database e collegamento della policy.
6. Per Feature (Caratteristica), selezionare S3_INTEGRATION.



7. Scegliere Add role (Aggiungi ruolo).

AWS CLI

Il AWS CLI comando seguente aggiunge il ruolo a un'istanza Oracle DB denominata*mydbinstance*.

Example

Per LinuxmacOS, oUnix:

```
aws rds add-role-to-db-instance \
   --db-instance-identifier mydbinstance \
   --feature-name S3_INTEGRATION \
   --role-arn your-role-arn
```

Per Windows:

```
aws rds add-role-to-db-instance ^
   --db-instance-identifier mydbinstance ^
   --feature-name S3_INTEGRATION ^
   --role-arn your-role-arn
```

Sostituire *your-role-arn* con il ruolo ARN annotato nel passaggio precedente. `S3_INTEGRATION` deve essere specificato per l'opzione `--feature-name`.

Aggiunta dell'opzione di integrazione Amazon S3

Per utilizzare l'integrazione tra Amazon RDS per Oracle e Amazon S3, l'istanza database deve essere associata a un gruppo di opzioni che include l'opzione `S3_INTEGRATION`.

Console

Per configurare un gruppo di opzioni per l'integrazione Amazon S3

1. Creare un nuovo gruppo di opzioni o identificare un gruppo opzioni esistente a cui è possibile aggiungere l'opzione `S3_INTEGRATION`.

   Per informazioni sulla creazione di un gruppo di opzioni, consulta Creazione di un gruppo di opzioni.
2. Aggiungere l'opzione `S3_INTEGRATION` al gruppo di opzioni.

   Per informazioni sull'aggiunta di un'opzione a un gruppo di opzioni, consulta Aggiunta di un'opzione a un gruppo di opzioni.
3. Crea una nuova istanza database RDS per Oracle e associarvi il gruppo opzioni oppure modificare un'istanza database RDS per Oracle per associare il gruppo opzioni a essa.

   Per informazioni sulla creazione di un'istanza database, consulta Creazione di un'istanza database Amazon RDS.

   Per ulteriori informazioni sulla modifica di un'istanza di database, consulta Modifica di un'istanza database Amazon RDS.

AWS CLI

Per configurare un gruppo di opzioni per l'integrazione Amazon S3

1. Creare un nuovo gruppo di opzioni o identificare un gruppo opzioni esistente a cui è possibile aggiungere l'opzione `S3_INTEGRATION`.

   Per informazioni sulla creazione di un gruppo di opzioni, consulta Creazione di un gruppo di opzioni.
2. Aggiungere l'opzione `S3_INTEGRATION` al gruppo di opzioni.

Ad esempio, il AWS CLI comando seguente aggiunge l'`S3_INTEGRATIONO`pzione a un gruppo di opzioni denominato**`myoptiongroup`**.

Example

Per LinuxmacOS, oUnix:

```
aws rds add-option-to-option-group \
   --option-group-name myoptiongroup \
   --options OptionName=S3_INTEGRATION,OptionVersion=1.0
```

Per Windows:

```
aws rds add-option-to-option-group ^
   --option-group-name myoptiongroup ^
   --options OptionName=S3_INTEGRATION,OptionVersion=1.0
```

3. Crea una nuova istanza database RDS per Oracle e associarvi il gruppo opzioni oppure modificare un'istanza database RDS per Oracle per associare il gruppo opzioni a essa.

   Per informazioni sulla creazione di un'istanza database, consulta Creazione di un'istanza database Amazon RDS.

   Per informazioni sulla modifica di un'istanza database RDS per Oracle, consulta Modifica di un'istanza database Amazon RDS.

Trasferimento dei file tra Amazon RDS for Oracle e un bucket Amazon S3

Puoi utilizzare il package Amazon RDS `rdsadmin_s3_tasks` per trasferire file tra un'istanza database RDS per Oracle e un bucket Amazon S3. È possibile comprimere i file con GZIP al momento del caricamento e decomprimerli al momento del download.

Argomenti

- Requisiti e limitazioni per i trasferimenti di file
- Caricamento di file da un'istanza database Oracle a un bucket Amazon S3
- Download di file da un bucket Amazon S3 a un'istanza database Oracle
- Monitoraggio dello stato di un file transfer

Requisiti e limitazioni per i trasferimenti di file

Prima di trasferire file tra l'istanza DB e un bucket Amazon S3, tieni presente quanto segue:

- Il `rdsadmin_s3_tasks` pacchetto trasferisce i file che si trovano in un'unica directory. Non è possibile includere sottodirectory in un trasferimento.
- La dimensione massima dell'oggetto in un bucket Amazon S3 è di 5 TB.
- Le attività create da vengono eseguite in modo `rdsadmin_s3_tasks` asincrono.
- È possibile caricare file dalla directory Data Pump, ad esempio`DATA_PUMP_DIR`, o da qualsiasi directory creata dall'utente. Non è possibile caricare file da una directory utilizzata dai processi in background di Oracle, ad esempio le `trace` directory `adump`o `bdump`.
- Il limite di download è di 2000 file per procedura richiesta. `download_from_s3` Se devi scaricare più di 2000 file da Amazon S3, suddividi il processo di scaricamento in operazioni distinte contenenti non più di 2000 file per chiamata.
- Se esiste un file nella cartella di scaricamento e si tenta di scaricare un file con lo stesso nome, `download_from_s3` ignora lo scaricamento del file in questione. Per rimuovere un file dalla directory di download, utilizzare la procedura PL/SQL UTL_FILE.REMOVE.

Caricamento di file da un'istanza database Oracle a un bucket Amazon S3

Per caricare i file da un'istanza database a un bucket Amazon S3, utilizza la procedura `rdsadmin.rdsadmin_s3_tasks.upload_to_s3`. Ad esempio, è possibile caricare i file di backup di Oracle Recovery Manager (RMAN) o i file di Oracle Data Pump. Per informazioni sull'utilizzo di oggetti, consulta Guida per l'utente di Amazon Simple Storage Service. Per ulteriori informazioni sull'esecuzione dei backup RMAN, consulta Esecuzione di attività RMAN comuni per le istanze database Oracle.

La procedura `rdsadmin.rdsadmin_s3_tasks.upload_to_s3` include i seguenti parametri.

Nome del parametro	Tipo di dati	Default	Campo obbligatorio	Descrizione
`p_bucket_name`	VARCHAR2	–	obbligatorio	Il nome del bucket Amazon S3 in cui caricare i file.

Nome del parametro	Tipo di dati	Default	Campo obbligatorio	Descrizione
`p_directory_name`	VARCHAR2	–	obbligatorio	Il nome dell'oggetto directory Oracle da cui caricare i file. La directory può essere un qualsiasi oggetto directory creato dall'utente o directory Data Pump, come `DATA_PUMP_DIR` . Non è possibile caricare file da una directory utilizzata dai processi in background, ad esempio, e. `adump` `bdump` `trace` **Note** È possibile solo caricare i file dalla directory specificata. NOn è possibile caricare i file nelle sottodirectory nella directory specificata.

Nome del parametro	Tipo di dati	Default	Campo obbligatorio	Descrizione
`p_s3_prefix`	VARCHAR2	–	obbligatorio	Un prefisso del nome file Amazon S3 con cui vengono caricati i file. Un prefisso vuoto carica tutti i file al livello superiore nel bucket Amazon S3 specificato e non aggiunge un prefisso ai nomi file. Ad esempio, se il prefisso è `folder_1/oradb` , i file vengono caricati su `folder_1`. In questo caso, il prefisso `oradb` viene aggiunto a ogni file.
`p_prefix`	VARCHAR2	–	obbligatorio	Un prefisso nome file con cui i nomi file devono corrispondere per essere caricati. Un prefisso vuoto carica tutti i file nella directory specificata.

Nome del parametro	Tipo di dati	Default	Campo obbligatorio	Descrizione
`p_compression_level`	NUMBER	`0`	facoltativo	Il livello di compressione GZIP. I valori validi sono compresi tra `0` e 9. • `0`: nessuna compressione • 1: compressione più veloce • 9: compressione massima
`p_bucket_owner_full_control`	VARCHAR2	–	facoltativo	L'impostazione di controllo degli accessi per il bucket. Gli unici valori validi sono null o `FULL_CONTROL` . Questa impostazione è necessaria solo se si caricano file da un account (account A) in un bucket di proprietà di un account diverso (account B) e l'account B ha bisogno del pieno controllo dei file.

Il valore restituito per la procedura `rdsadmin.rdsadmin_s3_tasks.upload_to_s3` è un ID attività.

L'esempio seguente carica tutti i file nella directory *`DATA_PUMP_DIR`* nel bucket Amazon S3 denominato *`mys3bucket`*. I file non vengono compressi.

```
SELECT rdsadmin.rdsadmin_s3_tasks.upload_to_s3(
```

```
      p_bucket_name    =>  'mys3bucket',
      p_prefix         =>  '',
      p_s3_prefix      =>  '',
      p_directory_name =>  'DATA_PUMP_DIR')
   AS TASK_ID FROM DUAL;
```

L'esempio seguente carica tutti i file con prefisso *db* nella directory *DATA_PUMP_DIR* nel bucket Amazon S3 denominato *mys3bucket*. Amazon RDS applica ai file il livello di compressione GZIP più alto.

```
SELECT rdsadmin.rdsadmin_s3_tasks.upload_to_s3(
      p_bucket_name       =>  'mys3bucket',
      p_prefix            =>  'db',
      p_s3_prefix         =>  '',
      p_directory_name    =>  'DATA_PUMP_DIR',
      p_compression_level =>  9)
   AS TASK_ID FROM DUAL;
```

L'esempio seguente carica tutti i file nella directory *DATA_PUMP_DIR* nel bucket Amazon S3 denominato *mys3bucket*. I file vengono caricati in una cartella `dbfiles`. In questo esempio, il livello di compressione GZIP è *1*, che è il livello di compressione più veloce.

```
SELECT rdsadmin.rdsadmin_s3_tasks.upload_to_s3(
      p_bucket_name       =>  'mys3bucket',
      p_prefix            =>  '',
      p_s3_prefix         =>  'dbfiles/',
      p_directory_name    =>  'DATA_PUMP_DIR',
      p_compression_level =>  1)
   AS TASK_ID FROM DUAL;
```

L'esempio seguente carica tutti i file nella directory *DATA_PUMP_DIR* nel bucket Amazon S3 denominato *mys3bucket*. I file vengono caricati in una cartella `dbfiles` e `ora` viene aggiunto all'inizio di ogni nome file. Non viene applicata alcuna compressione.

```
SELECT rdsadmin.rdsadmin_s3_tasks.upload_to_s3(
      p_bucket_name    =>  'mys3bucket',
      p_prefix         =>  '',
      p_s3_prefix      =>  'dbfiles/ora',
      p_directory_name =>  'DATA_PUMP_DIR')
   AS TASK_ID FROM DUAL;
```

L'esempio seguente presuppone che il comando sia eseguito nell'account A, ma l'account B richiede il pieno controllo del contenuto del bucket. Il comando `rdsadmin_s3_tasks.upload_to_s3` trasferisce tutti i file nella directory *`DATA_PUMP_DIR`* al bucket denominato *`s3bucket0wnedByAccountB`*. Il controllo degli accessi è impostato su `FULL_CONTROL` in modo che l'account B possa accedere ai file nel bucket. Il livello di compressione GZIP è *`6`*, ovvero il giusto compromesso tra velocità e dimensioni dei file.

```
SELECT rdsadmin.rdsadmin_s3_tasks.upload_to_s3(
      p_bucket_name               =>  's3bucket0wnedByAccountB',
      p_prefix                    =>  '',
      p_s3_prefix                 =>  '',
      p_directory_name            =>  'DATA_PUMP_DIR',
      p_bucket_owner_full_control =>  'FULL_CONTROL',
      p_compression_level         =>  6)
   AS TASK_ID FROM DUAL;
```

In ogni esempio, l'istruzione `SELECT` restituisce l'ID dell'attività in un tipo di dati `VARCHAR2`.

È possibile visualizzare il risultato visualizzando il file di output dell'attività.

```
SELECT text FROM table(rdsadmin.rds_file_util.read_text_file('BDUMP','dbtask-task-id.log'));
```

Sostituire *`task-id`* con l'ID attività restituito dalla procedura.

> **Note**
>
> Le attività vengono eseguite in modo asincrono.

Download di file da un bucket Amazon S3 a un'istanza database Oracle

Per scaricare i file da un bucket Amazon S3 a un'istanza database RDS per Oracle, utilizza la procedura Amazon RDS `rdsadmin.rdsadmin_s3_tasks.download_from_s3`.

La procedura `download_from_s3` include i seguenti parametri.

Nome del parametro	Tipo di dati	Default	Campo obbligatorio	Descrizione
p_bucket_name	VARCHAI	–	Richiesto	Il nome del bucket Amazon S3 da cui scaricare i file.
p_directory_name	VARCHAI	–	Richiesto	Il nome dell'oggetto directory Oracle su cui scaricare i file. La directory può essere un qualsiasi oggetto directory creato dall'utente o directory Data Pump, come DATA_PUMP_DIR .
p_error_on_zero_downloads	VARCHAI	FALSE	Facoltativo	Un flag che determina se l'attività genera un errore quando nessun oggetto nel bucket Amazon S3 corrisponde al prefisso. Se questo parametro non è impostato o è impostato su FALSE (impostazione predefinita), l'attività stampa un messaggio che indica che non è stato trovato alcun oggetto, ma non genera un'eccezione o un errore. Se questo parametro è TRUE, l'attività genera un'eccezione e non riesce. Esempi di specifiche dei prefissi che possono non riuscire nei test di corrispondenza sono gli spazi nei prefissi, come in ' import/test9.log' , e le mancate corrispondenze tra maiuscole e minuscole, come in test9.log e test9.LOG .
p_s3_prefix	VARCHAI	–	Richiesto	Un prefisso nome file con cui i nomi file devono corrispondere per essere scaricati. Un prefisso vuoto scarica

Nome del parametro	Tipo di dati	Default	Campo obbligatorio	Descrizione
				tutti i file di livello superiore nel bucket Amazon S3 specificato, ma non i file nelle cartelle nel bucket. La procedura scarica gli oggetti Amazon S3 solo dalla cartella di primo livello che corrisponde al prefisso. Le strutture directory nidificate corrispondenti al prefisso specificato non vengono scaricate. Ad esempio, si supponga che un bucket Amazon S3 abbia la struttura della cartella `folder_1/folder_2/folder_3` . Specifica il prefisso `'folder_1/folder_2/'` . In questo caso, vengono scaricati solo i file in `folder_2`, non i file in `folder_1` o `folder_3`. Se, invece, viene specificato il prefisso `'folder_1/folder_2'` , tutti i file in `folder_1` che corrispondono al prefisso `'folder_2'` vengono scaricati, mentre i file in `folder_2` non vengono scaricati.
`p_decompression_format`	VARCHAI	–	Facoltativo	Il formato di decompressione. I valori validi sono `NONE` per non eseguire la decompressione e `GZIP` per eseguire la decompressione.

Il valore restituito per la procedura `rdsadmin.rdsadmin_s3_tasks.download_from_s3` è un ID attività.

L'esempio seguente scarica tutti i file nel bucket Amazon S3 denominato *`mys3bucket`* nella directory *`DATA_PUMP_DIR`*. I file non sono compressi, quindi non viene applicata alcuna decompressione.

```
SELECT rdsadmin.rdsadmin_s3_tasks.download_from_s3(
      p_bucket_name    =>  'mys3bucket',
      p_directory_name =>  'DATA_PUMP_DIR')
   AS TASK_ID FROM DUAL;
```

L'esempio seguente scarica tutti i file con prefisso *`db`* nel bucket Amazon S3 denominato *`mys3bucket`* nella directory *`DATA_PUMP_DIR`*. I file non sono compressi, quindi non viene applicata alcuna decompressione. Il parametro `p_error_on_zero_downloads` attiva il controllo degli errori dei prefissi, quindi se il prefisso non corrisponde a nessun file nel bucket, l'attività genera un'eccezione e non riesce.

```
SELECT rdsadmin.rdsadmin_s3_tasks.download_from_s3(
      p_bucket_name               =>  'mys3bucket',
      p_s3_prefix                 =>  'db',
      p_directory_name            =>  'DATA_PUMP_DIR',
      p_decompression_format      =>  'GZIP',
      p_error_on_zero_downloads   =>  'TRUE')
   AS TASK_ID FROM DUAL;
```

L'esempio seguente scarica tutti i file nella cartella *`myfolder/`* nel bucket Amazon S3 denominato *`mys3bucket`* nella directory *`DATA_PUMP_DIR`*. Utilizzare il parametro `p_s3_prefix` per specificare la cartella Amazon S3. I file caricati vengono compressi con GZIP, ma non vengono decompressi durante il download.

```
SELECT rdsadmin.rdsadmin_s3_tasks.download_from_s3(
      p_bucket_name          =>  'mys3bucket',
      p_s3_prefix            =>  'myfolder/',
      p_directory_name       =>  'DATA_PUMP_DIR',
      p_decompression_format =>  'NONE')
   AS TASK_ID FROM DUAL;
```

L'esempio seguente scarica il file *`mydumpfile.dmp`* nel bucket Amazon S3 denominato *`mys3bucket`* nella directory *`DATA_PUMP_DIR`*. Non viene applicata alcuna decompressione.

```
SELECT rdsadmin.rdsadmin_s3_tasks.download_from_s3(
      p_bucket_name    =>  'mys3bucket',
      p_s3_prefix      =>  'mydumpfile.dmp',
      p_directory_name =>  'DATA_PUMP_DIR')
   AS TASK_ID FROM DUAL;
```

In ogni esempio, l'istruzione SELECT restituisce l'ID dell'attività in un tipo di dati VARCHAR2.

È possibile visualizzare il risultato visualizzando il file di output dell'attività.

```
SELECT text FROM table(rdsadmin.rds_file_util.read_text_file('BDUMP','dbtask-task-id.log'));
```

Sostituire *task-id* con l'ID attività restituito dalla procedura.

> **Note**
>
> Le attività vengono eseguite in modo asincrono.
> Per rimuovere i file da una directory, è possibile utilizzare la procedura Oracle UTL_FILE.FREMOVE. Per ulteriori informazioni, consulta la sezione relativa alla Procedura FREMOVE nella documentazione di Oracle.

Monitoraggio dello stato di un file transfer

Le attività di file transfer pubblicano gli eventi Amazon RDS al loro inizio e al completamento. Il messaggio dell'evento contiene l'ID dell'attività per il trasferimento file. Per informazioni sulla visualizzazione degli eventi, consultare Visualizzazione di eventi Amazon RDS.

È possibile visualizzare lo stato di un'attività in corso in un file bdump. I file bdump si trovano nella directory /rdsdbdata/log/trace. Il nome del file bdump ha il formato che segue.

```
dbtask-task-id.log
```

Sostituire *task-id* con l'ID dell'attività da monitorare.

> **Note**
>
> Le attività vengono eseguite in modo asincrono.

Puoi usare la procedura memorizzata in `rdsadmin.rds_file_util.read_text_file` per visualizzare i contenuti dei file bdump. Ad esempio, la query seguente restituisce i contenuti del file bdump *dbtask-1234567890123-1234.log*.

```
SELECT text FROM
 table(rdsadmin.rds_file_util.read_text_file('BDUMP','dbtask-1234567890123-1234.log'));
```

L'esempio seguente mostra il file di log di un trasferimento non riuscito.

```
TASK_ID

--------------------------------------------------------------------------------------------------
1234567890123-1234

TEXT

--------------------------------------------------------------------------------------------------
2023-04-17 18:21:33.993 UTC [INFO ] File #1: Uploading the file /rdsdbdata/datapump/
A123B4CDEF567890G1234567890H1234/sample.dmp to Amazon S3 with bucket name mys3bucket
 and key sample.dmp.
2023-04-17 18:21:34.188 UTC [ERROR] RDS doesn't have permission to write to Amazon S3
 bucket name mys3bucket and key sample.dmp.
2023-04-17 18:21:34.189 UTC [INFO ] The task failed.
```

Risoluzione dei problemi di integrazione Amazon S3

Per suggerimenti sulla risoluzione dei problemi, consulta l'articolo AWS Re:post Come si risolvono i problemi quando integro Amazon RDS for Oracle con Amazon S3? .

Rimozione dell'opzione di integrazione Amazon S3

Puoi rimuovere l'opzione di integrazione Amazon S3 da un'istanza database.

Per rimuovere l'opzione di integrazione Amazon S3 da un'istanza database, procedi in uno dei seguenti modi:

- Per rimuovere l'opzione di integrazione Amazon S3 da più istanze database, rimuovere l'opzione `S3_INTEGRATION` dal gruppo di opzioni a cui appartengono le istanze database. Questa modifica coinvolge tutte le istanze database che usano il gruppo di opzioni. Per ulteriori informazioni, consulta Rimozione di un'opzione da un gruppo di opzioni.

- Per rimuovere l'opzione di integrazione Amazon S3 da una singola istanza database, modifica l'istanza database e specifica un gruppo di opzioni diverso che non comprenda l'opzione `S3_INTEGRATION`. È possibile specificare il gruppo di opzioni predefinito (vuoto) o un gruppo di opzioni personalizzato diverso. Per ulteriori informazioni, consulta Modifica di un'istanza database Amazon RDS.

Oracle Application Express (APEX)

Amazon RDS supporta Oracle Application Express (APEX) tramite l'utilizzo delle opzioni `APEX` e `APEX-DEV`. Puoi implementare Oracle APEX come ambiente di runtime o come ambiente di sviluppo completo per le applicazioni basate sul Web. Utilizzando Oracle APEX, puoi compilare le applicazioni interamente all'interno del browser Web. Per ulteriori informazioni, consulta Oracle Application Express nella documentazione di Oracle.

Argomenti

Componenti APEX

Oracle APEX è costituito dai seguenti componenti principali:

- Un repository che archivia i metadati per i componenti e le applicazioni APEX. Il repository è formato da tabelle, indici e altri oggetti che sono installati nella tua istanza database Amazon RDS.
- Un listener che gestisce le comunicazioni HTTP con i client Oracle APEX. L'ascoltatore risiede in un host separato, come un'istanza Amazon EC2, un server on-premise della tua azienda oppure il tuo computer desktop. Il listener accetta le connessioni in entrata dai browser Web, le inoltra all'istanza database Amazon RDS per l'elaborazione e quindi invia nuovamente i risultati dal repository ai browser. RDS per Oracle supporta i seguenti tipi di ascoltatori:

- Per APEX versione 5.0 e successive, utilizzare Oracle REST Data Services (ORDS) versione 19.1 e successive. Si consiglia di utilizzare la versione più recente supportata di Oracle APEX e ORDS. Questa documentazione descrive le versioni precedenti solo per la compatibilità con le versioni precedenti.
- Per APEX versione 4.1.1, è possibile utilizzare Oracle APEX Listener versione 1.1.4.
- È possibile utilizzare Oracle HTTP Server e i listener `mod_plsql`.

> **Note**
>
> Amazon RDS non supporta il server HTTP Oracle XML DB con il gateway PL/SQL integrato, pertanto non puoi utilizzarlo come listener per APEX. In generale, Oracle consiglia di utilizzare il gateway PL/SQL integrato per le applicazioni eseguite su Internet.

Per ulteriori informazioni su questi tipi di listener, consulta About Choosing a Web Listener nella documentazione di Oracle.

Quando vengono aggiunte le opzioni APEX Amazon RDS all'istanza database RDS per Oracle, Amazon RDS installa solo il repository Oracle APEX. Installa l'ascoltatore su un host separato.

Requisiti di versione APEX

L'opzione APEX utilizza lo storage nella classe di istanza database per la tua istanza database. Di seguito si riportano le versioni supportate e i requisiti di storage approssimativi per Oracle APEX.

Versione APEX	Requisiti di storage	Versioni di Oracle Database supportate	Note
Oracle APEX versione 23.2.v1	110 MiB	19c e versioni successive	Questa versione include la patch 35895964: PACCHETTO PSE PER APEX 23.2 (PSES IN AGGIUNTA ALLA 23.2.0), PATCH_VERSION 6.

Versione APEX	Requisiti di storage	Versioni di Oracle Database supportate	Note
Oracle APEX versione 23.1.v1	106 MiB	19c e versioni successive	Questa versione include la patch 35283657: PSE BUNDLE FOR APEX 23.1 (PSES ON TOP OF 23.1.0), PATCH_VERSION 2.
Oracle APEX versione 22.2.v1	106 MiB	Tutti	Questa versione include la patch 34628174: PSE BUNDLE FOR APEX 22.2 (PSES SU 22.2.0), PATCH_VERSION 4.
Oracle APEX versione 22.1.v1	124 MiB	Tutti	Questa versione include la patch 34020981: PSE BUNDLE FOR APEX 22.1 (PSES SU 22.1.0), PATCH_VERSION 6.
Oracle APEX versione 21.2.v1	125 MiB	Tutti	Questa versione include la patch 33420059: PSE BUNDLE FOR APEX 21.2 (PSES SU 21.2.0), PATCH_VERSION 8.
Oracle APEX versione 21.1.v1	125 MiB	Tutti	Questa versione include la patch 32598392: PSE BUNDLE FOR APEX 21.1, PATCH_VERSION 3.
Oracle APEX versione 20.2.v1	148 MiB	Tutti tranne 21c	Questa versione include la patch 32006852: PSE BUNDLE FOR APEX 20.2, PATCH_VERSION 2020.11.12. È possibile visualizzare il numero e la data della patch eseguendo la seguente query: `SELECT PATCH_VERSION, PATCH_NUMBER FROM APEX_PATCHES;`
Oracle APEX versione 20.1.v1	173 MiB	Tutti tranne 21c	Questa versione include la patch 30990551: PSE BUNDLE FOR APEX 20.1, PATCH_VERSION 2020.07.15.

Versione APEX	Requisiti di storage	Versioni di Oracle Database supportate	Note
Oracle APEX versione 19.2.v1	149 MiB	Tutti tranne 21c	
Oracle APEX versione 19.1.v1	148 MiB	Tutti tranne 21c	
Oracle APEX versione 18.2.v1	146 MiB	Solo 12.1 e 12.2	
Oracle APEX versione 18.1.v1	145 MiB	Solo 12.1 e 12.2	
Oracle APEX versione 5.1.4.v1	220 MiB	Solo 12.1 e 12.2	
Oracle APEX versione 5.1.2.v1	150 MiB	Solo 12.1 e 12.2	
Oracle APEX versione 5.0.4.v1	140 MiB	Solo 12.1 e 12.2	
Oracle APEX versione 4.2.6.v1	160 MiB	Solo 12.1	

Per i file .zip APEX scaricabili, consulta Oracle APEX Prior Release Archives (Archivi delle versioni precedenti di Oracle APEX) sul sito Web di Oracle.

Requisiti e limitazioni di Oracle APEX e ORDS

Tieni presente i seguenti requisiti di APEX e ORDS:

- Devi utilizzare l'ambiente di runtime Java (JRE).
- L'installazione client Oracle deve includere quanto segue:
  - SQL*Plus o SQL Developer per le attività di amministrazione

- Oracle Net Services per la configurazione delle connessioni all'istanza database RDS per Oracle

Nota le seguenti limitazioni per APEX e ORDS:

- Non è possibile utilizzare un RDS per Oracle CDB con ORDS 22 e versioni successive. Come soluzione alternativa, puoi invece utilizzare una versione precedente di ORDS o utilizzare un database non CDB Oracle Database 19c.

Aggiunta delle opzioni APEX e APEX-DEV

Per aggiungere le opzioni `APEX` e `APEX-DEV` a un'istanza database, esegui le operazioni seguenti:

1. Creare un nuovo gruppo di opzioni oppure copiare o modificare un gruppo di opzioni esistente.
2. Aggiungi le opzioni `APEX` e `APEX-DEV` al gruppo di opzioni.
3. Associare il gruppo di opzioni a questa istanza database.

Quando aggiungi le opzioni APEX Amazon RDS, si verifica una breve interruzione mentre l'istanza database viene riavviata automaticamente.

> **Note**
>
> `APEX_MAIL` è disponibile se l'opzione `APEX` è installata. Il privilegio di esecuzione per il pacchetto `APEX_MAIL` è concesso a `PUBLIC` e quindi non è necessario l'account amministrativo APEX per utilizzarlo.

Per aggiungere le opzioni APEX a un'istanza database

1. Determinare il gruppo di opzioni che si desidera usare. È possibile creare un nuovo gruppo di opzioni oppure usare un gruppo di opzioni esistente. Se si desidera usare un gruppo di opzioni esistente, puoi passare alla fase successiva. In caso contrario, creare un gruppo di opzioni database personalizzato con le seguenti impostazioni:

   a. Per Engine (Motore), scegliere l'edizione di Oracle da utilizzare. Le opzioni APEX sono supportate in tutte le edizioni.
   b. In Major engine version (Versione principale del motore), scegliere la versione dell'istanza database.

Per ulteriori informazioni, consulta Creazione di un gruppo di opzioni.

2. Aggiungere le opzioni al gruppo di opzioni. Per distribuire solo l'ambiente di runtime Oracle APEX aggiungere solo l'opzione `APEX`. Per distribuire l'ambiente di sviluppo completo aggiungere entrambe le opzioni `APEX` e `APEX-DEV`. Per Oracle Database 12c aggiungere le opzioni APEX e APEX-DEV.

   In Version (Versione) scegliere la versione di `APEX` da utilizzare. Se non si sceglie una versione, la versione 4.2.6.v1 sarà quella predefinita per Oracle Database 12c.

   > **Important**
   >
   > Se aggiungi le opzioni APEX a un gruppo di opzioni esistente già associato a una o più istanze database, si verifica una breve interruzione. Durante questa interruzione, tutte le istanze database vengono riavviate automaticamente.

   Per ulteriori informazioni sull'aggiunta di opzioni, consulta Aggiunta di un'opzione a un gruppo di opzioni.

3. Applicare il gruppo di opzioni a un'istanza database nuova o esistente:

   - Per una nuova istanza database, si applica il gruppo di opzioni quando viene avviata l'istanza. Per ulteriori informazioni, consulta Creazione di un'istanza database Amazon RDS.
   - Per un'istanza database esistente, viene applicato il gruppo di opzioni modificando l'istanza e collegando il nuovo gruppo di opzioni. Quando aggiungi le opzioni APEX a un'istanza database esistente, si verifica una breve interruzione mentre l'istanza database viene riavviata automaticamente. Per ulteriori informazioni, consulta Modifica di un'istanza database Amazon RDS.

Sblocco dell'account utente pubblico

Dopo aver installato le opzioni APEX Amazon RDS, assicurati di eseguire le operazioni seguenti:

1. Modificare la password per l'account dell'utente pubblico APEX.
2. Sbloccare l'account.

Per farlo, utilizza l'utilità a riga di comando Oracle SQL*Plus. Connettiti alla tua istanza database come utente master e utilizza i seguenti comandi. Sostituisci `new_password` con una password a tua scelta.

```
ALTER USER APEX_PUBLIC_USER IDENTIFIED BY new_password;
ALTER USER APEX_PUBLIC_USER ACCOUNT UNLOCK;
```

Configurazione dei servizi RESTful per Oracle APEX

Per configurare i servizi RESTful in APEX (non necessario per APEX 4.1.1.V1), utilizza SQL*Plus per connetterti alla tua istanza database come utente master Successivamente, esegui la stored procedure `rdsadmin.rdsadmin_run_apex_rest_config`. Quando esegui la stored procedure, fornisci le password per i seguenti utenti:

- `APEX_LISTENER`
- `APEX_REST_PUBLIC_USER`

La stored procedure esegue lo script `apex_rest_config.sql`, che crea nuovi account database per questi utenti.

> **Note**
>
> Oracle APEX versione 4.1.1.v1 non necessita configurazione. Per questa versione di Oracle APEX non devi eseguire la procedura archiviata.

Il comando seguente avvia la procedura archiviata.

```
EXEC rdsadmin.rdsadmin_run_apex_rest_config('apex_listener_password',
 'apex_rest_public_user_password');
```

Preparazione all'installazione di ORDS

Prima di installare ORDS, è necessario creare un utente del sistema operativo senza privilegi, quindi scaricare e decomprimere il file di installazione APEX.

Per preparare l'installazione di ORDS

1. Accedere a `myapexhost.example.com` come `root`.

2. Creare un utente del sistema operativo senza privilegi che possieda l'installazione del listener. Il comando seguente crea un nuovo utente denominato apexuser.

```
useradd -d /home/apexuser apexuser
```

Il seguente comando assegna una password al nuovo utente.

```
passwd apexuser;
```

3. Accedi a `myapexhost.example.com` come `apexuser` e scarica il file di installazione APEX da Oracle e salvalo nella directory `/home/apexuser`:

- http://www.oracle.com/technetwork/developer-tools/apex/downloads/index.html
- Oracle Application Express Prior Release Archives

4. Decomprimi il file nella directory `/home/apexuser`.

```
unzip apex_version.zip
```

Dopo aver decompresso il file, troverai una directory `apex` nella directory `/home/apexuser`.

5. Quando sei ancora collegato a `myapexhost.example.com` come `apexuser`, scarica il file Oracle REST Data Services da Oracle nella directory `/home/apexuser`: http://www.oracle.com/technetwork/developer-tools/apex-listener/downloads/index.html

Installazione e configurazione di ORDS 2.1 e versioni precedenti

È ora possibile installare e configurare Oracle Rest Data Services (ORDS) per l'utilizzo con Oracle APEX. Per la versione 5.0 e successive di APEX, utilizza le versioni da 19.1 a 21 di ORDS.
Per informazioni su come installare ORDS 22 e versioni successive, consulta. Installazione e configurazione di ORDS 2.2 e versioni successive

Installa il listener in un host separato, come un'istanza Amazon EC2, un server locale nella tua azienda oppure il tuo computer desktop. Per gli esempi in questa sezione, supponiamo che il nome dell'host sia `myapexhost.example.com` e che l'host esegua Linux.

Per installare e configurare ORDS 2.1 e versioni precedenti per l'utilizzo con Oracle APEX

1. Vai ai servizi dati Oracle REST ed esamina il Readme. Assicurati di avere installata la versione richiesta di Java.

2. Crea una nuova directory per l'installazione di ORDS.

```
mkdir /home/apexuser/ORDS
cd /home/apexuser/ORDS
```

3. Scarica il file `ords.version.number.zip` da Oracle REST Data Services.
4. Decomprimere il file nella directory `/home/apexuser/ORDS`.
5. Se stai installando ORDS in un database multilocazione, aggiunge la riga seguente al file `/home/apexuser/ORDS/params/ords_params.properties`:

```
pdb.disable.lockdown=false
```

6. Concedere all'utente master i privilegi necessari per installare ORDS.

   Una volta installata l'opzione APEX Amazon RDS, concedere all'utente master i privilegi necessari per installare lo schema ORDS. Questa operazione può essere eseguita collegandosi al database ed eseguendo questi comandi: Sostituisci *MASTER_USER* con il nome in maiuscolo dell'utente master.

   > **Important**
   >
   > Quando si immette il nome utente, utilizzare maiuscole a meno che l'utente non sia stato creato con un identificatore con distinzione tra maiuscole e minuscole. Ad esempio, se esegui `CREATE USER myuser` o `CREATE USER MYUSER`, il dizionario dati memorizza `MYUSER`. Tuttavia, se si utilizzano virgolette doppie in `CREATE USER "MyUser"`, il dizionario dati memorizza `MyUser`. Per ulteriori informazioni, consulta Concedere privilegi SELECT o EXECUTE agli oggetti SYS.

```
exec rdsadmin.rdsadmin_util.grant_sys_object('DBA_OBJECTS', 'MASTER_USER',
 'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('DBA_ROLE_PRIVS', 'MASTER_USER',
 'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('DBA_TAB_COLUMNS', 'MASTER_USER',
 'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('USER_CONS_COLUMNS', 'MASTER_USER',
 'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('USER_CONSTRAINTS', 'MASTER_USER',
 'SELECT', true);
```

```
exec rdsadmin.rdsadmin_util.grant_sys_object('USER_OBJECTS', 'MASTER_USER',
 'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('USER_PROCEDURES', 'MASTER_USER',
 'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('USER_TAB_COLUMNS', 'MASTER_USER',
 'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('USER_TABLES', 'MASTER_USER',
 'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('USER_VIEWS', 'MASTER_USER', 'SELECT',
 true);
exec rdsadmin.rdsadmin_util.grant_sys_object('WPIUTL', 'MASTER_USER', 'EXECUTE',
 true);
exec rdsadmin.rdsadmin_util.grant_sys_object('DBMS_SESSION', 'MASTER_USER',
 'EXECUTE', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('DBMS_UTILITY', 'MASTER_USER',
 'EXECUTE', true);
```

> **Note**
>
> Questi comandi si applicano a ORDS versione 19.1 e successive.

7. Installare lo schema ORDS utilizzando il file scaricato ords.war.

```
java -jar ords.war install advanced
```

Il programma richiede le seguenti informazioni. I valori predefiniti sono riportati tra parentesi. Per altre informazioni, consulta l'articolo relativo a Introduzione a Oracle REST Data Services nella documentazione Oracle.

- Destinazione di archiviazione dei dati di configurazione

  Inserisci */home/apexuser/ORDS*. Questa è la posizione dei file di configurazione di ORDS.
- Specificare il tipo di connessione al database da utilizzare. Immettere il numero per [1] Basic [2] TNS [3] URL personalizzato [1]:

  Scegliere il tipo di connessione desiderato.
- Nome del server del database [localhost]: *DB_instance_endpoint*

  Scegliere il valore predefinito o digitare il valore corretto.
- Immettere la porta del listener del database [1521]: *DB_Instance_port*

Scegliere il valore predefinito o digitare il valore corretto.

- Immettere 1 per specificare il nome del servizio di database o 2 per specificare il SID del database [1]:

  Scegliere 2 per specificare il SID del database.

- SID del database [xe]

  Scegliere il valore predefinito o digitare il valore corretto.

- Immettere 1 se si desidera verificare/installare lo schema di Oracle REST Data Services o 2 per ignorare questo passaggio [1]:

  Scegliere 1. Questo passaggio crea l'utente proxy di Oracle REST Data Services denominato ORDS_PUBLIC_USER.

- Immettere la password del database per ORDS_PUBLIC_USER:

  Immettere la password, quindi confermarla.

- Richiede l'accesso con i privilegi di amministratore per verificare lo schema Oracle REST Data Services.

  Inserire il nome utente dell'amministratore: *`master_user`*

  Inserire la password del database per *`master_user`*: *`master_user_password`*

  Confermare la password: *`master_user_password`*

  > **Note**
  >
  > Specifica una password diversa dal prompt mostrato qui come best practice per la sicurezza.

- Inserire lo spazio tabella predefinito per ORDS_METADATA [SYSAUX].

  Inserire lo spazio tabella temporaneo per ORDS_METADATA [TEMP].

  Inserire lo spazio tabella predefinito per ORDS_PUBLIC_USER [USERS].

  Inserire lo spazio tabella temporaneo per ORDS_PUBLIC_USER [USERS].

- Inserire 1 per utilizzare PL/SQL Gateway o 2 per saltare questo passaggio. Se si sta utilizzando Oracle Application Express o si sta migrando da mod_plsql, inserire 1 [1].

  Scegliere il valore predefinito.

- Inserire il nome utente del database PL/SQL Gateway [APEX_PUBLIC_USER]

  Scegliere il valore predefinito.

- Inserire la password del database per APEX_PUBLIC_USER

  Immettere la password, quindi confermarla.

- Inserire 1 per specificare le password per gli utenti dei database Application Express RESTful Services (APEX_LISTENER, APEX_REST_PUBLIC_USER) o 2 per saltare questo passaggio [1]

  Scegliere 2 per APEX 4.1.1.V1 oppure scegliere 1 per tutte le altre versioni di APEX.

- [Non necessario per APEX 4.1.1.v1] Password di database per APEX_LISTENER

  Immettere la password (se necessario), quindi confermarla.

- [Non necessario per APEX 4.1.1.v1] Password di database per APEX_REST_PUBLIC_USER

  Immettere la password (se necessario), quindi confermarla.

- Immettere un numero per selezionare una funzione da abilitare:

  Immettere 1 per abilitare tutte le funzioni: SQL Developer Web, REST Enabled SQL e Database API.

- Immettere 1 se si desidera avviare in modalità standalone o 2 per uscire [1]:

  Specificare (sì 1.

- Immettere l'ubicazione delle risorse statiche APEX:

  Se i file di installazione APEX sono stati decompressi in `/home/apexuser`, immettere `/home/apexuser/apex/images`. In caso contrario, immettere *`unzip_path`*`/apex/images`, dove *`unzip_path`* è la directory in cui è stato decompresso il file.

- Immettere 1 se si utilizza HTTP o 2 se si utilizza HTTPS [1]:

  Se si immette 1, specificare la porta HTTP. Se si immette 2, specificare la porta HTTPS e il nome host SSL. L'opzione HTTPS richiede di specificare come si fornirà il certificato:

  - Immettere 1 per utilizzare il certificato autofirmato.

- Immettere 2 per fornire il proprio certificato. Se si immette 2, specificare il percorso per il certificato SSL e il percorso per la chiave privata del certificato SSL.

8. Impostare una password per l'utente `admin` APEX. Per farlo, utilizzare SQL*Plus per connettersi alla propria istanza database come utente master, quindi utilizzare i seguenti comandi:

```
EXEC rdsadmin.rdsadmin_util.grant_apex_admin_role;
grant APEX_ADMINISTRATOR_ROLE to master;
@/home/apexuser/apex/apxchpwd.sql
```

   Sostituire *`master`* con il proprio nome utente master. Quando lo script `apxchpwd.sql` lo richiede, inserire una nuova password `admin`.

9. Avviare il listener ORDS. Eseguire il seguente codice.

```
java -jar ords.war
```

   Al primo avvio dell'ORDS viene richiesto di fornire la posizione delle risorse statiche APEX. Questa cartella di immagini è posizionata nella directory `/apex/images`all'interno della directory di installazione per APEX.

10. Tornare alla finestra di amministrazione APEX nel proprio browser e scegliere l'opzione Administration (Amministrazione). Scegliere quindi Application Express Internal Administration (Amministrazione interna Application Express). Quando vengono richieste le credenziali, inserire le seguenti informazioni:

    - User name (Nome utente – `admin`
    - Password – La password impostata usando lo script `apxchpwd.sql`

    Scegliere Login (Accedi), quindi impostare una nuova password per l'utente `admin`.

Il listener è ora pronto per essere utilizzato.

Installazione e configurazione di ORDS 2.2 e versioni successive

È ora possibile installare e configurare Oracle Rest Data Services (ORDS) per l'utilizzo con Oracle APEX. Le istruzioni per ORDS 22 differiscono dalle istruzioni per le versioni precedenti.

Per installare e configurare ORDS 2.2 e versioni successive da utilizzare con Oracle APEX

1. Vai ai servizi dati Oracle REST ed esamina il Readme per la versione ORDS che intendi scaricare. Assicurati di avere installata la versione richiesta di Java.
2. Crea una nuova directory per l'installazione di ORDS.

```
mkdir /home/apexuser/ORDS
cd /home/apexuser/ORDS
```

3. Scarica il file `ords.version.number.zip` o `ords-latest.zip` dai servizi dati Oracle REST.
4. Decomprimere il file nella directory `/home/apexuser/ORDS`.
5. Concedere all'utente master i privilegi necessari per installare ORDS.

   Una volta installata l'opzione APEX Amazon RDS, concedere all'utente master i privilegi necessari per installare lo schema ORDS. È possibile farlo accedendo al database ed eseguendo i seguenti comandi. Sostituisci *MASTER_USER* con il nome in maiuscolo dell'utente master.

   > **Important**
   >
   > Quando si immette il nome utente, utilizzare maiuscole a meno che l'utente non sia stato creato con un identificatore con distinzione tra maiuscole e minuscole. Ad esempio, se esegui `CREATE USER myuser` o `CREATE USER MYUSER`, il dizionario dati memorizza `MYUSER`. Tuttavia, se si utilizzano virgolette doppie in `CREATE USER "MyUser"`, il dizionario dati memorizza `MyUser`. Per ulteriori informazioni, consulta Concedere privilegi SELECT o EXECUTE agli oggetti SYS.

```
exec rdsadmin.rdsadmin_util.grant_sys_object('DBA_OBJECTS', 'MASTER_USER',
 'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('DBA_ROLE_PRIVS', 'MASTER_USER',
 'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('DBA_TAB_COLUMNS', 'MASTER_USER',
 'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('USER_CONS_COLUMNS', 'MASTER_USER',
 'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('USER_CONSTRAINTS', 'MASTER_USER',
 'SELECT', true);
```

```
exec rdsadmin.rdsadmin_util.grant_sys_object('USER_OBJECTS', 'MASTER_USER',
 'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('USER_PROCEDURES', 'MASTER_USER',
 'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('USER_TAB_COLUMNS', 'MASTER_USER',
 'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('USER_TABLES', 'MASTER_USER',
 'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('USER_VIEWS', 'MASTER_USER', 'SELECT',
 true);
exec rdsadmin.rdsadmin_util.grant_sys_object('WPIUTL', 'MASTER_USER', 'EXECUTE',
 true);
exec rdsadmin.rdsadmin_util.grant_sys_object('DBMS_SESSION', 'MASTER_USER',
 'EXECUTE', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('DBMS_UTILITY', 'MASTER_USER',
 'EXECUTE', true);

exec rdsadmin.rdsadmin_util.grant_sys_object('DBMS_LOB', 'MASTER_USER', 'EXECUTE',
 true);
exec rdsadmin.rdsadmin_util.grant_sys_object('DBMS_ASSERT', 'MASTER_USER',
 'EXECUTE', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('DBMS_OUTPUT', 'MASTER_USER',
 'EXECUTE', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('DBMS_SCHEDULER', 'MASTER_USER',
 'EXECUTE', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('HTP', 'MASTER_USER', 'EXECUTE',
 true);
exec rdsadmin.rdsadmin_util.grant_sys_object('OWA', 'MASTER_USER', 'EXECUTE',
 true);
exec rdsadmin.rdsadmin_util.grant_sys_object('WPG_DOCLOAD', 'MASTER_USER',
 'EXECUTE', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('DBMS_CRYPTO', 'MASTER_USER',
 'EXECUTE', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('DBMS_METADATA', 'MASTER_USER',
 'EXECUTE', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('DBMS_SQL', 'MASTER_USER', 'EXECUTE',
 true);
exec rdsadmin.rdsadmin_util.grant_sys_object('UTL_SMTP', 'MASTER_USER', 'EXECUTE',
 true);
exec rdsadmin.rdsadmin_util.grant_sys_object('DBMS_NETWORK_ACL_ADMIN',
 'MASTER_USER', 'EXECUTE', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('SESSION_PRIVS', 'MASTER_USER',
 'SELECT', true);
```

```
exec rdsadmin.rdsadmin_util.grant_sys_object('DBA_USERS', 'MASTER_USER', 'SELECT',
 true);
exec rdsadmin.rdsadmin_util.grant_sys_object('DBA_NETWORK_ACL_PRIVILEGES',
 'MASTER_USER', 'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('DBA_NETWORK_ACLS', 'MASTER_USER',
 'SELECT', true);
exec rdsadmin.rdsadmin_util.grant_sys_object('DBA_REGISTRY', 'MASTER_USER',
 'SELECT', true);
```

> **Note**
>
> I comandi precedenti si applicano a ORDS 22 e versioni successive.

6. Installa lo schema ORDS utilizzando lo `ords` script scaricato. Specificate le directory in cui contenere i file di configurazione e i file di registro. Oracle Corporation consiglia di non inserire queste directory nella directory che contiene il software del prodotto ORDS.

```
mkdir -p /home/apexuser/ords_config /home/apexuser/ords_logs

/home/apexuser/ORDS/bin/ords \
  --config /home/apexuser/ords_config \
  install --interactive --log-folder /home/apexuser/ords_logs
```

Per le istanze DB che eseguono l'architettura del database dei contenitori (CDB), utilizzate ORDS 23.2 e versioni successive e passate l'`--pdb-skip-disable-lockdown`argomento durante l'installazione di ORDS.

```
/home/apexuser/ORDS/bin/ords \
  --config /home/apexuser/ords_config \
  install --interactive --log-folder /home/apexuser/ords_logs --pdb-skip-disable-
lockdown
```

Il programma richiede le seguenti informazioni. I valori predefiniti sono riportati tra parentesi. Per altre informazioni, consulta l'articolo relativo a Introduzione a Oracle REST Data Services nella documentazione Oracle.

- `Choose the type of installation:`

  Scegliete **2** di installare gli schemi ORDS nel database e creare un pool di connessioni al database nei file di configurazione ORDS locali.

- `Specify the database connection type to use. Enter number for [1] Basic [2] TNS [3] Custom URL:`

  Scegliere il tipo di connessione desiderato. Questo esempio presuppone che tu scelga. **1**

- `Enter the name of the database server [localhost]:` ***DB_instance_endpoint***

  Scegliere il valore predefinito o digitare il valore corretto.

- `Enter the database listener port [1521]:` ***DB_instance_port***

  Scegliete il valore predefinito **1521** o immettete il valore corretto.

- `Enter the database service name [orcl]:`

  Immetti il nome del database utilizzato dall'istanza DB di RDS for Oracle.

- `Provide database user name with administrator privileges`

  Immettere il nome utente principale per l'istanza DB RDS for Oracle.

- `Enter the database password for [username]:`

  Inserisci la password utente principale per la tua istanza DB RDS for Oracle.

- `Enter the default tablespace for ORDS_METADATA and ORDS_PUBLIC_USER [SYSAUX]:`
- `Enter the temporary tablespace for ORDS_METADATA [TEMP]. Enter the default tablespace for ORDS_PUBLIC_USER [USERS]. Enter the temporary tablespace for ORDS_PUBLIC_USER [TEMP].`
- `Enter a number to select additional feature(s) to enable [1]:`
- `Enter a number to configure and start ORDS in standalone mode [1]:`

  Scegli **2** di saltare immediatamente l'avvio di ORDS in modalità standalone.

- `Enter a number to select the protocol [1] HTTP`
- `Enter the HTTP port [8080]:`
- `Enter the APEX static resources location:`

  Immettete il percorso dei file di installazione APEX (). `/home/apexuser/apex/images`

7. Impostare una password per l'utente `admin` APEX. Per farlo, utilizzare SQL*Plus per connettersi alla propria istanza database come utente master, quindi utilizzare i seguenti comandi:

```
EXEC rdsadmin.rdsadmin_util.grant_apex_admin_role;
grant APEX_ADMINISTRATOR_ROLE to master;
@/home/apexuser/apex/apxchpwd.sql
```

Sostituire *master* con il proprio nome utente master. Quando lo script `apxchpwd.sql` lo richiede, inserire una nuova password `admin`.

8. Esegui ORDS in modalità autonoma utilizzando `ords` lo script con il `serve` comando. Per le implementazioni di produzione, prendi in considerazione l'utilizzo di server applicativi Java EE supportati come Apache Tomcat o Oracle Server. WebLogic Per ulteriori informazioni, vedere Distribuzione e monitoraggio di Oracle REST Data Services nella documentazione di Oracle Database.

   ```
   /home/apexuser/ORDS/bin/ords \
     --config /home/apexuser/ords_config serve \
     --port 8193 \
     --apex-images /home/apexuser/apex/images
   ```

   Se ORDS è in esecuzione ma non è possibile accedere all'installazione di APEX, è possibile che venga visualizzato il seguente errore, in particolare sulle istanze non CDB.

   ```
   The procedure named apex_admin could not be accessed, it may not be declared,
    or the user executing this request may not have been granted execute privilege
    on the procedure, or a function specified by security.requestValidationFunction
    configuration property has prevented access.
   ```

   Per correggere questo errore, modifica la funzione di convalida della richiesta utilizzata da ORDS eseguendo lo script con il `ords` comando. `config` Per impostazione predefinita, ORDS utilizza la `ords_util.authorize_plsql_gateway` procedura, che è supportata solo sulle istanze CDB. Per le istanze non CDB, è possibile modificare questa procedura nel pacchetto. `wwv_flow_epg_include_modules.authorize` Consulta la documentazione di Oracle Database e Oracle Support per le best practice sulla configurazione della funzione di convalida delle richieste appropriata per il tuo caso d'uso.

9. Tornare alla finestra di amministrazione APEX nel proprio browser e scegliere l'opzione Administration (Amministrazione). Scegliere quindi Application Express Internal Administration (Amministrazione interna Application Express). Quando vengono richieste le credenziali, inserire le seguenti informazioni:

- User name (Nome utente – `admin`
- Password – La password impostata usando lo script `apxchpwd.sql`

Scegliere Login (Accedi), quindi impostare una nuova password per l'utente `admin`.

Il listener è ora pronto per essere utilizzato.

Impostazione del listener Oracle APEX

> **Note**
>
> Il listener Oracle APEX è obsoleto.

Amazon RDS for Oracle continua a supportare APEX versione 4.1.1 e Oracle APEX Listener versione 1.1.4. Si consiglia di utilizzare le versioni più recenti supportate di Oracle APEX e ORDS.

Installa Oracle APEX Listener in un host separato, come un'istanza Amazon EC2, un server locale nella tua azienda oppure il tuo computer desktop. Ipotizziamo che il nome del tuo host sia `myapexhost.example.com` e che esegua Linux.

Preparazione all'installazione del listener Oracle APEX

Prima di installare il listener Oracle APEX, è necessario creare un utente del sistema operativo senza privilegi, quindi scaricare e decomprimere il file di installazione APEX.

Per preparare l'installazione del listener Oracle APEX

1. Accedere a `myapexhost.example.com` come `root`.
2. Creare un utente del sistema operativo senza privilegi che possieda l'installazione del listener. Il comando seguente crea un nuovo utente denominato apexuser.

   ```
   useradd -d /home/apexuser apexuser
   ```

   Il seguente comando assegna una password al nuovo utente.

   ```
   passwd apexuser;
   ```

3. Accedi a `myapexhost.example.com` come `apexuser` e scarica il file di installazione APEX da Oracle e salvalo nella directory `/home/apexuser`:

   - http://www.oracle.com/technetwork/developer-tools/apex/downloads/index.html
   - Oracle Application Express Prior Release Archives

4. Decomprimi il file nella directory `/home/apexuser`.

   ```
   unzip apex_<version>.zip
   ```

   Dopo aver decompresso il file, troverai una directory `apex` nella directory `/home/apexuser`.

5. Rimanendo collegato a `myapexhost.example.com` come `apexuser`, scarica il file di APEX Listener da Oracle e salvalo nella directory `/home/apexuser`:

Installazione e configurazione del listener Oracle APEX

Prima di poter utilizzare APEX, è necessario scaricare il file apex.war, utilizzare Java per installare il listener Oracle APEX e quindi avviare il listener.

Per installare e configurare il listener Oracle APEX

1. Creare una nuova directory basata sul listener Oracle APEX e aprire il file listener:

   Eseguire il seguente codice:

   ```
   mkdir /home/apexuser/apexlistener
   cd /home/apexuser/apexlistener
   unzip ../apex_listener.version.zip
   ```

2. Eseguire il seguente codice.

   ```
   java -Dapex.home=./apex -Dapex.images=/home/apexuser/apex/images -Dapex.erase -
   jar ./apex.war
   ```

3. Il programma richiede le seguenti informazioni.

   - Nome dell'utente amministratore di APEX Listener. Il valore predefinito è adminlistener.
   - Una password per l'amministratore APEX Listener.
   - Nome dell'utente manager di APEX Listener. Il valore predefinito è managerlistener.
   - Una password per l'amministratore APEX Listener.

Il programma stampa un URL necessario per completare la configurazione nel modo seguente:

```
INFO: Please complete configuration at: http://localhost:8080/apex/
listenerConfigure
Database is not yet configured
```

4. Lasciare il listener Oracle APEX in esecuzione in modo da poter utilizzare Oracle Application Express. Al termine della procedura di configurazione è possibile eseguire il listener in background.
5. Dal browser Web, accedere all'URL fornito dal programma APEX Listener. Viene visualizzata la finestra di amministrazione di Oracle Application Express Listener. Immettere le seguenti informazioni:
   - Username (Nome utente – `APEX_PUBLIC_USER`
   - Password – La password per APEX_PUBLIC_USER. La password è una di quelle specificate in precedenza durante la configurazione del repository APEX. Per ulteriori informazioni, consulta Sblocco dell'account utente pubblico.
   - Connection Type (Tipo di connessione) – Basic (Di base)
   - Hostname (Nome host) – Endpoint dell'istanza database Amazon RDS, ad esempio `mydb.f9rbfa893tft.us-east-1.rds.amazonaws.com`.
   - Port (Porta – 1521
   - SID – Nome del database nell'istanza database Amazon RDS, ad esempio `mydb`.
6. Scegliere Apply (Applica). Viene visualizzata la finestra di amministrazione APEX.
7. Impostare una password per l'utente `admin` APEX. Per farlo, utilizzare SQL*Plus per connettersi alla propria istanza database come utente master, quindi utilizzare i seguenti comandi:

```
EXEC rdsadmin.rdsadmin_util.grant_apex_admin_role;
grant APEX_ADMINISTRATOR_ROLE to master;
@/home/apexuser/apex/apxchpwd.sql
```

   Sostituire *`master`* con il proprio nome utente master. Quando lo script `apxchpwd.sql` lo richiede, inserire una nuova password `admin`.
8. Tornare alla finestra di amministrazione APEX nel proprio browser e scegliere l'opzione Administration (Amministrazione). Scegliere quindi Application Express Internal Administration

(Amministrazione interna Application Express). Quando vengono richieste le credenziali, inserire le seguenti informazioni:

- User name (Nome utente – `admin`
- Password – La password impostata usando lo script `apxchpwd.sql`

Scegliere Login (Accedi), quindi impostare una nuova password per l'utente `admin`.

Il listener è ora pronto per essere utilizzato.

Aggiornamento della versione di APEX

> **Important**
>
> Effettua il backup della tua istanza database prima di aggiornare APEX. Per ulteriori informazioni, consulta Creazione di uno snapshot DB per un'istanza DB Single-AZ e Verifica di un aggiornamento del database Oracle.

Per aggiornare APEX con la tua istanza database, utilizza la seguente procedura:

- Creare un nuovo gruppo di opzioni per la versione aggiornata dell'istanza database.
- Aggiungere le versioni aggiornate di APEX e APEX-DEV al nuovo gruppo di opzioni. Assicurarsi di includere eventuali altre opzioni utilizzate dall'istanza database. Per ulteriori informazioni, consulta Considerazioni su gruppi di opzioni.
- Al momento dell'aggiornamento dell'istanza database, specificare il nuovo gruppo di opzioni per la propria istanza database aggiornata.

Dopo l'aggiornamento della versione di APEX, lo schema APEX per la precedente versione potrebbe comunque rimanere all'interno del tuo database. Se non ne hai più bisogno, puoi eliminare il vecchio schema APEX dal database dopo l'aggiornamento.

Se aggiorni la versione APEX, ma i servizi RESTful non erano configurati nella precedente versione APEX, ti consigliamo di configurarli. Per ulteriori informazioni, consulta Configurazione dei servizi RESTful per Oracle APEX.

In alcuni casi, quando hai intenzione di eseguire un aggiornamento della versione principale dell'istanza database, è possibile che stia utilizzando una versione APEX non compatibile con la versione del database di destinazione. In questi casi, è possibile aggiornare la versione di APEX prima di aggiornare l'istanza database. In questo modo puoi ridurre il tempo necessario per aggiornare la tua istanza database.

> **Note**
>
> Dopo l'aggiornamento di APEX, installa e configura un listener da utilizzare con la versione aggiornata. Per istruzioni, consulta Impostazione del listener Oracle APEX.

Rimozione dell'opzione APEX

Puoi rimuovere le opzioni APEX Amazon RDS da un'istanza database. Per rimuovere le opzioni APEX dall'istanza database, procedi in uno dei seguenti modi:

- Per rimuovere le opzioni APEX da più istanze database, rimuovile dal gruppo di opzioni a cui appartengono. Questa modifica coinvolge tutte le istanze database che usano il gruppo di opzioni. Quando rimuovi le opzioni APEX da un gruppo di opzioni associato a più istanze database, si verifica una breve interruzione mentre tutte le istanze database vengono riavviate.

  Per ulteriori informazioni, consulta Rimozione di un'opzione da un gruppo di opzioni.
- Per rimuovere l'opzione APEX da una singola istanza database, modifica l'istanza database e specifica un diverso gruppo di opzioni che non includa le opzioni APEX. È possibile specificare il gruppo di opzioni predefinito (vuoto) o un gruppo di opzioni personalizzato diverso. Quando rimuovi le opzioni APEX, si verifica una breve interruzione mentre l'istanza database viene riavviata automaticamente.

  Per ulteriori informazioni, consulta Modifica di un'istanza database Amazon RDS.

Quando rimuovi le opzioni APEX da un'istanza database, lo schema APEX viene rimosso dal tuo database.

Integrazione Amazon EFS

Amazon Elastic File System (Amazon EFS) fornisce un'archiviazione di file serverless e completamente elastica in modo da poter condividere i dati dei file senza dover fornire o gestire la capacità e le prestazioni di archiviazione. Con Amazon EFS, è possibile creare un file system e quindi montarlo nel VPC tramite il protocollo NFS versioni 4.0 e 4.1 (NFSv4). È quindi utilizzare il file system EFS come qualsiasi altro file system compatibile con POSIX. Per informazioni generali, consulta l'argomento relativo ad Amazon Elastic File System e il post del blog AWS relativo all'integrazione di Amazon RDS per Oracle con Amazon EFS.

Argomenti

Panoramica dell'integrazione Amazon EFS

Con Amazon EFS, è possibile trasferire file tra l'istanza database RDS per Oracle e un file system EFS. Ad esempio, è possibile utilizzare EFS per supportare i seguenti casi d'uso:

- Condividere un file system tra applicazioni e più server di database.
- Creare una directory condivisa per i file relativi alla migrazione, inclusi i file di dati delle tablespace trasportabili. Per ulteriori informazioni, consulta Migrazione utilizzando le tablespace trasportabili Oracle.
- Archiviare e condividere i file di redo log archiviati senza allocare spazio di archiviazione aggiuntivo sul server.
- Utilizzare le utilità di Oracle Database, ad esempio `UTL_FILE` per leggere e scrivere file.

Vantaggi dell'integrazione Amazon EFS

Quando si sceglie un file system EFS rispetto a soluzioni di trasferimento dati alternative, si ottengono i seguenti vantaggi:

- È possibile trasferire file di Oracle Data Pump tra Amazon EFS e l'istanza database RDS per Oracle. Non è necessario copiare questi file localmente perché Data Pump importa direttamente dal file system EFS. Per ulteriori informazioni, consulta Importazione di dati in Oracle in Amazon RDS.
- La migrazione dei dati è più rapida rispetto all'utilizzo di un collegamento al database.
- Si evita di allocare spazio di archiviazione sull'istanza database RDS per Oracle per contenere i file.
- Un file system EFS può dimensionare automaticamente lo spazio di archiviazione senza richiedere il provisioning.
- L'integrazione Amazon EFS non prevede tariffe o costi di configurazione minimi. I prezzi sono calcolati solo in base all'uso effettivo.

Requisiti per l'integrazione Amazon EFS

Verificare che vengano soddisfatti i seguenti requisiti:

- Il database esegue la versione 19.0.0.0.ru-2022-07.rur-2022-07.r1 o successiva.
- L'istanza database e il file system EFS devono trovarsi nella stessa Regione AWS e nello stesso VPC.
- Il VPC ha l'attributo `enableDnsSupport` abilitato. Per ulteriori informazioni, consulta Attributi DNS nel VPC nella Guida per l'utente di Amazon Virtual Private Cloud.
- Il file system EFS deve utilizzare la classe di archiviazione Standard o Standard-IA.
- Per poter utilizzare un nome DNS nel comando `mount`, devono essere soddisfatte le seguenti condizioni:
  - L'istanza database che desidera connettersi si trova all'interno di un VPC ed è configurata in modo da utilizzare il server DNS fornito da Amazon. I server DNS personalizzati non sono supportati.
  - Il VPC dell'istanza database che desidera connettersi deve avere abilitate entrambe le opzioni Risoluzione DNS e Nomi host DNS.
  - L'istanza database che desidera connettersi deve trovarsi all'interno dello stesso VPC del file system EFS.

- Utilizzare soluzioni non RDS per eseguire il backup del file system EFS. RDS per Oracle non supporta backup automatici o snapshot di database manuali di un file system EFS. Per ulteriori informazioni, consulta Backing up your Amazon EFS file systems (Backup dei file system Amazon EFS).

Configurazione delle autorizzazioni di rete per l'integrazione RDS per Oracle con Amazon EFS

Affinché RDS per Oracle si integri con Amazon EFS, assicurati che l'istanza database disponga dell'accesso di rete a un file system EFS. Per ulteriori informazioni, consulta Controlling network access to Amazon EFS file systems for NFS clients (Controllo dell'accesso di rete ai file system Amazon EFS per i client NFS) nella Guida per l'utente di Amazon Elastic File System.

Argomenti

- Controllo degli accessi di rete con i gruppi di sicurezza
- Controllo degli accesso di rete con le policy di file system

Controllo degli accessi di rete con i gruppi di sicurezza

È possibile controllare l'accesso dell'istanza database ai file system EFS utilizzando meccanismi di sicurezza a livello di rete come i gruppi di sicurezza VPC. Per consentire l'accesso a un file system EFS per l'istanza database, assicurati che il file system EFS soddisfi i seguenti requisiti:

- In ogni zona di disponibilità utilizzata da un'istanza database RDS per Oracle è presente una destinazione di montaggio EFS.

  Una destinazione di montaggio EFS fornisce un indirizzo IP per un endpoint NFSv4 in cui è possibile montare un file system EFS;. Il file system è montato utilizzando il relativo nome DNS, che si risolverà nell'indirizzo IP della destinazione di montaggio di EFS nella stessa zona di disponibilità dell'istanza database.

  Puoi configurare istanze database in zone di disponibilità diverse in modo che utilizzino lo stesso file system EFS. Per implementazioni Multi-AZ, è necessario disporre di un punto di montaggio per ogni zona di disponibilità definita nell'implementazione interessata. Potrebbe essere necessario spostare un'istanza database in una zona di disponibilità diversa. Per questi motivi, è consigliabile creare un punto di montaggio EFS in ogni zona di disponibilità presente nel cloud privato virtuale

(VPC) in uso. Per impostazione predefinita, quando crei un nuovo file system EFS utilizzando la console, RDS crea destinazioni di montaggio per tutte le zone di disponibilità.

- Un gruppo di sicurezza è collegato alla destinazione di montaggio.
- Il gruppo di sicurezza dispone di una regola in entrata per autorizzare la sottorete o il gruppo di sicurezza di rete dell'istanza database RDS per Oracle su TCP/2049 (tipo NFS).

Per ulteriori informazioni, consulta Creazione di file system Amazon EFS e Creating and managing EFS mount targets and security groups (Creazione e gestione delle destinazioni di montaggio e dei gruppi di sicurezza EFS) nella Guida per l'utente di Amazon Elastic File System.

Controllo degli accesso di rete con le policy di file system

L'integrazione Amazon EFS con RDS per Oracle funziona con la policy di file system EFS predefinita (vuota). La policy predefinita non utilizza IAM per l'autenticazione. Garantisce invece l'accesso completo a qualsiasi client anonimo in grado di connettersi al file system utilizzando una destinazione di montaggio. La policy predefinita è effettiva ogni volta che non esiste una policy di file system configurata dall'utente, anche a livello di creazione del file system. Per ulteriori informazioni, consulta Policy EFS predefinita per il file system EFS nella Guida per l'utente di Amazon Elastic File System.

Per rafforzare l'accesso al file system EFS per tutti i client, incluso RDS per Oracle, puoi configurare le autorizzazioni IAM. Con questo approccio si crea una policy di file system. Per ulteriori informazioni, consulta Creating file system policies (Creazione di policy di file system) nella Guida per l'utente di Amazon Elastic File System.

Configurazione delle autorizzazioni IAM per l'integrazione RDS per Oracle con Amazon EFS

Per impostazione predefinita, la funzionalità di integrazione di Amazon EFS non utilizza un ruolo IAM: l'impostazione dell'`USE_IAM_ROLE`opzione è`FALSE`. Per integrare RDS for Oracle con Amazon EFS e un ruolo IAM, l'istanza DB deve disporre delle autorizzazioni IAM per accedere a un file system Amazon EFS.

Argomenti

- Fase 1: creazione di un ruolo IAM per l'istanza database e collegamento della policy
- Fase 2: creazione di una policy per il file system Amazon EFS
- Fase 3: associazione del ruolo IAM all'istanza database RDS per Oracle

Fase 1: creazione di un ruolo IAM per l'istanza database e collegamento della policy

In questa fase, viene creato un ruolo per l'istanza database RDS per Oracle per consentire ad Amazon RDS di accedere al file system EFS.

Console

Per creare un ruolo IAM per consentire ad Amazon RDS di accedere a un file system EFS

1. Aprire la console di gestione IAM.
2. Nel pannello di navigazione, seleziona Roles (Ruoli).
3. Selezionare Create role (Crea ruolo).
4. In AWS Service scegliere RDS.
5. Per Select your use case (Seleziona caso di utilizzo), selezionare RDS – Add Role to Database (RDS – Aggiungi ruolo al database).
6. Seleziona Avanti.
7. Non aggiungere alcuna policy di autorizzazione. Seleziona Avanti.
8. Impostare Role Name (Nome ruolo) su un nome per il ruolo IAM, ad esempio `rds-efs-integration-role`. È anche possibile aggiungere un valore Description (Descrizione) facoltativo.
9. Scegli Crea ruolo.

AWS CLI

Per limitare le autorizzazioni del servizio a una risorsa specifica, si consiglia di utilizzare le chiavi di contesto delle condizioni globali `aws:SourceArn` e `aws:SourceAccount` nelle relazioni di trust basate sulle risorse. Questo è il modo più efficace per proteggersi dal problema di deputy confused.

Puoi usare le chiavi di contesto delle condizioni globali e avere il valore `aws:SourceArn` che contiene l'ID dell'account. In questo caso, il valore `aws:SourceAccount` e l'account nel valore `aws:SourceArn` deve utilizzare lo stesso ID account quando viene utilizzato nella stessa istruzione.

- Utilizzare `aws:SourceArn` se si desidera un accesso cross-service per una singola risorsa.
- Utilizzare `aws:SourceAccount` se si desidera consentire l'associazione di qualsiasi risorsa in tale account all'uso cross-service.

Nella relazione di trust, assicurati di utilizzare la chiave di contesto della condizione globale `aws:SourceArn` con l'Amazon Resource Name (ARN) completo delle risorse che accedono al ruolo.

Il seguente comando AWS CLI crea il ruolo chiamato *rds-efs-integration-role* a questo proposito.

Example

PerLinux, omacOS: Unix

```
aws iam create-role \
   --role-name rds-efs-integration-role \
   --assume-role-policy-document '{
     "Version": "2012-10-17",
     "Statement": [
       {
         "Effect": "Allow",
         "Principal": {
            "Service": "rds.amazonaws.com"
          },
         "Action": "sts:AssumeRole",
         "Condition": {
             "StringEquals": {
                 "aws:SourceAccount": my_account_ID,
                 "aws:SourceArn": "arn:aws:rds:Region:my_account_ID:db:dbname"
             }
         }
       }
     ]
   }'
```

Per Windows:

```
aws iam create-role ^
   --role-name rds-efs-integration-role ^
   --assume-role-policy-document '{
     "Version": "2012-10-17",
     "Statement": [
       {
         "Effect": "Allow",
         "Principal": {
            "Service": "rds.amazonaws.com"
```

```
         },
        "Action": "sts:AssumeRole",
        "Condition": {
            "StringEquals": {
                "aws:SourceAccount": my_account_ID,
                "aws:SourceArn": "arn:aws:rds:Region:my_account_ID:db:dbname"
            }
        }
      }
    ]
}'
```

Per ulteriori informazioni, consulta la pagina relativa alla creazione di un ruolo per delegare le autorizzazioni a un utente IAM nella Guida per l'utente IAM.

Fase 2: creazione di una policy per il file system Amazon EFS

In questa fase viene creata una policy per il file system EFS.

Per creare o modificare una policy di file system EFS

1. Apri la console di gestione EFS.
2. Selezionare File Systems (File system).
3. Nella pagina File systems (File system), scegli il file system per cui vuoi creare una policy di file system. Viene visualizzata la pagina dei dettagli per il file system scelto.
4. Scegli la scheda File system policy (Policy di file system).

   Se è vuota, viene utilizzata la policy di file system EFS predefinita. Per ulteriori informazioni, consulta Policy EFS predefinita per il file system EFS nella Guida per l'utente di Amazon Elastic File System.
5. Scegli Modifica. Viene visualizzata la pagina Policy del file system.
6. Nell'editor di policy, immetti una policy come la seguente, quindi scegli Save (Salva).

```
{
    "Version": "2012-10-17",
    "Id": "ExamplePolicy01",
    "Statement": [
        {
            "Sid": "ExampleStatement01",
            "Effect": "Allow",
```

```
            "Principal": {
                "AWS": "arn:aws:iam::123456789012:role/rds-efs-integration-role"
            },
            "Action": [
                "elasticfilesystem:ClientMount",
                "elasticfilesystem:ClientWrite",
                "elasticfilesystem:ClientRootAccess"
            ],
            "Resource": "arn:aws:elasticfilesystem:us-east-1:123456789012:file-
system/fs-1234567890abcdef0"
        }
    ]
}
```

Fase 3: associazione del ruolo IAM all'istanza database RDS per Oracle

In questa fase il ruolo IAM viene associato all'istanza database. Tieni presenti i seguenti requisiti:

- Devi disporre dell'accesso a un ruolo IAM a cui è collegata la policy di autorizzazione richiesta di Amazon EFS.
- È possibile associare un solo ruolo IAM alla volta all'istanza database RDS per Oracle.
- Lo stato dell'istanza deve essere Available (Disponibile).

Per ulteriori informazioni, consulta Identity and access management for Amazon EFS (Identity and Access Management per Amazon EFS) nella Guida per l'utente di Amazon Elastic File System.

Console

Per associare il ruolo IAM all'istanza database RDS per Oracle

1. Accedere alla AWS Management Console e aprire la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Scegli Database.
3. Se l'istanza del database non è disponibile, scegli Operazioni , quindi Avvia. Quando lo stato dell'istanza mostra Avviato, vai al passaggio successivo.
4. Scegliere il nome dell'istanza database Oracle per visualizzarne i dettagli.
5. Sulla scheda Connettività e sicurezza, scorri verso il basso fino alla sezione Gestisci i ruoli IAM in fondo alla pagina.

6. Scegli il ruolo da aggiungere nella sezione Aggiungi ruoli IAM a questa istanza.
7. Per Feature (Caratteristica) scegli EFS_INTEGRATION.
8. Scegliere Add role (Aggiungi ruolo).

AWS CLI

Il comando AWS CLI seguente aggiunge il ruolo a un'istanza database Oracle denominata *mydbinstance*.

Example

Per LinuxmacOS, oUnix:

```
aws rds add-role-to-db-instance \
   --db-instance-identifier mydbinstance \
   --feature-name EFS_INTEGRATION \
   --role-arn your-role-arn
```

Per Windows:

```
aws rds add-role-to-db-instance ^
   --db-instance-identifier mydbinstance ^
   --feature-name EFS_INTEGRATION ^
   --role-arn your-role-arn
```

Sostituire *your-role-arn* con il ruolo ARN annotato nel passaggio precedente. `EFS_INTEGRATION` deve essere specificato per l'opzione `--feature-name`.

Aggiunta dell'opzione EFS_INTEGRATION

Per utilizzare l'integrazione Amazon RDS per Oracle con Amazon EFS, l'istanza database deve essere associata a un gruppo di opzioni che include l'opzione `EFS_INTEGRATION`.

Più istanze database Oracle che appartengono allo stesso gruppo di opzioni condividono lo stesso file system EFS. Istanze database diverse possono accedere agli stessi dati e l'accesso può essere diviso in diverse directory Oracle. Per ulteriori informazioni, consulta Trasferimento di file tra RDS per Oracle e un file system Amazon EFS.

Console

Per configurare un gruppo di opzioni per l'integrazione Amazon EFS

1. Creare un nuovo gruppo di opzioni o identificare un gruppo opzioni esistente a cui è possibile aggiungere l'opzione `EFS_INTEGRATION`.

   Per informazioni sulla creazione di un gruppo di opzioni, consulta Creazione di un gruppo di opzioni.
2. Aggiungere l'opzione `EFS_INTEGRATION` al gruppo di opzioni. È necessario specificare l'ID file system `EFS_ID` e impostare il flag `USE_IAM_ROLE`.

   Per ulteriori informazioni, consulta Aggiunta di un'opzione a un gruppo di opzioni.
3. Associa il gruppo di opzioni all'istanza database in uno dei seguenti modi:

   - Crea una nuova istanza database Oracle e associa il gruppo di opzioni. Per informazioni sulla creazione di un'istanza database, consulta Creazione di un'istanza database Amazon RDS.
   - Modifica un'istanza database Oracle per associare il gruppo di opzioni. Per informazioni sulla modifica di un'istanza database di Oracle, consulta Modifica di un'istanza database Amazon RDS.

AWS CLI

Per configurare un gruppo di opzioni per l'integrazione EFS

1. Creare un nuovo gruppo di opzioni o identificare un gruppo opzioni esistente a cui è possibile aggiungere l'opzione `EFS_INTEGRATION`.

   Per informazioni sulla creazione di un gruppo di opzioni, consulta Creazione di un gruppo di opzioni.
2. Aggiungere l'opzione `EFS_INTEGRATION` al gruppo di opzioni.

   Ad esempio, il comando seguente AWS CLI aggiunge l'opzione `EFS_INTEGRATION` a un gruppo opzioni denominato **`myoptiongroup`**.

   Example

   Per LinuxmacOS, oUnix:

```
aws rds add-option-to-option-group \
   --option-group-name myoptiongroup \
   --options "OptionName=EFS_INTEGRATION,OptionSettings=\
   [{Name=EFS_ID,Value=fs-1234567890abcdef0},{Name=USE_IAM_ROLE,Value=TRUE}]"
```

Per Windows:

```
aws rds add-option-to-option-group ^
   --option-group-name myoptiongroup ^
   --options "OptionName=EFS_INTEGRATION,OptionSettings=^
   [{Name=EFS_ID,Value=fs-1234567890abcdef0},{Name=USE_IAM_ROLE,Value=TRUE}]"
```

3. Associa il gruppo di opzioni all'istanza database in uno dei seguenti modi:

   - Crea una nuova istanza database Oracle e associa il gruppo di opzioni. Per informazioni sulla creazione di un'istanza database, consulta Creazione di un'istanza database Amazon RDS.
   - Modifica un'istanza database Oracle per associare il gruppo di opzioni. Per informazioni sulla modifica di un'istanza database di Oracle, consulta Modifica di un'istanza database Amazon RDS.

Configurazione delle autorizzazioni del file system Amazon EFS

Per impostazione predefinita, solo l'utente root (UID `0`) dispone delle autorizzazioni di lettura, scrittura ed esecuzione per un file system EFS appena creato. Affinché gli altri utenti possano modificare il file system, l'utente root deve esplicitamente concedere loro l'accesso. L'utente dell'istanza database RDS per Oracle rientra nella categoria `others`. Per ulteriori informazioni, consulta Working with users, groups, and permissions at the Network File System (NFS) Level (Utilizzo di utenti, gruppi e autorizzazioni a livello NFS (Network File System) nella Guida per l'utente di Amazon Elastic File System.

Per consentire all'istanza database RDS per Oracle di leggere e scrivere file nel file system EFS, procedi come segue:

- Monta un file system EFS localmente nell'istanza on-premise o Amazon EC2.
- Configura le autorizzazioni in modo dettagliato.

Ad esempio, per concedere a `other` utenti le autorizzazioni di scrittura nella root del file system EFS, esegui `chmod 777` sulla directory. Per ulteriori informazioni, consulta Esempio di casi d'uso e autorizzazioni del file system Amazon EFS nella Guida per l'utente di Amazon Elastic File System.

Trasferimento di file tra RDS per Oracle e un file system Amazon EFS

Per trasferire file tra un'istanza RDS per Oracle e un file system Amazon EFS, crea almeno una directory Oracle e configura le autorizzazioni del file system EFS per controllare l'accesso all'istanza database.

Argomenti

- Creazione di una directory Oracle
- Trasferimento di dati da e verso un file system EFS - Esempi

Creazione di una directory Oracle

Per creare una directory Oracle, usa la procedura `rdsadmin.rdsadmin_util.create_directory_efs`. La procedura include i seguenti parametri.

Nome del parametro	Tipo di dati	Default	Campo obbligatorio	Descrizione
`p_directory_name`	VARCHAR	–	Sì	Il nome della directory Oracle.
`p_path_on_efs`	VARCHAR	–	Sì	Il percorso del file system EFS. Il prefisso del nome del percorso utilizza lo schema `/rdsefs-`*`fsid`*`/`, dove *`fsid`* è un segnaposto per l'ID file system EFS. Ad esempio, se il file system EFS è denominato `fs-1234567890abcdef0` e crei una sottodirectory su questo file system denominata `mydir`, puoi specificare il seguente valore:

Nome del parametro	Tipo di dati	Defaul	Campo obbliga rio	Descrizione
				`/rdsefs-fs-1234567890abcdef0/mydir`

Supponi di creare una sottodirectory denominata `/datapump1` nel file system EFS `fs-1234567890abcdef0`. L'esempio seguente crea una directory Oracle `DATA_PUMP_DIR_EFS` che punta alla directory `/datapump1` sul file system EFS. Il valore del percorso del file system per il parametro`p_path_on_efs` è preceduto dalla stringa `/rdsefs-`.

```
BEGIN
  rdsadmin.rdsadmin_util.create_directory_efs(
    p_directory_name => 'DATA_PUMP_DIR_EFS',
    p_path_on_efs    => '/rdsefs-fs-1234567890abcdef0/datapump1');
END;
/
```

Trasferimento di dati da e verso un file system EFS - Esempi

Nell'esempio seguente si usa Oracle Data Pump per esportare la tabella denominata `MY_TABLE` nel file `datapump.dmp`. Questo file si trova in un file system EFS.

```
DECLARE
  v_hdnl NUMBER;
BEGIN
  v_hdnl := DBMS_DATAPUMP.OPEN(operation => 'EXPORT', job_mode => 'TABLE',
 job_name=>null);
  DBMS_DATAPUMP.ADD_FILE(
    handle    => v_hdnl,
    filename  => 'datapump.dmp',
    directory => 'DATA_PUMP_DIR_EFS',
    filetype  => dbms_datapump.ku$_file_type_dump_file);
  DBMS_DATAPUMP.ADD_FILE(
    handle    => v_hdnl,
    filename  => 'datapump-exp.log',
    directory => 'DATA_PUMP_DIR_EFS',
    filetype  => dbms_datapump.ku$_file_type_log_file);
  DBMS_DATAPUMP.METADATA_FILTER(v_hdnl,'NAME_EXPR','IN (''MY_TABLE'')');
  DBMS_DATAPUMP.START_JOB(v_hdnl);
```

```
END;
/
```

Nell'esempio seguente si usa Oracle Data Pump per importare la tabella denominata `MY_TABLE` dal file `datapump.dmp`. Questo file si trova in un file system EFS.

```
DECLARE
  v_hdnl NUMBER;
BEGIN
  v_hdnl := DBMS_DATAPUMP.OPEN(
    operation => 'IMPORT',
    job_mode  => 'TABLE',
    job_name  => null);
  DBMS_DATAPUMP.ADD_FILE(
    handle    => v_hdnl,
    filename  => 'datapump.dmp',
    directory => 'DATA_PUMP_DIR_EFS',
    filetype  => dbms_datapump.ku$_file_type_dump_file );
  DBMS_DATAPUMP.ADD_FILE(
    handle    => v_hdnl,
    filename  => 'datapump-imp.log',
    directory => 'DATA_PUMP_DIR_EFS',
    filetype  => dbms_datapump.ku$_file_type_log_file);
  DBMS_DATAPUMP.METADATA_FILTER(v_hdnl,'NAME_EXPR','IN (''MY_TABLE'')');
  DBMS_DATAPUMP.START_JOB(v_hdnl);
END;
/
```

Per ulteriori informazioni, consulta Importazione di dati in Oracle in Amazon RDS.

Rimozione dell'opzione EFS_INTEGRATION

Per rimuovere l'opzione `EFS_INTEGRATION` dall'istanza database RDS per Oracle, procedi in uno dei seguenti modi:

- Per rimuovere l'opzione `EFS_INTEGRATION` da più istanze database, rimuovere l'opzione `EFS_INTEGRATION` dal gruppo di opzioni a cui appartengono le istanze database. Questa modifica coinvolge tutte le istanze database che usano il gruppo di opzioni. Per ulteriori informazioni, consulta Rimozione di un'opzione da un gruppo di opzioni.
- Per rimuovere l'opzione `EFS_INTEGRATION` da una singola istanza database, modifica l'istanza e specifica un gruppo di opzioni diverso che non comprenda l'opzione `EFS_INTEGRATION`. È

possibile specificare il gruppo di opzioni predefinito (vuoto) o un gruppo di opzioni personalizzato diverso. Per ulteriori informazioni, consulta Modifica di un'istanza database Amazon RDS.

Risoluzione dei problemi di integrazione Amazon EFS

L'istanza database RDS per Oracle monitora la connettività al file system Amazon EFS. Quando il monitoraggio rileva un problema, viene eseguito un tentativo di correzione e la pubblicazione di un evento nella console RDS. Per ulteriori informazioni, consulta Visualizzazione di eventi Amazon RDS.

Utilizza le informazioni contenute in questa sezione per diagnosticare e risolvere problemi comuni quando si utilizza l'integrazione Amazon EFS.

Notification	Descrizione	Azione
`The EFS for RDS Oracle instance` *`instance_name`* `isn't available on the primary host. NFS port 2049 of your EFS isn't reachable.`	L'istanza database non è in grado di comunicare con il file system EFS.	Assicurati che siano soddisfatte le seguenti condizioni: • Il file system EFS è presente. • Il gruppo di sicurezza collegato alla destinazione di montaggio EFS dispone di una regola in entrata per autorizzare il gruppo di sicurezza o la sottorete di rete dell'istanza database RDS per Oracle su TCP/2049 (tipo NFS).
`The EFS isn't reachable.`	Si è verificato un errore durante l'installazione dell'opzione `EFS_INTEGRATION` .	Assicurati che siano soddisfatte le seguenti condizioni: • Il file system EFS è presente. • Il gruppo di sicurezza collegato alla destinazione di montaggio EFS dispone di una regola in

Notification	Descrizione	Azione
		entrata per autorizzare il gruppo di sicurezza o la sottorete di rete dell'istanza database RDS per Oracle su TCP/2049 (tipo NFS). • L'attributo `enableDnsSupport` è attivato per il VPC. • Stai utilizzando il server DNS fornito da Amazon nel tuo cloud privato virtuale (VPC). L'integrazione con Amazon EFS non funziona con un DNS DHCP personalizzato.
`The associated role with your DB instance wasn't found.`	Si è verificato un errore durante l'installazione dell'opzione `EFS_INTEGRATION` .	Assicurati di aver associato un ruolo IAM all'istanza database RDS per Oracle.
`The associated role with your DB instance wasn't found.`	Si è verificato un errore durante l'installazione dell'opzione `EFS_INTEGRATION` . RDS per Oracle è stato ripristinato da uno snapshot del DB con l'impostazione dell'`USE_IAM_ROLE` opzione di. `TRUE`	Assicurati di aver associato un ruolo IAM all'istanza database RDS per Oracle.

Notification	Descrizione	Azione
`The associated role with your DB instance wasn't found.`	Si è verificato un errore durante l'installazione dell'opzione `EFS_INTEGRATION` . RDS for Oracle è stato creato a partire da un all-in-one CloudFormation modello con l'`USE_IAM_ROLE` opzione impostata di. `TRUE`	Come soluzione alternativa, completa i seguenti passaggi: 1. Crea un'istanza DB con il ruolo IAM e il gruppo di opzioni predefinito. 2. In un successivo aggiornamento dello stack, aggiungi il gruppo di opzioni personalizzato con l'`EFS_INTEGRATION` opzione.
`PLS-00302: component 'CREATE_DIRECTORY_EFS' must be declared`	Questo errore può verificarsi quando si utilizza una versione di RDS per Oracle che non supporta Amazon EFS.	Assicurati di utilizzare l'istanza database RDS per Oracle versione 19.0.0.0.ru-2022-07.rur-2022-07.r1 o successive.
`Read access of your EFS is denied. Check your file system policy.`	L'istanza database non è in grado di leggere il file system EFS.	Assicurati che il file system EFS consenta l'accesso in lettura tramite il ruolo IAM o a livello di file system EFS.
N/D	L'istanza database non è in grado di scrivere nel file system EFS.	Utilizza le fasi seguenti: 1. Assicurati che il file system EFS sia montato su un'istanza Amazon EC2. 2. Fornisci l'accesso in scrittura di gruppo `others` all'utente RDS. La tecnica più semplice consiste nell'eseguire il comando `chmod 777` nella directory principale del file system EFS.

Notification	Descrizione	Azione
Il comando `host -s` restituisce *`hostname`* `not found: 3(NXDOMAIN)` .	Stai utilizzando un server DNS personalizzato.	Per poter utilizzare un nome DNS nel comando `mount`, devono essere soddisfatte le seguenti condizioni: • L'istanza database che desidera connettersi si trova all'interno di un VPC ed è configurata in modo da utilizzare il server DNS fornito da Amazon. I server DNS personalizzati non sono supportati. • Il VPC dell'istanza database che desidera connetter si deve avere abilitate entrambe le opzioni Risoluzione DNS e Nomi host DNS. • L'istanza database che desidera connettersi deve trovarsi all'interno dello stesso VPC del file system EFS.

Oracle Java Virtual Machine

Amazon RDS supporta Oracle Java Virtual Machine (JVM) tramite l'utilizzo dell'opzione `JVM`. Oracle Java fornisce uno schema SQL e funzioni che facilitano l'utilizzo delle caratteristiche Oracle Java in un database Oracle. Per ulteriori informazioni, consulta l'articolo relativo all'introduzione a Java in Oracle Database nella documentazione Oracle.

Puoi utilizzare Oracle JVM con le versioni seguenti di Oracle Database:

- Oracle Database 21c (21.0.0), tutte le versioni
- Oracle Database 19c (19.0.0), tutte le versioni
- Oracle Database 12c Release 2 (12.2), tutte le versioni
- Oracle Database 12c Release 1 (12.1), versione 12.1.0.2.v13 e versioni successive

L'implementazione Java in Amazon RDS ha un set di autorizzazioni limitato. L'utente master riveste il ruolo `RDS_JAVA_ADMIN`, che concede un sottoinsieme dei privilegi concessi dal ruolo `JAVA_ADMIN`. Per elencare i privilegi concessi al ruolo `RDS_JAVA_ADMIN`, esegui la query seguente sull'istanza database:

```
SELECT * FROM dba_java_policy
   WHERE grantee IN ('RDS_JAVA_ADMIN', 'PUBLIC')
   AND enabled = 'ENABLED'
   ORDER BY type_name, name, grantee;
```

Prerequisiti per Oracle JVM

Di seguito sono indicati i prerequisiti per l'utilizzo di Oracle Java:

- L'istanza database deve appartenere a una classe di dimensioni sufficienti. Oracle Java non è supportato per le classi di istanza database db.t3.micro o db.t3.small. Per ulteriori informazioni, consulta Classi di istanze database.
- L'istanza database deve avere l'opzione Auto Minor Version Upgrade (Aggiornamento minore automatico della versione) abilitata. Questa opzione consente all'istanza database di ricevere automaticamente gli aggiornamenti della versione secondaria del motore database quando diventano disponibili. Amazon RDS utilizza questa opzione per aggiornare l'istanza database all'ultimo PSU (Patch Set Update) o RU (Release Update) di Oracle. Per ulteriori informazioni, consulta Modifica di un'istanza database Amazon RDS.

Best practice per Oracle JVM

Di seguito sono indicate le best practice per l'utilizzo di Oracle Java:

- Per la massima sicurezza, è necessario utilizzare l'opzione `JVM` con Secure Sockets Layer (SSL). Per ulteriori informazioni, consulta Oracle Secure Sockets Layer.
- Configurare l'istanza database per limitare l'accesso alla rete. Per ulteriori informazioni, consulta Scenari per accedere a un'istanza database in un VPC e Uso di un'istanza database in un VPC.
- Aggiorna la configurazione degli endpoint HTTPS per supportare TLSv1.2 se si soddisfano le seguenti condizioni:
  - Utilizza Oracle Java Virtual Machine (JVM) per connetterti a un endpoint HTTPS tramite protocolli TLSv1 o TLSv1.1.
  - L'endpoint non supporta il protocollo TLSv1.2.
  - Non hai applicato l'aggiornamento della release di aprile 2021 al tuo Oracle DB.

  Aggiornando la configurazione dell'endpoint, assicurati che la connettività della JVM all'endpoint HTTPS continui a funzionare. Per ulteriori informazioni sulle modifiche di TLS in Oracle JRE e JDK, consulta Oracle JRE and JDK Cryptographic Roadmap.

Aggiunta dell'opzione Oracle JVM

Di seguito è riportato il processo generale per aggiungere l'opzione `JVM` a un'istanza database:

1. Creare un nuovo gruppo di opzioni oppure copiare o modificare un gruppo di opzioni esistente.
2. Aggiungere l'opzione al gruppo di opzioni.
3. Associare il gruppo di opzioni a questa istanza database.

Si verificherà una breve interruzione durante l'aggiunta dell'opzione `JVM`. Una volta aggiunta l'opzione, non è necessario riavviare l'istanza database. Non appena il gruppo di opzioni diventa attivo, Oracle Java è disponibile.

> **Note**
>
> Durante questa interruzione, le funzioni di verifica delle password vengono temporaneamente disabilitate. Durante l'interruzione si possono verificare eventi correlati alle funzioni di verifica

delle password. Le funzioni di verifica delle password vengono riabilitate prima di rendere disponibile l'istanza database Oracle.

Per aggiungere l'opzione JVM a un'istanza database

1. Determinare il gruppo di opzioni che si vuole usare. È possibile creare un nuovo gruppo di opzioni oppure usare un gruppo di opzioni esistente. Se si desidera usare un gruppo di opzioni esistente, puoi passare alla fase successiva. In caso contrario, creare un gruppo di opzioni database personalizzato con le seguenti impostazioni:

   - Per Engine (Motore) scegliere il motore database utilizzato dall'istanza database (oracle-ee, oracle-se, oracle-se1 o oracle-se2).
   - In Major engine version (Versione principale del motore), scegliere la versione dell'istanza database.

   Per ulteriori informazioni, consulta Creazione di un gruppo di opzioni.
2. Aggiungere l'opzione JVM al gruppo di opzioni. Per ulteriori informazioni sull'aggiunta di opzioni, consulta Aggiunta di un'opzione a un gruppo di opzioni.
3. Applicare il gruppo di opzioni a un'istanza database nuova o esistente:

   - Per una nuova istanza database, applicare il gruppo di opzioni quando viene avviata l'istanza. Per ulteriori informazioni, consulta Creazione di un'istanza database Amazon RDS.
   - Per un'istanza database esistente, applicare il gruppo di opzioni modificando l'istanza e collegando il nuovo gruppo di opzioni. Per ulteriori informazioni, consulta Modifica di un'istanza database Amazon RDS.
4. Concedere le autorizzazioni richieste agli utenti.

   L'utente master di Amazon RDS ha le autorizzazioni per utilizzare l'opzione `JVM` per impostazione predefinita. Se altri utenti richiedono queste autorizzazioni, connettersi all'istanza database come utente master in un client SQL e concedere le autorizzazioni a questi utenti.

   Nell'esempio seguente le autorizzazioni all'utilizzo dell'opzione `JVM` vengono concesse all'utente `test_proc`.

```
create user test_proc identified by password;
```

```
CALL dbms_java.grant_permission('TEST_PROC',
 'oracle.aurora.security.JServerPermission', 'LoadClassInPackage.*', '');
```

> **Note**
>
> Specifica una password diversa dal prompt mostrato qui come best practice per la sicurezza.

Dopo aver concesso le autorizzazioni, la query seguente dovrebbe restituire un output.

```
select * from dba_java_policy where grantee='TEST_PROC';
```

> **Note**
>
> Il nome utente Oracle rispetta la distinzione tra maiuscole e minuscole e di solito è formato solo da caratteri maiuscoli.

Rimozione dell'opzione Oracle JVM

Puoi rimuovere l'opzione `JVM` dall'istanza database. Si verificherà una breve interruzione durante la rimozione dell'opzione . Una volta rimossa l'opzione `JVM` non è necessario riavviare la tua istanza database.

> **Warning**
>
> La rimozione dell'opzione `JVM` può causare la perdita di dati se l'istanza database utilizza i tipi di dati abilitati come parte dell'opzione. Eseguire il backup dei dati prima di procedere. Per ulteriori informazioni, consulta Backup, ripristino ed esportazione dei dati.

Per rimuovere l'opzione `JVM` dall'istanza database, procedi in uno dei seguenti modi:

- Rimuovere l'opzione `JVM` dal gruppo di opzioni a cui appartiene. Questa modifica coinvolge tutte le istanze database che usano il gruppo di opzioni. Per ulteriori informazioni, consulta Rimozione di un'opzione da un gruppo di opzioni.

- Modificare l'istanza database e specificare un diverso gruppo di opzioni che non comprenda l'opzione `JVM`. Questa modifica coinvolge una singola istanza database. È possibile specificare il gruppo di opzioni predefinito (vuoto) o un gruppo di opzioni personalizzato diverso. Per ulteriori informazioni, consulta Modifica di un'istanza database Amazon RDS.

Oracle Enterprise Manager

Amazon RDS supporta Oracle Enterprise Manager (OEM). OEM è la linea di prodotti Oracle per la gestione integrata della tecnologia informatica aziendale.

Amazon RDS supporta OEM tramite le seguenti opzioni.

Opzione	ID opzione	Versioni OEM supportate	Versioni Oracle Database supportate
OEM Database Express	OEM	OEM Database Express 12c	Oracle Database 19c (solo non CDB) Database Oracle 12c
OEM Management Agent	OEM_AGEN	OEM Cloud Control per 13c OEM Cloud Control per 12c	Oracle Database 19c (solo non CDB) Database Oracle 12c

> **Note**
>
> Puoi utilizzare OEM Database oppure OEM Management Agent, ma non entrambi.

> **Note**
>
> Queste opzioni non sono supportate per l'architettura Oracle multi-tenant.

Oracle Enterprise Manager Database Express

Amazon RDS supporta Oracle Enterprise Manager (OEM) Database Express tramite l'utilizzo dell'opzione OEM. Amazon RDS supporta Oracle Enterprise Manager Database Express per le seguenti versioni:

- Oracle Database 19c (solo non CDB)
- Database Oracle 12c

OEM Database Express e Database Control sono strumenti simili con un interfaccia basata sul Web per l'amministrazione di Oracle Database. Per ulteriori informazioni su questi strumenti, consulta Accesso a Enterprise Manager Database Express 18c e Accesso a Enterprise Manager 12c Database Control nella documentazione Oracle.

Di seguito è riportata una limitazione per OEM Database Express:

- OEM Database Express non è supportato nelle classi di istanze database db.t3.micro o db.t3.small.

  Per altre informazioni sulle classi di istanza database, consulta Classi di istanza RDS for Oracle.

Impostazioni dell'opzione OEM Database

Amazon RDS supporta le seguenti impostazioni per l'opzione OEM.

Impostazione opzioni	Valori validi	Descrizione
Porta	Un valore intero.	La porta dell'istanza database in ascolto di OEM Database. L'impostazione predefinita per OEM Database Express è 5500.
Gruppi di sicurezza	—	Un gruppo di sicurezza che ha accesso a Port (Porta).

Aggiunta dell'opzione OEM Database

Di seguito è riportato il processo generale per aggiungere l'opzione OEM a un'istanza database:

1. Creare un nuovo gruppo di opzioni oppure copiare o modificare un gruppo di opzioni esistente.
2. Aggiungere l'opzione al gruppo di opzioni.
3. Associare il gruppo di opzioni a questa istanza database.

Quando aggiungi l'opzione OEM a un'istanza database Oracle Database 12c o successivo, si verifica una breve interruzione durante il riavvio automatico dell'istanza database.

Per aggiungere l'opzione OEM a un'istanza database

1. Determinare il gruppo di opzioni che si desidera usare. È possibile creare un nuovo gruppo di opzioni oppure usare un gruppo di opzioni esistente. Se si desidera usare un gruppo di opzioni esistente, puoi passare alla fase successiva. In caso contrario, creare un gruppo di opzioni database personalizzato con le seguenti impostazioni:

   a. Per Engine (Motore) scegliere l'edizione Oracle per l'istanza database.
   b. In Major engine version (Versione principale del motore), scegliere la versione dell'istanza database.

   Per ulteriori informazioni, consulta Creazione di un gruppo di opzioni.
2. Aggiungere l'opzione OEM al gruppo di opzioni e configurare le impostazioni dell'opzione. Per ulteriori informazioni sull'aggiunta di opzioni, consulta Aggiunta di un'opzione a un gruppo di opzioni. Per ulteriori informazioni su ciascuna impostazione, consulta Impostazioni dell'opzione OEM Database.

   > **Note**
   >
   > Se aggiungi l'opzione OEM a un gruppo di opzioni esistente già associato a una o più istanze database di Oracle Database 19c (solo non CDB) o di Oracle Database 12c, si verifica una breve interruzione durante il riavvio automatico di tutte le istanze database.
3. Applicare il gruppo di opzioni a un'istanza database nuova o esistente:

   - Per una nuova istanza database, si applica il gruppo di opzioni quando viene avviata l'istanza. Per ulteriori informazioni, consulta Creazione di un'istanza database Amazon RDS.
   - Per un'istanza database esistente, viene applicato il gruppo di opzioni modificando l'istanza e collegando il nuovo gruppo di opzioni. Quando aggiungi l'opzione OEM a un'istanza database di Oracle Database 19c(solo non CDB) o a un'istanza database Oracle Database 12c, si

verifica una breve interruzione durante il riavvio automatico dell'istanza database. Per ulteriori informazioni, consultare Modifica di un'istanza database Amazon RDS.

> **Note**
>
> Puoi inoltre utilizzare AWS CLI per aggiungere l'opzione OEM. Per alcuni esempi, consulta Aggiunta di un'opzione a un gruppo di opzioni.

Accesso all'OEM tramite il browser

Dopo avere abilitato l'opzione OEM, potrai iniziare a utilizzare lo strumento OEM Database dal browser Web.

Dal browser Web, puoi accedere a OEM Database Control oppure a OEM Database Express. Ad esempio, se l'endpoint dell'istanza database Amazon RDS è `mydb.f9rbfa893tft.us-east-1.rds.amazonaws.com` e la porta OEM è 1158, l'URL per accedere a OEM Database Control sarà quello indicato di seguito.

```
https://mydb.f9rbfa893tft.us-east-1.rds.amazonaws.com:1158/em
```

Quando accedi a uno di tali strumenti dal browser Web, viene visualizzata una finestra in cui ti viene chiesto di immettere il nome utente e la password. Immetti il nome utente master e la password master dell'istanza database. Ora puoi gestire gli Oracle Database.

Modifica delle impostazioni di OEM Database

Dopo avere abilitato OEM Database, puoi modificare l'impostazione dei gruppi di sicurezza dell'opzione.

Non è possibile modificare il numero di porta OEM dopo avere associato il gruppo di opzioni a un'istanza database. Per modificare il numero di porta OEM utilizzato per un'istanza database, procedi come indicato di seguito:

1. Crea un nuovo gruppo di opzioni.
2. Aggiungi l'opzione OEM con il numero di porta al nuovo gruppo di opzioni.
3. Rimuovi il gruppo di opzioni esistente dall'istanza database.

4. Aggiungi il nuovo gruppo di opzioni all'istanza database.

Per ulteriori informazioni su come modificare le impostazioni dell'opzione, consulta Modifica di un'impostazione di un'opzione. Per ulteriori informazioni su ciascuna impostazione, consulta Impostazioni dell'opzione OEM Database.

Esecuzione di attività OEM Database Express

Puoi utilizzare le procedure Amazon RDS per eseguire determinate attività OEM Database Express. Con queste procedure, puoi eseguire le attività elencate di seguito.

> **Note**
>
> Le attività OEM Database Express vengono eseguite in modo asincrono.

Attività

- Cambiare il front-end del sito Web per OEM Database Express in Adobe Flash
- Cambiare il front-end del sito Web per OEM Database Express in Oracle JET

Cambiare il front-end del sito Web per OEM Database Express in Adobe Flash

> **Note**
>
> Questa attività è disponibile solo per Oracle Database 19c non CDB.

A partire da Oracle Database 19c, Oracle ha dichiarato obsoleta la precedente interfaccia utente OEM Database Express, basata su Adobe Flash. OEM Database Express utilizza ora un'interfaccia creata con Oracle JET. In caso di problemi con la nuova interfaccia, puoi tornare all'interfaccia basata su Flash che è stata dichiarata obsoleta. I problemi che si possono verificare con la nuova interfaccia includono il restare bloccati in una schermata `Loading` dopo aver effettuato l'accesso a OEM Database Express. Potrebbero anche mancare alcune funzionalità importanti che erano presenti nella versione basata su Flash di OEM Database Express.

Per cambiare il front-end del sito Web di OEM Database Express in Adobe Flash, eseguire la procedura Amazon RDS `rdsadmin.rdsadmin_oem_tasks.em_express_frontend_to_flash`. Questa procedura equivale al comando SQL `execemx emx`.

Le best practice di sicurezza scoraggiano l'uso di Adobe Flash. Sebbene sia possibile ripristinare l'OEM Database Express basato su Flash, è consigliabile utilizzare i siti Web OEM Database Express basati su Jet, se possibile. Se ripristini l'utilizzo di Adobe Flash e desideri tornare a utilizzare Oracle JET, utilizza la procedura `rdsadmin.rdsadmin_oem_tasks.em_express_frontend_to_jet`. Dopo un aggiornamento del database Oracle, una versione più recente di Oracle JET potrebbe risolvere i problemi relativi a JET in OEM Database Express. Per ulteriori informazioni sul passaggio a Oracle JET, consulta Cambiare il front-end del sito Web per OEM Database Express in Oracle JET.

> **Note**
>
> L'esecuzione di questa attività dall'istanza database di origine per una replica di lettura ha anche come conseguenza che i front-end del sito Web di OEM Database Express cambiano in Adobe Flash.

La seguente invocazione di procedura crea un'attività per cambiare il sito Web di OEM Database Express in Adobe Flash e restituire l'ID dell'attività.

```
SELECT rdsadmin.rdsadmin_oem_tasks.em_express_frontend_to_flash() as TASK_ID from DUAL;
```

È possibile visualizzare il risultato visualizzando il file di output dell'attività.

```
SELECT text FROM table(rdsadmin.rds_file_util.read_text_file('BDUMP','dbtask-task-id.log'));
```

Sostituire *`task-id`* con l'ID attività restituito dalla procedura. Per ulteriori informazioni sulla procedura Amazon RDS `rdsadmin.rds_file_util.read_text_file`, consulta Lettura dei file in una directory di istanze database

Puoi inoltre visualizzare il contenuto del file di output dell'attività nella AWS Management Console ricercando le voci di log nella sezione Logs & events (Log ed eventi) per il `task-id`.

Cambiare il front-end del sito Web per OEM Database Express in Oracle JET

> **Note**
>
> Questa attività è disponibile solo per Oracle Database 19c non CDB.

Per cambiare il front-end del sito Web di OEM Database Express in Adobe Flash, eseguire la procedura Amazon RDS `rdsadmin.rdsadmin_oem_tasks.em_express_frontend_to_jet`. Questa procedura equivale al comando SQL `execemx omx`.

Per impostazione predefinita, i siti Web di OEM Database Express per le istanze database di Oracle che eseguono 19c o versioni successive utilizzano Oracle JET. Se è stata utilizzata la procedura `rdsadmin.rdsadmin_oem_tasks.em_express_frontend_to_flash` per cambiare il front-end del sito Web di OEM di Database Express in Adobe Flash, è possibile tornare a Oracle JET. Per fare ciò, utilizza la procedura `rdsadmin.rdsadmin_oem_tasks.em_express_frontend_to_jet`. Per ulteriori informazioni sul passaggio ad Adobe Flash, consulta Cambiare il front-end del sito Web per OEM Database Express in Adobe Flash.

> **Note**
>
> L'esecuzione di questa attività dall'istanza database di origine per una replica di lettura ha anche come conseguenza che i front-end del sito Web di OEM Database Express cambiano in Oracle JET.

La seguente invocazione di procedura crea un'attività per cambiare il sito Web di OEM Database Express in Oracle JET e restituire l'ID dell'attività.

```
SELECT rdsadmin.rdsadmin_oem_tasks.em_express_frontend_to_jet() as TASK_ID from DUAL;
```

È possibile visualizzare il risultato visualizzando il file di output dell'attività.

```
SELECT text FROM table(rdsadmin.rds_file_util.read_text_file('BDUMP','dbtask-task-id.log'));
```

Sostituire *`task-id`* con l'ID attività restituito dalla procedura. Per ulteriori informazioni sulla procedura Amazon RDS `rdsadmin.rds_file_util.read_text_file`, consulta Lettura dei file in una directory di istanze database

Puoi inoltre visualizzare il contenuto del file di output dell'attività nella AWS Management Console ricercando le voci di log nella sezione Logs & events (Log ed eventi) per il `task-id`.

Rimozione dell'opzione OEM Database

Puoi rimuovere l'opzione OEM da un'istanza database. Quando rimuovi l'opzione OEM da un'istanza database Oracle Database 12c o successivo, si verifica una breve interruzione durante il riavvio

automatico dell'istanza. Quindi, una volta rimossa l'opzione OEM, non è necessario riavviare l'istanza database.

Per rimuovere l'opzione OEM dall'istanza database, procedi in uno dei seguenti modi:

- Rimuovi l'opzione OEM dal gruppo di opzioni a cui appartiene. Questa modifica coinvolge tutte le istanze database che usano il gruppo di opzioni. Per ulteriori informazioni, consulta Rimozione di un'opzione da un gruppo di opzioni.
- Modifica l'istanza database e specifica un diverso gruppo di opzioni che non comprenda l'opzione OEM. Questa modifica coinvolge una singola istanza database. È possibile specificare il gruppo di opzioni predefinito (vuoto) o un gruppo di opzioni personalizzato diverso. Per ulteriori informazioni, consulta Modifica di un'istanza database Amazon RDS.

Oracle Management Agent per Enterprise Manager Cloud Control

Oracle Management Agent (OEM) è un componente software che monitora destinazioni in esecuzione su host e comunica tali informazioni a Oracle Management Service (OMS) di livello intermedio. Per ulteriori informazioni, consulta Overview of Oracle Enterprise Manager Cloud Control 12c e Overview of Oracle Enterprise Manager Cloud Control 13c nella documentazione Oracle.

Amazon RDS supporta Management Agent tramite l'utilizzo dell'opzione `OEM_AGENT`. Il Management Agent richiede un'istanza database Amazon RDS che esegue una delle versioni seguenti:

- Oracle Database 19c (19.0.0.0) utilizzando l'architettura non CDB
- Oracle Database 12c Release 2 (12.2.0.1)
- Oracle Database 12c Release 1 (12.1.0.2)

Amazon RDS supporta Management Agent per le seguenti versioni di OEM:

- Oracle Enterprise Manager Cloud Control per 13c
- Oracle Enterprise Manager Cloud Control per 12c

Argomenti

Prerequisiti per Management Agent

Per utilizzare Management Agent, assicurati di soddisfare i seguenti prerequisiti.

Prerequisiti generali

Di seguito sono riportati i prerequisiti per l'utilizzo di Management Agent:

- Un Oracle Management Service (OMS), configurato per effettuare la connessione all'istanza database Amazon RDS.
- Nella maggior parte dei casi, è necessario configurare il VPC per permettere le connessioni dal tuo OMS all'istanza database. Se non hai familiarità con Amazon Virtual Private Cloud (Amazon VPC), completa le fasi in Tutorial: Creazione di un Amazon VPC da utilizzare con un'istanza database (solo IPv4) prima di continuare.
- Management Agent versione 13.5.0.0.v1 richiede OMS versione 13.5.0.0 o successive.
- Management Agent versione 13.4.0.9.v1 richiede OMS versione 13.4.0.9 o successiva e la patch 32198287.
- Assicurati di disporre di spazio di storage sufficiente per la release OEM:
  - Almeno 8,5 GiB per OEM 13c Release 5
  - Almeno 8,5 GiB per OEM 13c Release 4
  - Almeno 8,5 GiB per OEM 13c Release 3
  - Almeno 5,5 GiB per OEM 13c Release 2
  - Almeno 4,5 GiB per OEM 13c Release 1
  - Almeno 2,5 GiB per OEM 12c
- Se stai usando le versioni Management Agent `OEM_AGENT 13.2.0.0.v3` e `13.3.0.0.v2` e desideri usare la connettività TCPS, segui le istruzioni riportate in Configurazione dei certificati CA di terze parti per la comunicazione con i database target nella documentazione Oracle. Inoltre, aggiorna JDK su OMS seguendo le istruzioni indicate nel documento Oracle contrassegnato dal numero identificativo 2241358.1. Quest'operazione consente a OMS di supportare tutte le suite di cifratura supportate dal database.

> **Note**
>
> La connettività TCPS tra il Management Agent e l'istanza database è supportata solo per Management Agent `OEM_AGENT 13.2.0.0.v3`, `13.3.0.0.v2`, `13.4.0.9.v1` e versioni successive.

Prerequisiti delle versioni di Oracle Database

Di seguito sono indicate le versioni di Oracle Database supportate per ogni versione di Management Agent.

Versione Management Agent	Oracle Database 19c utilizzando l'architettura non CDB	Oracle Database 12c Release 2 (12.2)	Oracle Database 12c Release 1 (12.1)
13.5.0.0.v1	Supportato	Supportato	Supportato
13.4.0.9.v1	Supportato	Supportato	Supportato
13.3.0.0.v2	Supportato	Supportato	Supportato
13.3.0.0.v1	Supportato	Supportato	Supportato
13.2.0.0.v3	Supportato	Supportato	Supportato
13.2.0.0.v2	Supportato	Supportato	Supportato
13.2.0.0.v1	Supportato	Supportato	Supportato
13.1.0.0.v1	Supportato	Supportato	Supportato
12.1.0.5.v1	Non supportato	Supportato	Supportato
12.1.0.4.v1	Non supportato	Supportato	Supportato

Di seguito sono riportati i prerequisiti per le diverse versioni del database:

- Per una istanza database Amazon RDS che esegue Oracle Database 19c (19.0.0.0), il `AGENT_VERSION` minimo è 13.1.0.0.v1.
- Per un'istanza database Amazon RDS che esegue Oracle Database Release 2 (12.2.0.1) o precedente, è necessario che vengano soddisfatti i seguenti requisiti:
  - Per OMS 13c Release 2 con Oracle patch 25163555 applicata, usare OEM Agent 13.2.0.0.v2 o versione successiva.

    Utilizzo di OMSPatcher per applicare la patch.
  - Per OMS 13c Release 2 senza la patch, utilizzare OEM Agent 13.2.0.0.v1.

  Utilizzo di OMSPatcher per applicare la patch.

Prerequisiti di comunicazione host OMS

Assicurati che il tuo host OMS e l'istanza database Amazon RDS possano comunicare. Esegui questa operazione:

- Per effettuare la connessione dal Management Agent al tuo OMS, se l'OMS è protetto da un firewall, aggiungi gli indirizzi IP delle istanze database all'OMS.

  Verifica che il firewall per OMS consente il traffico dalla porta del listener DB (predefinita 1521) e la porta OEM Agent (predefinita 3872), che ha origine dall'indirizzo IP dell'istanza database.
- Per effettuare la connessione dall'OMS al Management Agent, se l'OMS ha un nome host risolvibile pubblicamente, aggiungi l'indirizzo dell'OMS a un gruppo di sicurezza. Il gruppo di sicurezza deve avere regole in entrata che permettono l'accesso alla porta del listener di database e a quella del Management Agent. Per un esempio su come creare un gruppo di sicurezza e aggiungere regole in entrata, consulta Tutorial: Creazione di un Amazon VPC da utilizzare con un'istanza database (solo IPv4).
- Per effettuare la connessione dall'OMS al Management Agent, se l'OMS non ha un nome host risolvibile pubblicamente, utilizza uno dei seguenti:
  - Se OMS si trova in un'istanza Amazon Elastic Compute Cloud (Amazon EC2) in un VPC privato, è possibile configurare il peering VPC per effettuare la connessione da OMS a Management Agent. Per ulteriori informazioni, consulta Un'istanza database in un VPC a cui accede un'istanza EC2 in un VPC diverso.
  - Se l'OMS è ospitato in locale, è possibile configurare una connessione VPN per permettere l'accesso dall'OMS al Management Agent. Per ulteriori informazioni, consulta l'articolo relativo a Un'istanza database in un VPC a cui accede un'applicazione client tramite Internet o alle connessioni VPN.

Limitazioni per Management Agent

Di seguito sono riportate alcune delle limitazioni all'utilizzo di Management Agent:

- Non è possibile fornire immagini personalizzate di Oracle Management Agent.
- Le attività amministrative come l'esecuzione dei processi e l'applicazione di patch ai database, che richiedono credenziali host, non sono supportate.
- Non è garantito che i parametri host e l'elenco di processi riflettano lo stato reale del sistema. Pertanto, non è consigliabile utilizzare OEM per monitorare il file system radice o il file system

del punto di montaggio. Per ulteriori informazioni sul monitoraggio del sistema operativo, vedere Monitoraggio dei parametri del sistema operativo con il monitoraggio avanzato.

- L'individuazione automatica non è supportata. È necessario aggiungere manualmente le destinazioni dei database.
- La disponibilità dei moduli OMS dipende dall'edizione del database. Ad esempio, la diagnosi delle prestazioni del database e il modulo di ottimizzazione sono disponibili solo per Oracle Database Enterprise Edition.
- Management Agent consuma memoria e risorse di calcolo ulteriori. Se hai problemi di prestazioni dopo aver abilitato l'opzione `OEM_AGENT`, ti consigliamo di aumentare a una classe di istanza database più ampia. Per ulteriori informazioni, consulta Classi di istanze database e Modifica di un'istanza database Amazon RDS.
- L'utente che esegue il comando `OEM_AGENT` sull'host Amazon RDS non dispone dell'accesso al sistema operativo al log di avvisi. Pertanto, non è possibile raccogliere parametri per `DB Alert Log` e `DB Alert Log Error Status` in OEM.

Impostazioni dell'opzione per Management Agent

Amazon RDS supporta le seguenti impostazioni per l'opzione Management Agent.

Impostazione opzioni	Campo obbligatorio	Valori validi	Descrizione
Version (`AGENT_VERSION`)	Sì	`13.5.0.0.v1` `13.4.0.9.v1` `13.3.0.0.v2` `13.3.0.0.v1` `13.2.0.0.v3`	La versione del software Management Agent. Il nome dell'opzione AWS CLI è `OptionVersion`. **Note** Nelle regioni AWS GovCloud (US), le versioni 12.1 e 13.1 non sono disponibili.

Impostazione opzioni	Campo obbligatorio	Valori validi	Descrizione
		`13.2.0.0.v2` `13.2.0.0.v1` `13.1.0.0.v1` `12.1.0.5.v1` `12.1.0.4.v1`	
Porta (`AGENT_PORT`)	Sì	Un valore intero.	La porta dell'istanza database in ascolto dell'host OMS. Il valore predefinito è 3872. L'host OMS deve appartenere a un gruppo di sicurezza che ha accesso a questa porta. Il nome dell'opzione AWS CLI è `Port`.
Gruppi di sicurezza	Sì	Gruppi di sicurezza esistenti	Un gruppo di sicurezza che ha accesso a Port (Porta). L'host OMS deve appartenere a questo gruppo di sicurezza. Il nome dell'opzione AWS CLI è `VpcSecurityGroupMemberships` o `DBSecurityGroupMemberships`.

Impostazione opzioni	Campo obbligatorio	Valori validi	Descrizione
OMS_HOST	Sì	Un valore di stringa, ad esempio *my.example.oms*	Il nome host accessibile pubblicamente o l'indirizzo IP dell'OMS. Il nome dell'opzione AWS CLI è `OMS_HOST`.
OMS_PORT	Sì	Un valore intero.	La porta di caricamento HTTPS sull'host OMS che è in ascolto del Management Agent. Per determinare la porta di caricamento HTTPS, effettuare la connessione all'host OMS ed eseguire il seguente comando (che richiede la password di `SYSMAN`): `emctl status oms -details` Il nome dell'opzione AWS CLI è `OMS_PORT`.
AGENT_REGISTRATION_PASSWORD	Sì	Un valore di stringa.	La password utilizzata dal Management Agent per effettuare l'autenticazione con l'OMS. Consigliamo di creare una password persistente sull'OMS prima di abilitare l'opzione `OEM_AGENT` . Con una password persistente, è possibile condividere un singolo gruppo di opzioni di Management Agent con più database Amazon RDS. Il nome dell'opzione AWS CLI è `AGENT_REGISTRATION_PASSWORD` .

Impostazione opzioni	Campo obbligatorio	Valori validi	Descrizione
ALLOW_TLS_ONLY	No	`true`, `false` (default)	Un valore che configura l'agente OEM per supportare solo il protocollo `TLSv1` mentre l'agente è in ascolto come un server. Questa impostazione è supportata solo per le versioni di agente 12.1. Le versioni di agente successive supportano solo Transport Layer Security (TLS) per impostazione predefinita.
MINIMUM_TLS_VERSION	No	`TLSv1` (default), `TLSv1.2`	Una valore che specifica la versione TLS minima supportata dall'agente OEM mentre l'agente è in ascolto come un server. Questa impostazione è supportata solo per le versioni dell'agente 13.1.0.0.v1 e successive. Le versioni dell'agente precedenti supportano solo l'impostazione `TLSv1`.
TLS_CIPHER_SUITE	No	Per informazioni, consulta Impostazioni TLS dell'opzione Management Agent.	Un valore che specifica la suite di cifratura TLS utilizzata dall'agente OEM mentre l'agente è in ascolto come un server.

La tabella seguente elenca le suite di crittografia TLS supportate dall'opzione Management Agent.

Impostazioni TLS dell'opzione Management Agent

Suite di cifratura	Versione Agent supportata	Conformità agli standard FedRAMP
TLS_RSA_WITH_AES_128_CBC_SHA	Tutti	No
TLS_RSA_WITH_AES_128_CBC_SHA256	13.1.0.0.v1 o versione successiva	No
TLS_RSA_WITH_AES_256_CBC_SHA	13.2.0.0.v3 o versione successiva	No
TLS_RSA_WITH_AES_256_CBC_SHA256	13.2.0.0.v3 o versione successiva	No
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	13.2.0.0.v3 o versione successiva	Sì
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	13.2.0.0.v3 o versione successiva	Sì
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256	13.2.0.0.v3 o versione successiva	Sì
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384	13.2.0.0.v3 o versione successiva	Sì

Aggiunta dell'opzione Management Agent

Il processo generale per aggiungere l'opzione Management Agent a un'istanza database è il seguente:

1. Creare un nuovo gruppo di opzioni oppure copiare o modificare un gruppo di opzioni esistente.
2. Aggiungere l'opzione al gruppo di opzioni.
3. Associare il gruppo di opzioni a questa istanza database.

Se riscontri errori, puoi consultare i documenti di My Oracle Support per informazioni sulla risoluzione di problemi specifici.

Dopo aver aggiunto l'opzione Management Agent, non è necessario riavviare l'istanza database. Non appena il gruppo di opzioni è attivo, è attivo anche l'agente OEM.

Se l'host OMS utilizza un certificato di terze parti non attendibile, Amazon RDS restituisce il seguente errore.

```
You successfully installed the OEM_AGENT option. Your OMS host is using an untrusted
 third party certificate.
Configure your OMS host with the trusted certificates from your third party.
```

Se viene restituito questo errore, l'opzione Management Agent non è abilitata finché il problema non viene corretto. Per informazioni sulla correzione del problema, consulta il documento di supporto My Oracle 2202569.1.

Console

Per aggiungere l'opzione Management Agent a un'istanza database

1. Determinare il gruppo di opzioni che si desidera usare. È possibile creare un nuovo gruppo di opzioni oppure usare un gruppo di opzioni esistente. Se si desidera usare un gruppo di opzioni esistente, puoi passare alla fase successiva. In caso contrario, creare un gruppo di opzioni database personalizzato con le seguenti impostazioni:

   a. Per Engine (Motore) scegliere l'edizione Oracle per l'istanza database.

   b. In Major engine version (Versione principale del motore), scegliere la versione dell'istanza database.

   Per ulteriori informazioni, consulta Creazione di un gruppo di opzioni.

2. Aggiungere l'opzione OEM_AGENT (OEM_AGENT) al gruppo di opzioni e configurare le impostazioni di opzione. Per ulteriori informazioni sull'aggiunta di opzioni, consulta Aggiunta di un'opzione a un gruppo di opzioni. Per ulteriori informazioni su ciascuna impostazione, consulta Impostazioni dell'opzione per Management Agent.

3. Applicare il gruppo di opzioni a un'istanza database nuova o esistente:

   - Per una nuova istanza database, si applica il gruppo di opzioni quando viene avviata l'istanza. Per ulteriori informazioni, consulta Creazione di un'istanza database Amazon RDS.

- Per un'istanza database esistente, viene applicato il gruppo di opzioni modificando l'istanza e collegando il nuovo gruppo di opzioni. Per ulteriori informazioni, consulta Modifica di un'istanza database Amazon RDS.

AWS CLI

L'esempio seguente utilizza il comando AWS CLI add-option-to-option-group per aggiungere l'`OEM_AGENT`opzione a un gruppo di opzioni chiamato`myoptiongroup`.

Per LinuxmacOS, oUnix:

```
aws rds add-option-to-option-group \
    --option-group-name "myoptiongroup" \
    --options
 OptionName=OEM_AGENT,OptionVersion=13.1.0.0.v1,Port=3872,VpcSecurityGroupMemberships=sg-123456
{Name=OMS_PORT,Value=4903},{Name=AGENT_REGISTRATION_PASSWORD,Value=password}] \
    --apply-immediately
```

Per Windows:

```
aws rds add-option-to-option-group ^
    --option-group-name "myoptiongroup" ^
    --options
 OptionName=OEM_AGENT,OptionVersion=13.1.0.0.v1,Port=3872,VpcSecurityGroupMemberships=sg-123456
{Name=OMS_PORT,Value=4903},{Name=AGENT_REGISTRATION_PASSWORD,Value=password}] ^
    --apply-immediately
```

Utilizzo del Management Agent

Dopo aver abilitato l'opzione Management Agent, segui questa procedura per iniziare a utilizzarla.

Per utilizzare il Management Agent

1. Sbloccare e reimpostare le credenziali dell'account DBSNMP. Per farlo, eseguire questo codice nel database di destinazione sull'istanza database e utilizzare l'account utente master.

   ```
   ALTER USER dbsnmp IDENTIFIED BY new_password ACCOUNT UNLOCK;
   ```

2. Aggiunta manuale delle destinazioni alla console OMS:

a. Nella console OMS, selezionare Setup (Configurazione), Add Target (Aggiungi destinazione), Add Targets Manually (Aggiungi manualmente destinazioni).

b. Scegliere Add Targets Declaratively by Specifying Target Monitoring Properties (Aggiungi destinazioni in modo dichiarativo specificando le proprietà di monitoraggio della destinazione).

c. Per Target Type (Tipo di destinazione), scegliere Database Instance (Istanza database).

d. Per Monitoring Agent (Agente di monitoraggio) scegliere l'agente con l'ID uguale all'identificatore istanze DB di RDS.

e. Scegliere Add Manually (Aggiungi manualmente).

f. Inserire l'endpoint per l'istanza database Amazon RDS o selezionarlo dall'elenco di nomi host. Verificare che il nome dell'host specificato corrisponda all'endpoint dell'istanza database Amazon RDS.

   Per informazioni su come trovare l'endpoint per l'istanza database Amazon RDS, consultare Esito dell'endpoint dell'istanza database RDS per Oracle.

g. Specificare le seguenti proprietà del database:

   - Immettere un nome in Target Name (Nome destinazione).
   - Immettere un nome in Database system name (Nome del sistema di database).
   - Immettere **`dbsnmp`** in Monitor username (Nome utente monitoraggio).
   - In Monitor password (Password monitoraggio) immettere la password della Fase 1.
   - In Role (Ruolo) immettere normal (normale).
   - In Oracle home path (Percorso home di Oracle) immettere **`/oracle`**.
   - Per Listener Machine name (Nome macchina listener), viene già visualizzato l'identificatore dell'agente.
   - In Port (Porta) immettere la porta del database. La porta RDS predefinita è 1521.
   - In Database name (Nome del database) immettere il nome del database.

h. Scegliere Test Connection (Connessione di prova).

i. Seleziona Avanti. Il database di destinazione viene visualizzato nell'elenco di risorse monitorate.

Modifica delle impostazioni del Management Agent

Dopo aver abilitato il Management Agent, è possibile modificare le impostazioni per l'opzione. Per ulteriori informazioni su come modificare le impostazioni dell'opzione, consulta Modifica di un'impostazione di un'opzione. Per ulteriori informazioni su ciascuna impostazione, consulta Impostazioni dell'opzione per Management Agent.

Esecuzione delle attività di database con Management Agent

Puoi utilizzare le procedure Amazon RDS per eseguire alcuni comandi EMCTL in Management Agent. Con queste procedure, puoi eseguire le attività elencate di seguito.

> **Note**
>
> Le attività vengono eseguite in modo asincrono.

Attività

- Ottenere lo stato del Management Agent
- Riavvio di Management Agent
- Elenco delle destinazioni monitorate da Management Agent
- Elenco dei thread di raccolta monitorati da Management Agent
- Cancellazione dello stato di Management Agent
- Configurazione di Management Agent per il caricamento automatico del proprio OMS
- Ping dell'OMS
- Visualizzazione dello stato di un'attività in corso

Ottenere lo stato del Management Agent

Per ottenere lo stato del Management Agent, eseguire la Amazon RDS procedura `rdsadmin.rdsadmin_oem_agent_tasks.get_status_oem_agent`. Questa procedura equivale al comando `emctl status agent`.

La procedura seguente crea un'attività per ottenere lo stato del Management Agent e restituisce l'ID dell'attività.

```
SELECT rdsadmin.rdsadmin_oem_agent_tasks.get_status_oem_agent() as TASK_ID from DUAL;
```

Per visualizzare il risultato visualizzando il file di output dell'attività, consulta Visualizzazione dello stato di un'attività in corso.

Riavvio di Management Agent

Per riavviare Management Agent, esegui la procedura di Amazon RDS `rdsadmin.rdsadmin_oem_agent_tasks.restart_oem_agent`. Questa procedura equivale ai comandi `emctl stop agent` e `emctl start agent`.

La procedura seguente crea un'attività per riavviare Management Agent e restituisce l'ID dell'attività.

```
SELECT rdsadmin.rdsadmin_oem_agent_tasks.restart_oem_agent as TASK_ID from DUAL;
```

Per visualizzare il risultato visualizzando il file di output dell'attività, consulta Visualizzazione dello stato di un'attività in corso.

Elenco delle destinazioni monitorate da Management Agent

Per elencare le destinazioni monitorate da Management Agent, esegui la procedura di Amazon RDS `rdsadmin.rdsadmin_oem_agent_tasks.list_targets_oem_agent`. Questa procedura equivale al comando `emctl config agent listtargets`.

La procedura seguente crea un'attività per elencare le destinazioni monitorate dal Management Agent e restituisce l'ID dell'attività.

```
SELECT rdsadmin.rdsadmin_oem_agent_tasks.list_targets_oem_agent as TASK_ID from DUAL;
```

Per visualizzare il risultato visualizzando il file di output dell'attività, consulta Visualizzazione dello stato di un'attività in corso.

Elenco dei thread di raccolta monitorati da Management Agent

Per elencare tutti i thread di raccolta in esecuzione, pronti e pianificati monitorati dal Management Agent, esegui la procedura Amazon RDS `rdsadmin.rdsadmin_oem_agent_tasks.list_clxn_threads_oem_agent`. Questa procedura equivale al comando `emctl status agent scheduler`.

La procedura seguente crea un'attività per elencare i thread di raccolta e restituisce l'ID dell'attività.

```
SELECT rdsadmin.rdsadmin_oem_agent_tasks.list_clxn_threads_oem_agent() as TASK_ID from
 DUAL;
```

Per visualizzare il risultato visualizzando il file di output dell'attività, consulta Visualizzazione dello stato di un'attività in corso.

Cancellazione dello stato di Management Agent

Per cancellare lo stato di Management Agent, esegui la procedura di Amazon RDS `rdsadmin.rdsadmin_oem_agent_tasks.clearstate_oem_agent`. Questa procedura equivale al comando `emctl clearstate agent`.

La procedura seguente crea un'attività che cancella lo stato del Management Agent e restituisce l'ID dell'attività.

```
SELECT rdsadmin.rdsadmin_oem_agent_tasks.clearstate_oem_agent() as TASK_ID from DUAL;
```

Per visualizzare il risultato visualizzando il file di output dell'attività, consulta Visualizzazione dello stato di un'attività in corso.

Configurazione di Management Agent per il caricamento automatico del proprio OMS

Per configurare Management Agent in modo che carichi Oracle Management Server (OMS) a cui è associato, esegui la procedura di Amazon RDS `rdsadmin.rdsadmin_oem_agent_tasks.upload_oem_agent`. Questa procedura equivale al comando `emclt upload agent`.

Nella procedura seguente viene creata un'attività in cui Management Agent carica l'OMS associato e restituisce l'ID dell'attività.

```
SELECT rdsadmin.rdsadmin_oem_agent_tasks.upload_oem_agent() as TASK_ID from DUAL;
```

Per visualizzare il risultato visualizzando il file di output dell'attività, consulta Visualizzazione dello stato di un'attività in corso.

Ping dell'OMS

Per eseguire il ping dell'OMS di Management Agent, esegui la procedura di Amazon RDS `rdsadmin.rdsadmin_oem_agent_tasks.ping_oms_oem_agent`. Questa procedura equivale al comando `emctl pingOMS`.

Nella procedura seguente viene creata un'attività che esegue il ping dell'OMS del Management Agent e restituisce l'ID dell'attività.

```
SELECT rdsadmin.rdsadmin_oem_agent_tasks.ping_oms_oem_agent() as TASK_ID from DUAL;
```

Per visualizzare il risultato visualizzando il file di output dell'attività, consulta Visualizzazione dello stato di un'attività in corso.

Visualizzazione dello stato di un'attività in corso

È possibile visualizzare lo stato di un'attività in corso in un file bdump. I file bdump si trovano nella directory `/rdsdbdata/log/trace`. Il nome del file bdump ha il formato che segue.

```
dbtask-task-id.log
```

Per monitorare un'attività, sostituisci *`task-id`* con l'ID di tale attività.

Per visualizzare i contenuti dei file bdump, esegui la procedura di Amazon RDS `rdsadmin.rds_file_util.read_text_file`. La query seguente restituisce i contenuti del file bdump `dbtask-1546988886389-2444.log`.

```
SELECT text FROM
 table(rdsadmin.rds_file_util.read_text_file('BDUMP','dbtask-1546988886389-2444.log'));
```

Per ulteriori informazioni sulla procedura `rdsadmin.rds_file_util.read_text_file` di Amazon RDS, consulta Lettura dei file in una directory di istanze database.

Rimozione dell'opzione Management Agent

È possibile rimuovere l'OEM Agent dall'istanza database. Dopo aver rimosso l'opzione OEM Agent, non è necessario riavviare la tua istanza database.

Per rimuovere l'OEM Agent dall'istanza database, procedere in uno dei seguenti modi:

- Rimuovere l'opzione OEM Agent dal gruppo di opzioni a cui appartiene. Questa modifica coinvolge tutte le istanze database che usano il gruppo di opzioni. Per ulteriori informazioni, consulta Rimozione di un'opzione da un gruppo di opzioni.
- Modificare l'istanza database e specificare un diverso gruppo di opzioni che non comprenda l'opzione OEM Agent. Questa modifica coinvolge una singola istanza database. È possibile

specificare il gruppo di opzioni predefinito (vuoto) o un gruppo di opzioni personalizzato diverso. Per ulteriori informazioni, consulta Modifica di un'istanza database Amazon RDS.

Oracle Label Security

Amazon RDS supporta Oracle Label Security per la Enterprise Edition di Oracle Database tramite l'uso dell'opzione OLS.

La maggior parte della sicurezza del database controlla l'accesso a livello dell'oggetto. Oracle Label Security fornisce un controllo preciso dell'accesso alle singole righe della tabella. Ad esempio, è possibile usare Label Security per applicare la conformità alle normative con un modello di amministrazione basato sulla policy. È possibile usare le policy di Label Security per controllare l'accesso a dati sensibili e l'acceso limitato solo agli utenti con il livello adeguato di autorizzazioni. Per altre informazioni, consulta l'articolo relativo a Introduzione a Oracle Label Security nella documentazione Oracle.

Argomenti

Prerequisiti per Oracle Label Security

Acquisisci familiarità con i seguenti prerequisiti per Oracle Label Security:

- L'istanza database deve usare il modello Bring Your Own License (uso di licenze proprie). Per ulteriori informazioni, consulta Opzioni di licenza per RDS per Oracle.
- È necessario disporre di una licenza valida per Oracle Enterprise Edition con Licenza di aggiornamento software e supporto.
- La licenza Oracle deve comprendere l'opzione Label Security.
- È necessario utilizzare l'architettura di database non multi-tenant (non CDB). Per ulteriori informazioni, consulta Configurazione a tenant singolo dell'architettura CDB.

Aggiunta dell'opzione Oracle Label Security

La procedura generale per aggiungere l'opzione Label Security a un'istanza database è la seguente:

1. Creare un nuovo gruppo di opzioni oppure copiare o modificare un gruppo di opzioni esistente.
2. Aggiungere l'opzione al gruppo di opzioni.

> **Important**
>
> Oracle Label Security è un'opzione permanente e persistente.

3. Associare il gruppo di opzioni a questa istanza database.

Una volta aggiunta l'opzione Label Security, non appena il gruppo di opzioni sarà attivo, anche Label Security sarà attiva.

Per aggiungere l'opzione Label Security a un'istanza database

1. Determinare il gruppo di opzioni che si desidera usare. È possibile creare un nuovo gruppo di opzioni oppure usare un gruppo di opzioni esistente. Se si desidera usare un gruppo di opzioni esistente, puoi passare alla fase successiva. In caso contrario, creare un gruppo di opzioni database personalizzato con le seguenti impostazioni:

   a. In Engine (Motore) scegliere oracle-ee.

   b. In Major engine version (Versione principale del motore), scegliere la versione dell'istanza database.

   Per ulteriori informazioni, consulta Creazione di un gruppo di opzioni.

2. Aggiungere l'opzione OLS al gruppo di opzioni. Per ulteriori informazioni sull'aggiunta di opzioni, consulta Aggiunta di un'opzione a un gruppo di opzioni.

> **Important**
>
> Se aggiungi Label Security a un gruppo di opzioni esistente che è già associato a una o più istanze database, tutte le istanze database vengono riavviate.

3. Applicare il gruppo di opzioni a un'istanza database nuova o esistente:

   - Per una nuova istanza database, si applica il gruppo di opzioni quando viene avviata l'istanza. Per ulteriori informazioni, consulta Creazione di un'istanza database Amazon RDS.

- Per un'istanza database esistente, viene applicato il gruppo di opzioni modificando l'istanza e collegando il nuovo gruppo di opzioni. Quando aggiungi l'opzione Label Security a un'istanza database esistente, si verifica una breve interruzione mentre l'istanza database viene automaticamente riavviata. Per ulteriori informazioni, consulta Modifica di un'istanza database Amazon RDS.

Uso di Oracle Label Security

Per usare Oracle Label Security, è possibile creare policy che controllino l'accesso a righe specifiche nelle tabelle. Per altre informazioni, consulta l'articolo relativo a Creazione di una policy di Oracle Label Security nella documentazione Oracle.

Quando lavori con Label Security, esegui tutte le azioni come ruolo LBAC_DBA. All'utente master per l'istanza database è garantito il ruolo LBAC_DBA. È possibile garantire il ruolo LBAC_DBA ad altri utenti in modo che possano amministrare le policy di Label Security.

Per le versioni seguenti, assicurati di concedere l'accesso al pacchetto `OLS_ENFORCEMENT` per tutti i nuovi utenti che richiedono l'accesso a Oracle Label Security:

- Oracle Database 19c che utilizza l'architettura non CDB
- Oracle Database 12c Release 2 (12.2)

Per concedere l'accesso al pacchetto `OLS_ENFORCEMENT`, connettiti all'istanza database come utente master ed esegui l'istruzione SQL:

```
GRANT ALL ON LBACSYS.OLS_ENFORCEMENT TO username;
```

Puoi configurare Label Security tramite Oracle Enterprise Manager (OEM) Cloud Control. Amazon RDS supporta OEM Cloud Control tramite l'opzione Management Agent. Per ulteriori informazioni, consulta Oracle Management Agent per Enterprise Manager Cloud Control.

Rimozione dell'opzione Oracle Label Security (non supportata)

A partire da Oracle Database 12c Release 2 (12.2), Oracle Label Security è un'opzione permanente e persistente. Pertanto, non puoi rimuovere l'opzione da un gruppo di opzioni. Se aggiungi Oracle Label Security a un gruppo di opzioni e lo associ all'istanza database, è possibile in seguito associare un gruppo di opzioni diverso all'istanza database, ma anche questo gruppo deve contenere l'opzione Oracle Label Security.

Risoluzione dei problemi

Di seguito sono elencati i problemi che si potrebbero riscontrare quando si usa Oracle Label Security.

Problema	Suggerimenti sulla risoluzione dei problemi
Quando cerchi di creare una policy, comparirà un messaggio di errore simile al seguente: `insufficient authorization for the SYSDBA package`.	Un problema noto con la funzionalità Oracle Label Security impedisce agli utenti con nomi utente di 16 o 24 caratteri di eseguire i comandi di Label Security. È possibile creare un nuovo utente con un numero diverso di caratteri, garantire LBAC_DBA al nuovo utente, accedere come nuovo utente ed eseguire i comandi OLS come nuovo utente. Per altre informazioni in merito, contattare il supporto Oracle.

Oracle Locator

Amazon RDS supporta Oracle Locator tramite l'utilizzo dell'opzione `LOCATOR`. Oracle Locator offre funzionalità che sono solitamente richieste per il supporto di applicazioni Internet e wireless basate su servizi e di soluzioni GIS basate su partner. Oracle Locator è un sottoinsieme limitato di Oracle Spatial. Per ulteriori informazioni, consulta Oracle Locator nella documentazione di Oracle.

> **Important**
>
> Se utilizzi Oracle Locator, Amazon RDS aggiorna automaticamente la tua istanza database alla versione più recente di Oracle PSU, nel casi vi siano vulnerabilità della sicurezza con un punteggio del Common Vulnerability Scoring System (CVSS) di 9+ o altre vulnerabilità della sicurezza annunciate.

Amazon RDS supporta Oracle Locator per le seguenti versioni ed edizioni di Oracle Database:

- Oracle Database 19c (19.0.0.0)
- Oracle Database 12c Release 2 (12.2.0.1)
- Oracle Database 12c Release 1 (12.1), versione 12.1.0.2.v13 o successiva

Oracle Locator non è supportato per Oracle Database 21c, ma la sua funzionalità è disponibile nell'opzione Oracle Spatial. In precedenza, l'opzione Spatial richiedeva licenze aggiuntive. Oracle Locator rappresentava un sottoinsieme di funzionalità Oracle Spatial e non richiedeva licenze aggiuntive. Nel 2019, Oracle ha annunciato che tutte le funzionalità di Oracle Spatial erano incluse nelle licenze Enterprise Edition e Standard Edition 2 senza costi aggiuntivi. Di conseguenza, l'opzione Oracle Spatial non richiedeva più licenze aggiuntive.

A partire da Oracle Database 21c, l'opzione Oracle Locator non è più supportata. Per utilizzare le funzionalità di Oracle Locator in Oracle Database 21c, installare invece l'opzione Oracle Spatial. Per ulteriori informazioni, consulta Machine Learning, Spatial e Graph - Nessuna licenza richiesta! nel blog Oracle Database Insider.

Prerequisiti per Oracle Locator

Di seguito sono indicati i prerequisiti per l'utilizzo di Oracle Locator:

- L'istanza database deve appartenere a una classe sufficiente. Oracle Locator non è supportato per le classi di istanza database db.t3.micro o db.t3.small. Per ulteriori informazioni, consulta Classi di istanza RDS for Oracle.
- L'istanza database deve avere l'opzione Auto Minor Version Upgrade (Aggiornamento minore automatico della versione) abilitata. Questa opzione consente all'istanza database di ricevere automaticamente gli aggiornamenti secondari della versione del motore del database quando vengono resi disponibili ed è obbligatoria per tutte le opzioni che installano la Java Virtual Machine (JVM) Oracle. Amazon RDS utilizza questa opzione per aggiornare l'istanza database all'ultimo PSU (Patch Set Update) o RU (Release Update) di Oracle. Per ulteriori informazioni, consulta Modifica di un'istanza database Amazon RDS.

Best Practice per Oracle Locator

Di seguito sono indicate le best practice per l'utilizzo di Oracle Locator:

- Per la massima sicurezza, è necessario utilizzare l'opzione `LOCATOR` con Secure Sockets Layer (SSL). Per ulteriori informazioni, consulta Oracle Secure Sockets Layer.
- Configurare l'istanza database per limitare l'accesso all'istanza database. Per ulteriori informazioni, consulta Scenari per accedere a un'istanza database in un VPC e Uso di un'istanza database in un VPC.

Aggiunta dell'opzione Oracle Locator

Di seguito è riportato il processo generale per aggiungere l'opzione `LOCATOR` a un'istanza database:

1. Creare un nuovo gruppo di opzioni oppure copiare o modificare un gruppo di opzioni esistente.
2. Aggiungere l'opzione al gruppo di opzioni.
3. Associare il gruppo di opzioni a questa istanza database.

Se Oracle Java Virtual Machine non è installato nell'istanza database, durante l'aggiunta dell'opzione `LOCATOR` si verifica una breve interruzione. Se Oracle Java Virtual Machine è già installata nell'istanza database, non si verificherà alcuna interruzione. Una volta aggiunta l'opzione, non è necessario riavviare l'istanza database. Non appena il gruppo di opzioni è attivo, Oracle Locator è disponibile.

> **Note**
>
> Durante questa interruzione, le funzioni di verifica delle password vengono temporaneamente disabilitate. Durante l'interruzione si possono verificare eventi correlati alle funzioni di verifica delle password. Le funzioni di verifica delle password vengono riabilitate prima di rendere disponibile l'istanza database Oracle.

Per aggiungere l'opzione **LOCATOR** a un'istanza database

1. Determinare il gruppo di opzioni che si vuole usare. È possibile creare un nuovo gruppo di opzioni oppure usare un gruppo di opzioni esistente. Se si desidera usare un gruppo di opzioni esistente, puoi passare alla fase successiva. In caso contrario, creare un gruppo di opzioni database personalizzato con le seguenti impostazioni:

   a. In Engine (Motore) selezionare l'edizione Oracle per l'istanza database.

   b. In Major engine version (Versione principale del motore), scegliere la versione dell'istanza database.

   Per ulteriori informazioni, consulta Creazione di un gruppo di opzioni.
2. Aggiungere l'opzione LOCATOR al gruppo di opzioni. Per ulteriori informazioni sull'aggiunta di opzioni, consulta Aggiunta di un'opzione a un gruppo di opzioni.
3. Applicare il gruppo di opzioni a un'istanza database nuova o esistente:

   - Per una nuova istanza database, si applica il gruppo di opzioni quando viene avviata l'istanza. Per ulteriori informazioni, consulta Creazione di un'istanza database Amazon RDS.
   - Per un'istanza database esistente, viene applicato il gruppo di opzioni modificando l'istanza e collegando il nuovo gruppo di opzioni. Per ulteriori informazioni, consulta Modifica di un'istanza database Amazon RDS.

Utilizzo di Oracle Locator

L'opzione Oracle Locator può essere utilizzata immediatamente dopo l'abilitazione. Utilizza solo le caratteristiche di Oracle Locator. Non usare le caratteristiche di Oracle Spatial se non disponi di un'apposita licenza.

Per un elenco delle caratteristiche supportate da Oracle Locator, consulta Features Included with Locator nella documentazione di Oracle.

Per un elenco delle caratteristiche non supportate da Oracle Locator, consulta Features Not Included with Locator nella documentazione di Oracle.

Rimozione dell'opzione Oracle Locator

Dopo aver eliminato tutti gli oggetti che utilizzano i tipi di dati forniti dall'opzione LOCATOR, è possibile rimuovere l'opzione da un'istanza DB. Se Oracle Java Virtual Machine non è installata nell'istanza database, durante la rimozione dell'opzione LOCATOR si verifica una breve interruzione. Se Oracle Java Virtual Machine è già installata nell'istanza database, non si verificherà alcuna interruzione. Una volta rimossa l'opzione LOCATOR non è necessario riavviare la tua istanza database.

Per eliminare l'opzione **LOCATOR**

1. Eseguire il backup dei dati.

   > **Warning**
   >
   > Se l'istanza utilizza tipi di dati abilitati come parte dell'opzione e se si rimuove l'opzione LOCATOR, è possibile perdere i dati. Per ulteriori informazioni, consulta Backup, ripristino ed esportazione dei dati.

2. Verificare se gli oggetti esistenti fanno riferimento ai tipi di dati o alle feature dell'opzione LOCATOR.

   Se esistono opzioni LOCATOR, l'istanza può rimanere bloccata quando si applica il nuovo gruppo di opzioni che non dispone dell'opzione LOCATOR. È possibile identificare gli oggetti utilizzando le seguenti query:

```
SELECT OWNER, SEGMENT_NAME, TABLESPACE_NAME, BYTES/1024/1024 mbytes
FROM   DBA_SEGMENTS
WHERE  SEGMENT_TYPE LIKE '%TABLE%'
AND    (OWNER, SEGMENT_NAME) IN
       (SELECT DISTINCT OWNER, TABLE_NAME
        FROM   DBA_TAB_COLUMNS
        WHERE  DATA_TYPE='SDO_GEOMETRY'
        AND    OWNER <> 'MDSYS')
ORDER BY 1,2,3,4;
```

```
SELECT OWNER, TABLE_NAME, COLUMN_NAME
FROM   DBA_TAB_COLUMNS
WHERE  DATA_TYPE = 'SDO_GEOMETRY'
AND    OWNER <> 'MDSYS'
ORDER BY 1,2,3;
```

3. Eliminare gli oggetti che fanno riferimento ai tipi di dati o alle feature dell'opzione LOCATOR.
4. Scegliere una delle seguenti operazioni:

   - Rimuovere l'opzione LOCATOR dal gruppo di opzioni a cui appartiene. Questa modifica coinvolge tutte le istanze database che usano il gruppo di opzioni. Per ulteriori informazioni, consulta Rimozione di un'opzione da un gruppo di opzioni.
   - Modificare l'istanza database e specificare un diverso gruppo di opzioni che non comprenda l'opzione LOCATOR. Questa modifica coinvolge una singola istanza database. È possibile specificare il gruppo di opzioni predefinito (vuoto) o un gruppo di opzioni personalizzato diverso. Per ulteriori informazioni, consulta Modifica di un'istanza database Amazon RDS.

Oracle Multimedia

Amazon RDS supporta Oracle Multimedia tramite l'utilizzo dell'opzione `MULTIMEDIA`. Puoi utilizzare Oracle Multimedia per archiviare, gestire e recuperare immagini, audio, video e altri dati di supporti eterogenei. Per ulteriori informazioni, vedere Oracle Multimedia nella documentazione di Oracle.

> **Important**
>
> Se utilizzi Oracle Multimedia, Amazon RDS aggiorna automaticamente la tua istanza database alla versione più recente di Oracle PSU, nel casi vi siano vulnerabilità della sicurezza con un punteggio del Common Vulnerability Scoring System (CVSS) di 9+ o altre vulnerabilità della sicurezza annunciate.

Amazon RDS supporta Oracle Multimedia per tutte le edizioni delle seguenti versioni:

- Oracle Database 12c Release 2 (12.2)
- Oracle Database 12c Release 1 (12.1), versione 12.1.0.2.v13 o versioni successive

> **Note**
>
> Oracle non offre più supporto a Oracle Multimedia in Oracle Database 19c. Di conseguenza, Oracle Multimedia non è supportato per le istanze database di Oracle Database 19c. Per ulteriori informazioni, vedere Fine supporto Oracle Multimedia nella documentazione di Oracle.

Prerequisiti per Oracle Multimedia

Di seguito sono indicati i prerequisiti per l'utilizzo di Oracle Multimedia:

- L'istanza database deve appartenere a una classe sufficiente. Oracle Multimedia non è supportato per le classi di istanza database db.t3.micro o db.t3.small. Per ulteriori informazioni, consulta Classi di istanza RDS for Oracle.
- L'istanza database deve avere l'opzione Auto Minor Version Upgrade (Aggiornamento minore automatico della versione) abilitata. Questa opzione consente all'istanza database di ricevere automaticamente gli aggiornamenti secondari della versione del motore del database quando

vengono resi disponibili ed è obbligatoria per tutte le opzioni che installano la Java Virtual Machine (JVM) Oracle. Amazon RDS utilizza questa opzione per aggiornare l'istanza database all'ultimo PSU (Patch Set Update) o RU (Release Update) di Oracle. Per ulteriori informazioni, consulta Modifica di un'istanza database Amazon RDS.

Best practice per Oracle Multimedia

Di seguito sono indicate le best practice per l'utilizzo di Oracle Multimedia:

- Per la massima sicurezza, è necessario utilizzare l'opzione `MULTIMEDIA` con Secure Sockets Layer (SSL). Per ulteriori informazioni, consulta Oracle Secure Sockets Layer.
- Configurare l'istanza database per limitare l'accesso all'istanza database. Per ulteriori informazioni, consulta Scenari per accedere a un'istanza database in un VPC e Uso di un'istanza database in un VPC.

Aggiunta dell'opzione Oracle Multimedia

Di seguito è riportato il processo generale per aggiungere l'opzione `MULTIMEDIA` a un'istanza database:

1. Creare un nuovo gruppo di opzioni oppure copiare o modificare un gruppo di opzioni esistente.
2. Aggiungere l'opzione al gruppo di opzioni.
3. Associare il gruppo di opzioni a questa istanza database.

Se Oracle Java Virtual Machine non è installato nell'istanza database, durante l'aggiunta dell'opzione `MULTIMEDIA` si verifica una breve interruzione. Se Oracle Java Virtual Machine è già installata nell'istanza database, non si verificherà alcuna interruzione. Una volta aggiunta l'opzione, non è necessario riavviare l'istanza database. Non appena il gruppo di opzioni è attivo, Oracle Multimedia è disponibile.

> **Note**
>
> Durante questa interruzione, le funzioni di verifica delle password vengono temporaneamente disabilitate. Durante l'interruzione si possono verificare eventi correlati alle funzioni di verifica delle password. Le funzioni di verifica delle password vengono riabilitate prima di rendere disponibile l'istanza database Oracle.

Per aggiungere l'opzione **`MULTIMEDIA`** a un'istanza database

1. Determinare il gruppo di opzioni che si vuole usare. È possibile creare un nuovo gruppo di opzioni oppure usare un gruppo di opzioni esistente. Se si desidera usare un gruppo di opzioni esistente, puoi passare alla fase successiva. In caso contrario, creare un gruppo di opzioni database personalizzato con le seguenti impostazioni:

   a. Per Engine (Motore), scegliere l'edizione per l'istanza database Oracle.

   b. In Major engine version (Versione principale del motore), scegliere la versione dell'istanza database.

   Per ulteriori informazioni, consulta Creazione di un gruppo di opzioni.

2. Aggiungere l'opzione MULTIMEDIA al gruppo di opzioni. Per ulteriori informazioni sull'aggiunta di opzioni, consulta Aggiunta di un'opzione a un gruppo di opzioni.

3. Applicare il gruppo di opzioni a un'istanza database nuova o esistente:

   - Per una nuova istanza database, si applica il gruppo di opzioni quando viene avviata l'istanza. Per ulteriori informazioni, consulta Creazione di un'istanza database Amazon RDS.
   - Per un'istanza database esistente, viene applicato il gruppo di opzioni modificando l'istanza e collegando il nuovo gruppo di opzioni. Per ulteriori informazioni, consulta Modifica di un'istanza database Amazon RDS.

Rimozione dell'opzione Oracle Multimedia

Dopo aver eliminato tutti gli oggetti che utilizzano i tipi di dati forniti dall'opzione `MULTIMEDIA`, è possibile rimuovere l'opzione da un'istanza DB. Se Oracle Java Virtual Machine non è installata nell'istanza database, durante la rimozione dell'opzione `MULTIMEDIA` si verifica una breve interruzione. Se Oracle Java Virtual Machine è già installata nell'istanza database, non si verificherà alcuna interruzione. Una volta rimossa l'opzione `MULTIMEDIA` non è necessario riavviare la tua istanza database.

Per eliminare l'opzione **`MULTIMEDIA`**

1. Eseguire il backup dei dati.

> **Warning**
>
> Se l'istanza utilizza tipi di dati abilitati come parte dell'opzione e se si rimuove l'opzione `MULTIMEDIA`, è possibile perdere i dati. Per ulteriori informazioni, consulta Backup, ripristino ed esportazione dei dati.

2. Verificare se gli oggetti esistenti fanno riferimento ai tipi di dati o alle feature dell'opzione `MULTIMEDIA`.
3. Eliminare gli oggetti che fanno riferimento ai tipi di dati o alle feature dell'opzione `MULTIMEDIA`.
4. Scegliere una delle seguenti operazioni:

   - Rimuovere l'opzione `MULTIMEDIA` dal gruppo di opzioni a cui appartiene. Questa modifica coinvolge tutte le istanze database che usano il gruppo di opzioni. Per ulteriori informazioni, consulta Rimozione di un'opzione da un gruppo di opzioni.
   - Modificare l'istanza database e specificare un diverso gruppo di opzioni che non comprenda l'opzione `MULTIMEDIA`. Questa modifica coinvolge una singola istanza database. È possibile specificare il gruppo di opzioni predefinito (vuoto) o un gruppo di opzioni personalizzato diverso. Per ulteriori informazioni, consulta Modifica di un'istanza database Amazon RDS.

Oracle native network encryption

Amazon RDS supporta Native Network Encryption (NNE) di Oracle. Con la crittografia di rete nativa (NNE), puoi crittografare i dati quando vengono spostati da e verso un'istanza database. Amazon RDS supporta NNE per tutte le edizioni di Oracle Database.

Una descrizione dettagliata di NNE di Oracle non rientra nell'ambito di questa guida, ma è necessario conoscere i vantaggi e gli svantaggi di ogni algoritmo e chiave prima di scegliere una soluzione per la distribuzione. Per informazioni su algoritmi e chiavi disponibili mediante la crittografia di rete nativa di Oracle, consulta Configuring Network Data Encryption nella documentazione di Oracle. Per ulteriori informazioni sulla sicurezza AWS, consulta il Centro di Sicurezza AWS.

> **Note**
>
> Puoi utilizzare Secure Sockets Layer o Native Network Encryption, ma non entrambi. Per ulteriori informazioni, consulta Oracle Secure Sockets Layer.

Impostazioni dell'opzione NNE

È possibile specificare i requisiti di crittografia sia sul server che sul client. L'istanza database può fungere da client quando, ad esempio, utilizza un database link per connettersi a un altro database. È possibile evitare di forzare la crittografia sul lato server. Ad esempio, è possibile che non si desideri forzare tutte le comunicazioni client a utilizzare la crittografia perché il server lo richiede. In questo caso, è possibile forzare la crittografia sul lato client utilizzando le opzioni `SQLNET.*CLIENT`.

Amazon RDS supporta le seguenti impostazioni per l'opzione NNE.

> **Note**
>
> Quando si utilizzano virgole per separare i valori per un'impostazione di opzione, non inserire uno spazio dopo la virgola.

Impostazione opzioni	Valori validi	Valori predefiniti	Descrizione
SQLNET.ALLOW_WEAK_CRYPTO_CLIENTS	TRUE, FALSE	TRUE	Il comportamento del server quando un client che utilizza una crittografia non sicura tenta di connettersi al database. Se TRUE, i client possono connettersi anche se non sono stati sottoposti a patch con la PSU di luglio 2021. Se l'impostazione è FALSE, i client possono connettersi al database solo quando vengono sottoposti a patch con la PSU di luglio 2021. Prima di impostare SQLNET.ALLOW_WEAK_CRYPTO_CLIENTS su FALSE, assicurati che siano soddisfatte le seguenti condizioni: • SQLNET.ENCRYPTION_TYPES_SERVER e SQLNET.ENCRYPTION_TYPES_CLIENT hanno un metodo di crittografia corrispondente che non è DES, 3DES, oppure RC4 (tutte le lunghezze di chiave). • SQLNET.CHECKSUM_TYPES_SERVER e SQLNET.CHECKSUM_TYPES_CLIENT hanno un metodo di checksum sicuro corrispondente che non è MD5. • Il client viene sottoposto a patch con la PSU di luglio 2021. Se non è stato sottoposto a patch, il client

Impostazione opzioni	Valori validi	Valori predefiniti	Descrizione
			perde la connessione e riceve l'errore `ORA-12269` .

Impostazione opzioni	Valori validi	Valori predefiniti	Descrizione
SQLNET.ALLOW_WEAK_CRYPTO	TRUE, FALSE	TRUE	Il comportamento del server quando un client che utilizza una crittografia non sicura tenta di connettersi al database. Le seguenti crittografie sono considerate non sicure: • Metodo di crittografia DES (tutte le lunghezze di chiave) • Metodo di crittografia 3DES (tutte le lunghezze di chiave) • Metodo di crittografia RC4 (tutte le lunghezze di chiave) • Metodo di checksum MD5 Se l'impostazione è TRUE, i client possono connettersi quando utilizzano crittografie precedenti non sicure. Se l'impostazione è FALSE, il database impedisce ai client di connettersi quando utilizzano crittografie precedenti non sicure. Prima di impostare SQLNET.ALLOW_WEAK_CRYPTO su FALSE, assicurati che siano soddisfatte le seguenti condizioni: • SQLNET.ENCRYPTION_TYPES_SERVER e SQLNET.ENCRYPTION_TYPES_CLIENT hanno un metodo di crittografia corrispondente che non è DES, 3DES,

Impostazione opzioni	Valori validi	Valori predefiniti	Descrizione
			oppure RC4 (tutte le lunghezze di chiave). • SQLNET.CHECKSUM_TYPES_SERVER e SQLNET.CHECKSUM_TYPES_CLIENT hanno un metodo di checksum sicuro corrispondente che non è MD5. • Il client viene sottoposto a patch con la PSU di luglio 2021. Se non è stato sottoposto a patch, il client perde la connessione e riceve l'errore ORA-12269 .
SQLNET.CRYPTO_CHECKSUM_CLIENT	Accepted Rejected Requested , Required	Requested	Il comportamento dell'integrità dei dati quando un'istanza database si connette al client, oppure un server che agisce come client. Quando un'istanza database utilizza un collegamento di database, agisce come un client. Requested indica che il client non richiede che l'istanza database eseguire un checksum.

Impostazione opzioni	Valori validi	Valori predefiniti	Descrizione
SQLNET.CRYPTO_CHECKSUM_SERVER	Accepted Rejected Requested , Required	Requested	Il comportamento dell'integrità dei dati quando un client, oppure un server che agisce come client, si connette all'istanza database. Quando un'istanza database utilizza un collegamento di database, agisce come un client. Requested indica che l'istanza database non richiede al client di eseguire un checksum.
SQLNET.CRYPTO_CHECKSUM_TYPES_CLIENT	SHA256, SHA384, SHA512, SHA1, MD5	SHA256, SHA384, SHA512	Un elenco di algoritmi di checksum. È possibile specificare un valore o un elenco di valori separato da virgole. Se si utilizza una virgola, non inserire uno spazio dopo la virgola, altrimenti verrà visualizzato un errore InvalidParameterValue . Questo parametro e SQLNET.CRYPTO_CHECKSUM_TYPES_SERVER deve avere una crittografia comune.

Impostazione opzioni	Valori validi	Valori predefiniti	Descrizione
SQLNET.CRYPTO_CHECKSUM_TYPES_SERVER	SHA256, SHA384, SHA512, SHA1, MD5	SHA256, SHA384, SHA512, SHA1, MD5	Un elenco di algoritmi di checksum. È possibile specificare un valore o un elenco di valori separato da virgole. Se si utilizza una virgola, non inserire uno spazio dopo la virgola, altrimenti verrà visualizzato un errore InvalidParameterValue . Questo parametro e SQLNET.CRYPTO_CHECKSUM_TYPES_CLIENT deve avere una crittografia comune.
SQLNET.ENCRYPTION_CLIENT	Accepted Rejected Requested , Required	Requested	Il comportamento della crittografia del client quando un client, oppure un server che agisce come client, si connette all'istanza database. Quando un'istanza database utilizza un collegamento di database, agisce come un client. Requested indica che il client non richiede la crittografia del traffico dal server.

Impostazione opzioni	Valori validi	Valori predefiniti	Descrizione
SQLNET.ENCRYPTION_SERVER	Accepted Rejected Requested , Required	Requested	Il comportamento della crittografia del server quando un client, oppure un server che agisce come client, si connette all'istanza database. Quando un'istanza database utilizza un collegamento di database, agisce come un client. Requested indica che l'istanza database non richiede la crittografia del traffico dal client.

Impostazione opzioni	Valori validi	Valori predefiniti	Descrizione
SQLNET.ENCRYPTION_TYPES_CLIENT	RC4_256, AES256, AES192, 3DES168, RC4_128, AES128, 3DES112, RC4_56, DES, RC4_40, DES40	RC4_256, AES256, AES192, 3DES168, RC4_128, AES128, 3DES112, RC4_56, DES, RC4_40, DES40	Un elenco degli algoritmi di crittografia utilizzati dal client. Il client tenta di decrittografare l'input del server eseguendo ciascun algoritmo in sequenza, fino alla corretta esecuzione di un algoritmo o fino alla fine dell'elenco. Amazon RDS utilizza il seguente elenco predefinito di Oracle. RDS inizia con RC4_256 e procede in sequenza scorrendo l'elenco. Puoi cambiare l'ordine o limitare gli algoritmi che l'istanza database accetterà. 1. RC4_256: RSA RC4 (dimensione di chiave 256 bit) 2. AES256: AES (dimensione di chiave 256 bit) 3. AES192: AES (dimensione di chiave 192 bit) 4. 3DES168: 3-key Triple-DES (dimensione di chiave 112 bit) 5. RC4_128: RSA RC4 (dimensione di chiave 128 bit) 6. AES128: AES (dimensione di chiave 128 bit) 7. 3DES112: 2-key Triple-DES (dimensione di chiave 80 bit) 8. RC4_56: RSA RC4 (dimensione di chiave 56 bit)

Impostazione opzioni	Valori validi	Valori predefiniti	Descrizione
			9. `DES`: Standard DES (dimensione di chiave 56 bit) 10. `RC4_40`: RSA RC4 (dimensione di chiave 40 bit) 11. `DES40`: DES40 (dimensione di chiave 40 bit) È possibile specificare un valore o un elenco di valori separato da virgole. Se si utilizza una virgola, non inserire uno spazio dopo la virgola; in caso contrario, viene visualizzato un errore `InvalidParameterValue` . Questo parametro e `SQLNET.SQLNET.ENCRYPTION_TYPES_SERVER` deve avere una crittografia comune.

Impostazione opzioni	Valori validi	Valori predefiniti	Descrizione
SQLNET.ENCRYPTION_TYPES_SERVER	RC4_256, AES256, AES192, 3DES168, RC4_128, AES128, 3DES112, RC4_56, DES, RC4_40, DES40	RC4_256, AES256, AES192, 3DES168, RC4_128, AES128, 3DES112, RC4_56, DES, RC4_40, DES40	Un elenco degli algoritmi di crittografia utilizzati dall'istanza database. L'istanza database utilizza ogni algoritmo, nell'ordine, per tentare di decrittog rafare l'input del client fino alla corretta esecuzione di un algoritmo o fino alla fine dell'elenco. Amazon RDS utilizza il seguente elenco predefinito di Oracle. Puoi cambiare l'ordine o limitare gli algoritmi che il client accetterà. 1. RC4_256: RSA RC4 (dimensione di chiave 256 bit) 2. AES256: AES (dimensione di chiave 256 bit) 3. AES192: AES (dimensione di chiave 192 bit) 4. 3DES168: 3-key Triple-DES (dimensione di chiave 112 bit) 5. RC4_128: RSA RC4 (dimensione di chiave 128 bit) 6. AES128: AES (dimensione di chiave 128 bit) 7. 3DES112: 2-key Triple-DES (dimensione di chiave 80 bit) 8. RC4_56: RSA RC4 (dimensione di chiave 56 bit) 9. DES: Standard DES (dimensione di chiave 56 bit)

Impostazione opzioni	Valori validi	Valori predefiniti	Descrizione
			10`RC4_40`: RSA RC4 (dimensione di chiave 40 bit) 11`DES40`: DES40 (dimensione di chiave 40 bit) È possibile specificare un valore o un elenco di valori separato da virgole. Se si utilizza una virgola, non inserire uno spazio dopo la virgola; in caso contrario, viene visualizzato un errore `InvalidParameterValue` . Questo parametro e `SQLNET.SQLNET.ENCRYPTION_TYPES_SERVER` deve avere una crittografia comune.

Aggiunta dell'opzione NNE

Di seguito è riportata la procedura generale per aggiungere l'opzione NNE a un'istanza database:

1. Creare un nuovo gruppo di opzioni oppure copiare o modificare un gruppo di opzioni esistente.
2. Aggiungere l'opzione al gruppo di opzioni.
3. Associare il gruppo di opzioni a questa istanza database.

Quando il gruppo di opzioni è attivo, NNE è attiva.

Per aggiungere l'opzione NNE a un'istanza database utilizzando la AWS Management Console

1. Per Engine (Motore), scegliere l'edizione di Oracle da utilizzare. NNE è supportato in tutte le edizioni.

2. In Major engine version (Versione principale del motore), scegliere la versione dell'istanza database.

   Per ulteriori informazioni, consulta Creazione di un gruppo di opzioni.

3. Aggiungere l'opzione NNE al gruppo di opzioni. Per ulteriori informazioni sull'aggiunta di opzioni, consulta Aggiunta di un'opzione a un gruppo di opzioni.

   > **Note**
   >
   > Una volta aggiunta l'opzione NNE, non è necessario riavviare le istanze database. Non appena il gruppo di opzioni è attivo, NNE è attiva.

4. Applicare il gruppo di opzioni a un'istanza database nuova o esistente:

   - Per una nuova istanza database, si applica il gruppo di opzioni quando viene avviata l'istanza. Per ulteriori informazioni, consulta Creazione di un'istanza database Amazon RDS.
   - Per un'istanza database esistente, viene applicato il gruppo di opzioni modificando l'istanza e collegando il nuovo gruppo di opzioni. Una volta aggiunta l'opzione NNE, non è necessario riavviare l'istanza database. Non appena il gruppo di opzioni è attivo, NNE è attiva. Per ulteriori informazioni, consulta Modifica di un'istanza database Amazon RDS.

Impostazione dei valori NNE in sqlnet.ora

Con crittografia di rete nativa di Oracle, puoi impostare la crittografia di rete sul lato server e sul lato client. Il client è il computer utilizzato per connettersi all'istanza database. È possibile specificare le seguenti impostazioni client in slqnet.ora:

- `SQLNET.ALLOW_WEAK_CRYPTO`
- `SQLNET.ALLOW_WEAK_CRYPTO_CLIENTS`
- `SQLNET.CRYPTO_CHECKSUM_CLIENT`
- `SQLNET.CRYPTO_CHECKSUM_TYPES_CLIENT`
- `SQLNET.ENCRYPTION_CLIENT`
- `SQLNET.ENCRYPTION_TYPES_CLIENT`

Per informazioni, consulta Configuring Network Data Encryption and Integrity for Oracle Servers and Clients nella documentazione di Oracle.

Talvolta, l'istanza database rifiuta una richiesta di connessione da un'applicazione. Ad esempio, un rifiuto può verificarsi quando gli algoritmi di crittografia sul client e sul server non corrispondono. Per testare crittografia di rete nativa di Oracle, aggiungi le seguenti righe al file sqlnet.ora sul client:

```
DIAG_ADR_ENABLED=off
TRACE_DIRECTORY_CLIENT=/tmp
TRACE_FILE_CLIENT=nettrace
TRACE_LEVEL_CLIENT=16
```

Quando si esegue un tentativo di connessione, le righe precedenti generano un file di traccia sul client denominato `/tmp/nettrace*`. Il file di traccia contiene informazioni sulla connessione. Per ulteriori informazioni sui problemi relativi alla connessione quando si utilizza NNE, consulta About Negotiating Encryption and Integrity nella documentazione di Oracle Database.

Modifica delle impostazioni dell'opzione NNE

Dopo aver abilitato NNE, puoi modificarne le impostazioni. Attualmente, è possibile modificare le impostazioni dell'opzione NNE solo con la AWS CLI o l'API RDS. Non puoi utilizzare la console. Per informazioni su come modificare le impostazioni dell'opzione utilizzando la CLI, consulta AWS CLI. Per ulteriori informazioni su ciascuna impostazione, consulta Impostazioni dell'opzione NNE.

Argomenti

- Modifica dei valori CRYPTO_CHECKSUM_*
- Modifica delle impostazioni ALLOW_WEAK_CRYPTO*

Modifica dei valori CRYPTO_CHECKSUM_*

Se si modificano le impostazioni delle opzioni NNE, assicurati che le seguenti impostazioni delle opzioni abbiano almeno una crittografia comune:

- `SQLNET.CRYPTO_CHECKSUM_TYPES_SERVER`
- `SQLNET.CRYPTO_CHECKSUM_TYPES_CLIENT`

L'esempio seguente mostra uno scenario in cui si modifica `SQLNET.CRYPTO_CHECKSUM_TYPES_SERVER`. La configurazione è valida perché sia `CRYPTO_CHECKSUM_TYPES_CLIENT` che `CRYPTO_CHECKSUM_TYPES_SERVER` utilizzano `SHA256`.

Impostazione opzioni	Valori prima della modifica	Valori dopo la modifica
SQLNET.CRYPTO_CHECKSUM_TYPES_CLIENT	**SHA256**, SHA384, SHA512	Nessuna modifica
SQLNET.CRYPTO_CHECKSUM_TYPES_SERVER	**SHA256**, SHA384, SHA512, SHA1, MD5	SHA1,MD5, **SHA256**

Per un altro esempio, si supponga di voler modificare SQLNET.CRYPTO_CHECKSUM_TYPES_SERVER dalla sua impostazione di default a SHA1,MD5. In questo caso, assicurati di impostare SQLNET.CRYPTO_CHECKSUM_TYPES_CLIENT a SHA1 o MD5. Questi algoritmi non sono inclusi nei valori di default per SQLNET.CRYPTO_CHECKSUM_TYPES_CLIENT.

Modifica delle impostazioni ALLOW_WEAK_CRYPTO*

Per impostare le opzioni SQLNET.ALLOW_WEAK_CRYPTO* dal valore di default FALSE, accertati che siano soddisfatte le seguenti condizioni:

- SQLNET.ENCRYPTION_TYPES_SERVER e SQLNET.ENCRYPTION_TYPES_CLIENT hanno un metodo di crittografia sicuro corrispondente. Un metodo è considerato sicuro se non è DES, 3DES oppure RC4 (tutte le lunghezze di chiave).
- SQLNET.CHECKSUM_TYPES_SERVER e SQLNET.CHECKSUM_TYPES_CLIENT hanno un metodo di checksum sicuro corrispondente. Un metodo è considerato sicuro se non è MD5.
- Il client viene sottoposto a patch con la PSU di luglio 2021. Se non è stato sottoposto a patch, il client perde la connessione e riceve l'errore ORA-12269.

Il seguente esempio mostra impostazioni NNE di esempio. Supponiamo di voler impostare SQLNET.ENCRYPTION_TYPES_SERVER e SQLNET.ENCRYPTION_TYPES_CLIENT su FALSE, bloccando così connessioni non sicure. Le impostazioni dell'opzione checksum soddisfano i prerequisiti perché entrambe hanno SHA256. Tuttavia, SQLNET.ENCRYPTION_TYPES_CLIENT e SQLNET.ENCRYPTION_TYPES_SERVER utilizzano i metodi di crittografia DES, 3DES e RC4, che non sono sicuri. Pertanto, per impostare l'opzione SQLNET.ALLOW_WEAK_CRYPTO* su FALSE, imposta prima SQLNET.ENCRYPTION_TYPES_SERVER e SQLNET.ENCRYPTION_TYPES_CLIENT su un metodo di crittografia sicuro come AES256.

Impostazione opzioni	Valori
SQLNET.CRYPTO_CHECKSUM_TYPES_CLIENT	SHA256, SHA384, SHA512
SQLNET.CRYPTO_CHECKSUM_TYPES_SERVER	SHA1,MD5,SHA256
SQLNET.ENCRYPTION_TYPES_CLIENT	RC4_256, 3DES168, DES40
SQLNET.ENCRYPTION_TYPES_SERVER	RC4_256, 3DES168, DES40

Rimozione dell'opzione NNE

Puoi rimuovere NNE da un'istanza database.

Per rimuovere NNE da un'istanza database, esegui una delle seguenti procedure:

- Per rimuovere NNE da più istanze database, rimuovila dal gruppo di opzioni a cui le istanze appartengono. Questa modifica coinvolge tutte le istanze database che usano il gruppo di opzioni. Dopo avere rimosso l'opzione NNE, non devi riavviare le istanze database. Per ulteriori informazioni, consulta Rimozione di un'opzione da un gruppo di opzioni.
- Per rimuovere NNE da una singola istanza database, modifica l'istanza database e specifica un altro gruppo di opzioni che non include l'opzione NNE. È possibile specificare il gruppo di opzioni predefinito (vuoto) o un gruppo di opzioni personalizzato diverso. Dopo avere rimosso l'opzione NNE, non devi riavviare l'istanza database. Per ulteriori informazioni, consulta Modifica di un'istanza database Amazon RDS.

Oracle OLAP

Amazon RDS supporta Oracle OLAP tramite l'utilizzo dell'opzione OLAP. Questa opzione fornisce OLAP (OnLine Analytical Processing) per le istanze database di Oracle. È possibile utilizzare Oracle OLAP per analizzare grandi quantità di dati creando oggetti e cubi dimensionali in conformità con lo standard OLAP. Per ulteriori informazioni, consulta la documentazione di Oracle.

> **Important**
>
> Se utilizzi Oracle OLAP, Amazon RDS aggiorna automaticamente la tua istanza database all'Oracle PSU più recente in presenza di vulnerabilità della sicurezza con un punteggio Common Vulnerability Scoring System (CVSS) pari a 9+ o altre vulnerabilità della sicurezza annunciate.

Amazon RDS supporta Oracle OLAP per le versioni ed edizioni di Oracle seguenti:

- Oracle Database 21c Enterprise Edition, tutte le versioni
- Oracle Database 19c Enterprise Edition, tutte le versioni
- Oracle Database 12c Release 2 (12.2.0.1) Enterprise Edition, tutte le versioni
- Oracle Database 12c Release 1 (12.1.0.2) Enterprise Edition, versione 12.1.0.2.v13 o successiva

Prerequisiti per Oracle OLAP

Di seguito sono indicati i prerequisiti per l'utilizzo di Oracle OLAP:

- È necessario disporre di una licenza Oracle OLAP da Oracle. Per ulteriori informazioni, consulta Informazioni sulle licenze nella documentazione Oracle.
- L'istanza database deve essere di una classe di istanza sufficiente. Oracle OLAP non è supportato per le classi di istanza database db.t3.micro o db.t3.small. Per ulteriori informazioni, consulta Classi di istanza RDS for Oracle.
- L'istanza database deve avere l'opzione Auto Minor Version Upgrade (Aggiornamento minore automatico della versione) abilitata. Questa opzione consente all'istanza database di ricevere automaticamente gli aggiornamenti secondari della versione del motore del database quando vengono resi disponibili ed è obbligatoria per tutte le opzioni che installano la Java Virtual Machine (JVM) Oracle. Amazon RDS utilizza questa opzione per aggiornare l'istanza database all'ultimo

PSU (Patch Set Update) o RU (Release Update) di Oracle. Per ulteriori informazioni, consulta Modifica di un'istanza database Amazon RDS.

- L'istanza database non deve avere un utente denominato `OLAPSYS`. In tal caso, l'installazione dell'opzione OLAP non va a buon fine.

Best practice per Oracle OLAP

Di seguito sono indicate le best practice per l'utilizzo di Oracle OLAP:

- Per la massima sicurezza, è necessario utilizzare l'opzione `OLAP` con Secure Sockets Layer (SSL). Per ulteriori informazioni, consulta Oracle Secure Sockets Layer.
- Configurare l'istanza database per limitare l'accesso all'istanza database. Per ulteriori informazioni, consulta Scenari per accedere a un'istanza database in un VPC e Uso di un'istanza database in un VPC.

Aggiunta dell'opzione Oracle OLAP

Di seguito è riportato il processo generale per aggiungere l'opzione `OLAP` a un'istanza database:

1. Creare un nuovo gruppo di opzioni oppure copiare o modificare un gruppo di opzioni esistente.
2. Aggiungere l'opzione al gruppo di opzioni.
3. Associare il gruppo di opzioni a questa istanza database.

Se Oracle Java Virtual Machine non è installato nell'istanza database, durante l'aggiunta dell'opzione `OLAP` si verifica una breve interruzione. Se Oracle Java Virtual Machine è già installata nell'istanza database, non si verificherà alcuna interruzione. Una volta aggiunta l'opzione, non è necessario riavviare l'istanza database. Non appena il gruppo di opzioni diventa attivo, Oracle OLAP è disponibile.

Per aggiungere l'opzione OLAP a un'istanza database

1. Determinare il gruppo di opzioni che si vuole usare. È possibile creare un nuovo gruppo di opzioni oppure usare un gruppo di opzioni esistente. Se si desidera usare un gruppo di opzioni esistente, puoi passare alla fase successiva. In caso contrario, creare un gruppo di opzioni database personalizzato con le seguenti impostazioni:

   - Per Engine (Motore), scegliere l'edizione Oracle per l'istanza database.

- In Major engine version (Versione principale del motore), scegliere la versione dell'istanza database.

Per ulteriori informazioni, consulta Creazione di un gruppo di opzioni.

2. Aggiungere l'opzione OLAP al gruppo di opzioni. Per ulteriori informazioni sull'aggiunta di opzioni, consulta Aggiunta di un'opzione a un gruppo di opzioni.
3. Applicare il gruppo di opzioni a un'istanza database nuova o esistente:

   - Per una nuova istanza database, applicare il gruppo di opzioni quando viene avviata l'istanza. Per ulteriori informazioni, consulta Creazione di un'istanza database Amazon RDS.
   - Per un'istanza database esistente, applicare il gruppo di opzioni modificando l'istanza e collegando il nuovo gruppo di opzioni. Per ulteriori informazioni, consulta Modifica di un'istanza database Amazon RDS.

Utilizzo di Oracle OLAP

Dopo che è stata abilitata, è possibile iniziare a utilizzare immediatamente l'opzione Oracle OLAP. Per un elenco di funzionalità supportate per Oracle OLAP, consulta la documentazione di Oracle.

Rimozione dell'opzione Oracle OLAP

Dopo aver eliminato tutti gli oggetti che utilizzano i tipi di dati forniti dall'opzione `OLAP`, è possibile rimuovere l'opzione da un'istanza DB. Se Oracle Java Virtual Machine non è installata nell'istanza database, durante la rimozione dell'opzione `OLAP` si verifica una breve interruzione. Se Oracle Java Virtual Machine è già installata nell'istanza database, non si verificherà alcuna interruzione. Una volta rimossa l'opzione `OLAP` non è necessario riavviare la tua istanza database.

Per eliminare l'opzione **`OLAP`**

1. Eseguire il backup dei dati.

> **Warning**
>
> Se l'istanza utilizza tipi di dati abilitati come parte dell'opzione e se si rimuove l'opzione `OLAP`, è possibile perdere i dati. Per ulteriori informazioni, consulta Backup, ripristino ed esportazione dei dati.

2. Verificare se gli oggetti esistenti fanno riferimento ai tipi di dati o alle feature dell'opzione `OLAP`.
3. Eliminare gli oggetti che fanno riferimento ai tipi di dati o alle feature dell'opzione `OLAP`.
4. Scegliere una delle seguenti operazioni:

   - Rimuovere l'opzione `OLAP` dal gruppo di opzioni a cui appartiene. Questa modifica coinvolge tutte le istanze database che usano il gruppo di opzioni. Per ulteriori informazioni, consulta Rimozione di un'opzione da un gruppo di opzioni.
   - Modificare l'istanza database e specificare un diverso gruppo di opzioni che non comprenda l'opzione `OLAP`. Questa modifica coinvolge una singola istanza database. È possibile specificare il gruppo di opzioni predefinito (vuoto) o un gruppo di opzioni personalizzato diverso. Per ulteriori informazioni, consulta Modifica di un'istanza database Amazon RDS.

Oracle Secure Sockets Layer

Puoi abilitare la crittografia Secure Sockets Layer (SSL) per un'istanza database RDS per Oracle aggiungendo l'opzione Oracle SSL al gruppo di opzioni associato all'istanza database. Amazon RDS utilizza una seconda porta, come richiesto da Oracle, per le connessioni SSL. Questo approccio rende possibile allo stesso tempo sia testo in chiaro che comunicazioni con crittografia SSL tra un'istanza database e SQL*Plus. Ad esempio, è possibile utilizzare la porta con testo in chiaro per comunicare con altre risorse all'interno di un VPC mentre utilizzi la porta con crittografia SSL per comunicare con risorse all'esterno del VPC.

> **Note**
>
> È possibile utilizzare SSL o Native Network Encryption (NNE) sulla stessa istanza database RDS per Oracle. Se utilizzi la crittografia SSL, assicurati di disabilitare qualsiasi altro metodo di crittografia della connessione. Per ulteriori informazioni, consulta Oracle native network encryption.

SSL/TLS e NNE non fanno più parte di Oracle Advanced Security. In RDS per Oracle, puoi utilizzare la crittografia SSL con tutte le edizioni con licenza delle seguenti versioni del database:

- Oracle Database 21c (21.0.0)
- Oracle Database 19c (19.0.0)
- Oracle Database 12c Release 2 (12.2) - questa versione non è più supportata
- Oracle Database 12c Release 1 (12.1) - questa versione non è più supportata

Versioni TLS per l'opzione SSL di Oracle

Amazon RDS for Oracle supporta Transport Layer Security (TLS) versioni 1.0 e 1.2. Quando aggiungi una nuova opzione SSL di Oracle, imposta `SQLNET.SSL_VERSION` su un valore valido in modo esplicito. Di seguito sono indicati i valori consentiti per questa impostazione dell'opzione:

- `"1.0"`: i client possono connettersi all'istanza database solo tramite TLS versione 1.0. Per le opzioni SSL di Oracle esistenti, `SQLNET.SSL_VERSION` è impostato su `"1.0"` automaticamente. Puoi modificare questa impostazione, se necessario.
- `"1.2"` – I client possono connettersi all'istanza database solo tramite TLS 1.2.
- `"1.2 or 1.0"` – I client possono connettersi all'istanza database tramite TLS 1.2 o 1.0.

Suite di cifratura per l'opzione Oracle SSL

Amazon RDS for Oracle supporta suite di cifratura SSL multiple. Come impostazione predefinita, l'opzione Oracle SSL è configurata per utilizzare la suite di cifratura `SSL_RSA_WITH_AES_256_CBC_SHA`. Per specificare una suite di cifratura diversa da adottare nelle connessioni SSL, usare l'impostazione dell'opzione `SQLNET.CIPHER_SUITE`.

Nella tabella seguente viene indicato il supporto SSL di RDS per Oracle. Le versioni del database Oracle specificate supportano tutte le edizioni.

Suite di crittografia (SQLNET.CIPHER_SUITE)	Supporto della versione TLS (SQLNET.SSL_VERSION)	Versioni Oracle Database supportate	Supporto FIPS	Conformità agli standard FedRAMP
SSL_RSA_WITH_AES_256_CBC_SHA (predefinito)	1.0 e 1.2	12c, 19c, 21c	Sì	No
SSL_RSA_WITH_AES_256_CBC_SHA256	1.2	12c, 19c, 21c	Sì	No
SSL_RSA_WITH_AES_256_GCM_SHA384	1.2	12c, 19c, 21c	Sì	No
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	1.2	19c, 21c	Sì	Sì
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	1.2	19c, 21c	Sì	Sì
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384	1.2	19c, 21c	Sì	Sì
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256	1.2	19c, 21c	Sì	Sì
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	1.2	19c, 21c	Sì	Sì

Suite di crittografia (SQLNET.CIPHER_SUITE)	Supporto della versione TLS (SQLNET.SSL_VERSION)	Versioni Oracle Database supportate	Supporto FIPS	Conformità agli standard FedRAMP
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	1.2	19c, 21c	Sì	Sì

Supporto FIPS

RDS per Oracle consente di utilizzare lo standard Federal Information Processing Standard (FIPS) per 140-2. FIPS 140-2 è uno standard del governo degli Stati Uniti che definisce i requisiti di sicurezza del modulo crittografico. Attiva lo standard FIPS impostando `FIPS.SSLFIPS_140` su `TRUE` per l'opzione Oracle SSL. Quando FIPS 140-2 è configurato per SSL, le librerie crittografiche eseguono la crittografia dei dati tra il client e l'istanza database Oracle.

I client devono utilizzare la suite di crittografia conforme con FIPS. Quando si stabilisce una connessione, il client e l'istanza database RDS per Oracle negoziano quale suite di cifratura utilizzare durante la trasmissione dei messaggi in entrambe le direzioni. Nella tabella in Suite di cifratura per l'opzione Oracle SSL vengono illustrate le suite di crittografia SSL conformi a FIPS per ogni versione TLS. Per ulteriori informazioni, consulta la pagina relativa alle impostazioni FIPS 140-2 del database Oracle nella documentazione del database Oracle.

Aggiunta dell'opzione SSL

Per utilizzare SSL, l'istanza database RDS per Oracle deve essere associata a un gruppo di opzioni che include l'opzione `SSL`.

Console

Per aggiungere l'opzione SSL a un gruppo di opzioni

1. Creare un nuovo gruppo di opzioni o identificare un gruppo opzioni esistente a cui è possibile aggiungere l'opzione `SSL`.

   Per informazioni sulla creazione di un gruppo di opzioni, consulta Creazione di un gruppo di opzioni.
2. Aggiungere l'opzione `SSL` al gruppo di opzioni.

Se si desidera utilizzare solo suite di crittografia conformi a FIPS per le connessioni SSL, impostare l'opzione `FIPS.SSLFIPS_140` su `TRUE`. Per informazioni sullo standard FIPS, consulta Supporto FIPS.

Per informazioni sull'aggiunta di un'opzione a un gruppo di opzioni, consulta Aggiunta di un'opzione a un gruppo di opzioni.

3. Crea una nuova istanza database RDS per Oracle e associarvi il gruppo opzioni oppure modificare un'istanza database RDS per Oracle per associare il gruppo opzioni a essa.

   Per informazioni sulla creazione di un'istanza database, consulta Creazione di un'istanza database Amazon RDS.

   Per informazioni sulla modifica di un'istanza database, consulta Modifica di un'istanza database Amazon RDS.

AWS CLI

Per aggiungere l'opzione SSL a un gruppo di opzioni

1. Creare un nuovo gruppo di opzioni o identificare un gruppo opzioni esistente a cui è possibile aggiungere l'opzione `SSL`.

   Per informazioni sulla creazione di un gruppo di opzioni, consulta Creazione di un gruppo di opzioni.

2. Aggiungere l'opzione `SSL` al gruppo di opzioni.

   Specificare le seguenti impostazioni delle opzioni:

   - `Port` – Il numero di porta SSL
   - `VpcSecurityGroupMemberships` – Il gruppo di sicurezza VPC per cui è abilitata l'opzione
   - `SQLNET.SSL_VERSION` – La versione TLS utilizzabile dal client per connettersi all'istanza database

   Ad esempio, il comando seguente AWS CLI aggiunge l'opzione `SSL` a un gruppo opzioni denominato `ora-option-group`.

Example

Per LinuxmacOS, oUnix:

```
aws rds add-option-to-option-group --option-group-name ora-option-group \
  --options
 'OptionName=SSL,Port=2484,VpcSecurityGroupMemberships="sg-68184619",OptionSettings=[{Name=
```

Per Windows:

```
aws rds add-option-to-option-group --option-group-name ora-option-group ^
  --options
 'OptionName=SSL,Port=2484,VpcSecurityGroupMemberships="sg-68184619",OptionSettings=[{Name=
```

3. Crea una nuova istanza database RDS per Oracle e associarvi il gruppo opzioni oppure modificare un'istanza database RDS per Oracle per associare il gruppo opzioni a essa.

   Per informazioni sulla creazione di un'istanza database, consulta Creazione di un'istanza database Amazon RDS.

   Per informazioni sulla modifica di un'istanza database, consulta Modifica di un'istanza database Amazon RDS.

Configurazione di SQL*Plus per l'utilizzo di SSL con un'istanza database RDS per Oracle

Prima di connetterti a un'istanza database RDS per Oracle che utilizza l'opzione SSL di Oracle, devi configurare SQL*Plus.

> **Note**
>
> Per concedere l'accesso all'istanza database dai client appropriati, verifica che i gruppi di sicurezza siano configurati correttamente. Per ulteriori informazioni, consulta Controllo dell'accesso con i gruppi di sicurezza. Inoltre, queste istruzioni si riferiscono a SQL*Plus e ad altri client che usano direttamente una home directory Oracle. Per le connessioni JDBC, consulta Configurazione di una connessione SSL su JDBC.

Per configurare SQL*Plus in modo che utilizzi SSL per connettersi a un'istanza database RDS per Oracle

1. Imposta la variabile di ambiente `ORACLE_HOME` sulla posizione della home directory di Oracle.

   Il percorso della home directory di Oracle dipende dall'installazione. Nell'esempio seguente viene impostata la variabile di ambiente `ORACLE_HOME`.

   ```
   prompt>export ORACLE_HOME=/home/user/app/user/product/12.1.0/dbhome_1
   ```

   Per informazioni sull'impostazione delle variabili di ambiente di Oracle, consulta Variabili di ambiente di SQL*Plus nella documentazione di Oracle e la guida di installazione di Oracle per il tuo sistema operativo.

2. Aggiungi `$ORACLE_HOME/lib` alla variabile di ambiente `LD_LIBRARY_PATH`.

   Nell'esempio seguente viene impostata la variabile di ambiente LD_LIBRARY_PATH.

   ```
   prompt>export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ORACLE_HOME/lib
   ```

3. Crea una directory per il wallet Oracle nel percorso `$ORACLE_HOME/ssl_wallet`.

   Nell'esempio seguente viene creata la directory per il wallet Oracle.

   ```
   prompt>mkdir $ORACLE_HOME/ssl_wallet
   ```

4. Scarica il file .pem del pacchetto di certificati che funziona per tutti Regioni AWS e inserisci il file nella directory ssl_wallet. Per informazioni, consulta .

5. Nella directory `$ORACLE_HOME/network/admin`, modifica o crea il file `tnsnames.ora` e includi la voce seguente.

   ```
   net_service_name =
     (DESCRIPTION =
       (ADDRESS_LIST =
         (ADDRESS =
           (PROTOCOL = TCPS)
           (HOST = endpoint)
           (PORT = ssl_port_number)
         )
       )
       (CONNECT_DATA =
   ```

```
      (SID = database_name)
    )
    (SECURITY =
      (SSL_SERVER_CERT_DN =
"C=US,ST=Washington,L=Seattle,O=Amazon.com,OU=RDS,CN=endpoint")
    )
  )
```

6. Nella stessa directory, modifica o crea il file sqlnet.ora e includi i parametri seguenti.

> **Note**
>
> Per comunicare con le entità tramite una connessione protetta TLS, Oracle richiede un wallet con i certificati necessari per l'autenticazione. È possibile utilizzare l'utility ORAPKI di Oracle per creare e gestire i wallet Oracle, come mostrato nel fase 7. Per ulteriori informazioni, consulta Setting Up Oracle Wallet Using ORAPKI nella documentazione Oracle.

```
WALLET_LOCATION = (SOURCE = (METHOD = FILE) (METHOD_DATA = (DIRECTORY =
 $ORACLE_HOME/ssl_wallet)))
SSL_CLIENT_AUTHENTICATION = FALSE
SSL_VERSION = 1.0
SSL_CIPHER_SUITES = (SSL_RSA_WITH_AES_256_CBC_SHA)
SSL_SERVER_DN_MATCH = ON
```

> **Note**
>
> È possibile impostare `SSL_VERSION` su un valore più alto se supportato dall'istanza database.

7. Esegui il comando seguente per creare il portafoglio Oracle.

```
prompt>orapki wallet create -wallet $ORACLE_HOME/ssl_wallet -auto_login_only
```

8. Estrai ogni certificato nel file bundle .pem in un file.pem separato utilizzando un'utilità del sistema operativo.
9. Aggiungi ogni certificato al tuo portafoglio utilizzando `orapki` comandi separati, sostituendoli *`certificate-pem-file`* con il nome di file assoluto del file.pem.

```
prompt>orapki wallet add -wallet $ORACLE_HOME/ssl_wallet -trusted_cert -cert
      certificate-pem-file -auto_login_only
```

Per ulteriori informazioni, consulta Rotazione del certificato SSL/TLS.

Connessione a un'istanza database RDS per Oracle tramite SSL

Dopo aver configurato SQL*Plus per l'uso di SSL come descritto in precedenza, puoi connetterti all'istanza database RDS per Oracle con l'opzione SSL. Facoltativamente, puoi innanzitutto esportare il valore TNS_ADMIN che punta alla directory contenente i file tnsnames.ora e sqlnet.ora. In questo modo, SQL*Plus può trovare questi file in modo coerente. Nell'esempio seguente viene esportato il valore TNS_ADMIN.

```
export TNS_ADMIN = ${ORACLE_HOME}/network/admin
```

Effettua la connessione all'istanza database. Ad esempio, puoi connetterti tramite SQL*Plus e un *<net_service_name>* in un file tnsnames.ora.

```
sqlplus mydbuser@net_service_name
```

Puoi anche connetterti all'istanza database con SQL*Plus senza utilizzare un file tnsnames.ora tramite il comando seguente.

```
sqlplus 'mydbuser@(DESCRIPTION = (ADDRESS = (PROTOCOL = TCPS)(HOST = endpoint) (PORT
 = ssl_port_number))(CONNECT_DATA = (SID = database_name)))'
```

Puoi anche connetterti all'istanza database RDS per Oracle senza utilizzare SSL. Il comando seguente, ad esempio, consente la connessione all'istanza database sulla porta con testo in chiaro senza codifica SSL.

```
sqlplus 'mydbuser@(DESCRIPTION = (ADDRESS = (PROTOCOL = TCP)(HOST = endpoint) (PORT
 = port_number))(CONNECT_DATA = (SID = database_name)))'
```

Se desideri chiudere l'accesso alla porta Transmission Control Protocol (TCP), crea un gruppo di sicurezza senza ingressi con indirizzo IP e aggiungilo all'istanza. In questo modo vengono chiuse le connessioni sulla porta TCP, ma continuano a essere permesse quelle sulla porta SSL specificate dagli indirizzi IP compresi nell'intervallo consentito dal gruppo di sicurezza dell'opzione SSL.

Configurazione di una connessione SSL su JDBC

Per utilizzare una connessione SSL su JDBC, devi creare un keystore, approvare il certificato CA root di Amazon RDS e utilizzare il frammento di codice specificato di seguito.

Per creare il keystore in formato JKS, puoi usare il seguente comando. Per ulteriori informazioni sulla creazione del keystore, vedere Creazione di un keystore nella documentazione di Oracle. Per informazioni di riferimento, vedere keytool nella piattaforma Java, Standard Edition Tools Reference.

```
keytool -genkey -alias client -validity 365 -keyalg RSA -keystore clientkeystore
```

Segui i passaggi seguenti per considerare attendibile il certificato CA root di Amazon RDS.

Per approvare il certificato CA root di Amazon RDS

1. Scarica il file .pem del pacchetto di certificati che funziona per tutti Regioni AWS e inserisci il file nella directory ssl_wallet.

   Per ulteriori informazioni sul download dei certificati, consultare .
2. Estrai ogni certificato nel file.pem in un file separato utilizzando un'utilità del sistema operativo.
3. Converti ogni certificato in formato.der utilizzando un `openssl` comando separato, sostituendolo *`certificate-pem-file`*con il nome del file Certificate .pem (senza l'estensione.pem).

   ```
   openssl x509 -outform der -in certificate-pem-file.pem -out certificate-pem-file.der
   ```
4. Importa ogni certificato nel keystore usando il seguente comando.

   ```
   keytool -import -alias rds-root -keystore clientkeystore.jks -file certificate-pem-file.der
   ```

   Per ulteriori informazioni, consulta Rotazione del certificato SSL/TLS.
5. Verificare che il keystore sia stato creato correttamente.

   ```
   keytool -list -v -keystore clientkeystore.jks
   ```

   Inserire la password del keystore quando richiesto.

L'esempio di codice seguente mostra come impostare la connessione SSL utilizzando JDBC.

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.util.Properties;

public class OracleSslConnectionTest {
    private static final String DB_SERVER_NAME = "dns-name-provided-by-amazon-rds";
    private static final Integer SSL_PORT = "ssl-option-port-configured-in-option-
group";
    private static final String DB_SID = "oracle-sid";
    private static final String DB_USER = "user-name";
    private static final String DB_PASSWORD = "password";
    // This key store has only the prod root ca.
    private static final String KEY_STORE_FILE_PATH = "file-path-to-keystore";
    private static final String KEY_STORE_PASS = "keystore-password";

    public static void main(String[] args) throws SQLException {
        final Properties properties = new Properties();
        final String connectionString = String.format(
                "jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCPS)(HOST=%s)(PORT=
%d))(CONNECT_DATA=(SID=%s)))",
                DB_SERVER_NAME, SSL_PORT, DB_SID);
        properties.put("user", DB_USER);
        properties.put("password", DB_PASSWORD);
        properties.put("oracle.jdbc.J2EE13Compliant", "true");
        properties.put("javax.net.ssl.trustStore", KEY_STORE_FILE_PATH);
        properties.put("javax.net.ssl.trustStoreType", "JKS");
        properties.put("javax.net.ssl.trustStorePassword", KEY_STORE_PASS);
        final Connection connection = DriverManager.getConnection(connectionString,
 properties);
        // If no exception, that means handshake has passed, and an SSL connection can
 be opened
    }
}
```

> **Note**
>
> Specifica una password diversa dal prompt mostrato qui come best practice per la sicurezza.

Applicazione di una corrispondenza DN con connessione SSL

Puoi utilizzare il parametro `SSL_SERVER_DN_MATCH` di Oracle per applicare il nome distinto (DN) in modo che vi sia corrispondenza tra il server del database e il nome del relativo servizio. Se applichi la verifica delle corrispondenze, SSL fa in modo che il certificato provenga dal server. Se non esegui tale verifica, SSL esegue il controllo ma consente la connessione, indipendentemente dall'esistenza di una corrispondenza. Se non applichi la verifica, consenti virtualmente al server di simulare la propria identità.

Per applicare la corrispondenza DN, aggiungi la relativa proprietà e utilizza la stringa di connessione specificata di seguito.

Per applicare la corrispondenza DN, aggiungi tale proprietà alla connessione client.

```
properties.put("oracle.net.ssl_server_dn_match", "TRUE");
```

Utilizza la stringa di connessione seguente per applicare la corrispondenza DN quando utilizzi SSL.

```
final String connectionString = String.format(
    "jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCPS)(HOST=%s)(PORT=%d))" +
    "(CONNECT_DATA=(SID=%s))" +
    "(SECURITY = (SSL_SERVER_CERT_DN =
\"C=US,ST=Washington,L=Seattle,O=Amazon.com,OU=RDS,CN=%s\")))",
    DB_SERVER_NAME, SSL_PORT, DB_SID, DB_SERVER_NAME);
```

Risoluzione dei problemi relativi alle connessioni SSL

Quando esegui le query sul database è possibile ricevere l'errore `ORA-28860`.

```
ORA-28860: Fatal SSL error
28860. 00000 - "Fatal SSL error"
*Cause: An error occurred during the SSL connection to the peer. It is likely that this
 side sent data which the peer rejected.
*Action: Enable tracing to determine the exact cause of this error.
```

Questo errore si verifica quando il client tenta di connettersi utilizzando una versione di TLS non supportata dal server. Per evitare questo errore, modifica sqlnet.ora e imposta `SSL_VERSION` sulla versione TLS corretta. Per ulteriori informazioni, consulta il documento My Oracle Support 2748438.1.

Oracle Spatial

Amazon RDS supporta Oracle Spatial tramite l'utilizzo dell'opzione `SPATIAL`. Oracle Spatial offre uno schema SQL e funzioni che semplificano le operazioni di storage, recupero, aggiornamento ed esecuzione di query per le raccolte di dati spaziali in un database Oracle. Per ulteriori informazioni, consulta la pagina Spatial Concepts nella documentazione Oracle.

Important

Se si utilizza Oracle Spatial, Amazon RDS automaticamente aggiorna l'istanza DB all'ultima PSU Oracle quando esiste una delle seguenti opzioni:

- Vulnerabilità di sicurezza con un punteggio CVSS (Common Vulnerability Scoring System) pari a 9+
- Altre vulnerabilità annunciate per la sicurezza

Amazon RDS supporta Oracle Spatial solo in Oracle Enterprise Edition (EE) e Oracle Standard Edition 2 (SE2). Nella tabella seguente vengono illustrate le versioni del motore DB che supportano EE e SE2.

Versione Oracle DB	EE	SE2
21.0.0.0, tutte le versioni	Sì	Sì
19.0.0.0, tutte le versioni	Sì	Sì
12.2.0.1, tutte le versioni	Sì	Sì
12.1.0.2.v13 o versioni successive	Sì	No

Note

In Oracle Database 19c, i bundle di patch spaziali sono separati dai Patch Set Updates (PSU) e Release Updates (RU) del database. RDS per Oracle non supporta l'applicazione di bundle batch Spatial.

Prerequisiti per Oracle Spatial

Di seguito sono indicati i prerequisiti per l'utilizzo di Oracle Spatial:

- Assicurati che l'istanza DB sia di una classe di istanza sufficiente. Oracle Spatial non è supportato per le classi di istanza database db.t3.micro o db.t3.small. Per ulteriori informazioni, consulta Classi di istanza RDS for Oracle.
- Assicurarsi che l'istanza del DB abbia abilitato l'aggiornamento automatico della versione minore. Questa opzione consente all'istanza database di ricevere automaticamente gli aggiornamenti secondari della versione del motore del database quando vengono resi disponibili ed è obbligatoria per tutte le opzioni che installano la Java Virtual Machine (JVM) Oracle. Amazon RDS utilizza questa opzione per aggiornare l'istanza database all'ultimo PSU (Patch Set Update) o RU (Release Update) di Oracle. Per ulteriori informazioni, consulta Modifica di un'istanza database Amazon RDS.

Best practice per Oracle Spatial

Di seguito sono indicate le best practice per l'utilizzo di Oracle Spatial:

- Per la massima sicurezza, è necessario utilizzare l'opzione `SPATIAL` con Secure Sockets Layer (SSL). Per ulteriori informazioni, consulta Oracle Secure Sockets Layer.
- Configurare l'istanza database per limitare l'accesso all'istanza database. Per ulteriori informazioni, consulta Scenari per accedere a un'istanza database in un VPC e Uso di un'istanza database in un VPC.

Aggiunta dell'opzione Oracle Spatial

Di seguito è riportato il processo generale per aggiungere l'opzione `SPATIAL` a un'istanza database:

1. Creare un nuovo gruppo di opzioni oppure copiare o modificare un gruppo di opzioni esistente.
2. Aggiungere l'opzione al gruppo di opzioni.
3. Associare il gruppo di opzioni a questa istanza database.

Se Oracle Java Virtual Machine non è installato nell'istanza database, durante l'aggiunta dell'opzione `SPATIAL` si verifica una breve interruzione. Se Oracle Java Virtual Machine è già installata nell'istanza database, non si verificherà alcuna interruzione. Una volta aggiunta l'opzione, non è

necessario riavviare l'istanza database. Non appena il gruppo di opzioni è attivo, Oracle Spatial è disponibile.

> **Note**
>
> Durante questa interruzione, le funzioni di verifica delle password vengono temporaneamente disabilitate. Durante l'interruzione si possono verificare eventi correlati alle funzioni di verifica delle password. Le funzioni di verifica delle password vengono riabilitate prima di rendere disponibile l'istanza database Oracle.

Per aggiungere l'opzione **`SPATIAL`** a un'istanza database

1. Determinare il gruppo di opzioni che si vuole usare. È possibile creare un nuovo gruppo di opzioni oppure usare un gruppo di opzioni esistente. Se si desidera usare un gruppo di opzioni esistente, puoi passare alla fase successiva. In caso contrario, creare un gruppo di opzioni database personalizzato con le seguenti impostazioni:

   a. Per Engine (Motore), scegliere l'edizione Oracle per l'istanza database.
   b. In Major engine version (Versione principale del motore), scegliere la versione dell'istanza database.

   Per ulteriori informazioni, consulta Creazione di un gruppo di opzioni.
2. Aggiungere l'opzione SPATIAL al gruppo di opzioni. Per ulteriori informazioni sull'aggiunta di opzioni, consulta Aggiunta di un'opzione a un gruppo di opzioni.
3. Applicare il gruppo di opzioni a un'istanza database nuova o esistente:

   - Per una nuova istanza database, si applica il gruppo di opzioni quando viene avviata l'istanza. Per ulteriori informazioni, consulta Creazione di un'istanza database Amazon RDS.
   - Per un'istanza database esistente, viene applicato il gruppo di opzioni modificando l'istanza e collegando il nuovo gruppo di opzioni. Per ulteriori informazioni, consulta Modifica di un'istanza database Amazon RDS.

Rimozione dell'opzione Oracle Spatial

Dopo aver rimosso tutti gli oggetti che utilizzano i tipi di dati forniti dall'opzione `SPATIAL`, è possibile eliminare l'opzione da un'istanza DB. Se Oracle Java Virtual Machine non è installata nell'istanza

database, durante la rimozione dell'opzione SPATIAL si verifica una breve interruzione. Se Oracle Java Virtual Machine è già installata nell'istanza database, non si verificherà alcuna interruzione. Una volta rimossa l'opzione SPATIAL non è necessario riavviare la tua istanza database.

Per eliminare l'opzione **SPATIAL**

1. Eseguire il backup dei dati.

   > **Warning**
   >
   > Se l'istanza utilizza tipi di dati abilitati come parte dell'opzione e se si rimuove l'opzione SPATIAL, è possibile perdere i dati. Per ulteriori informazioni, consulta Backup, ripristino ed esportazione dei dati.

2. Verificare se gli oggetti esistenti fanno riferimento ai tipi di dati o alle feature dell'opzione SPATIAL.

   Se esistono opzioni SPATIAL, l'istanza può rimanere bloccata quando si applica il nuovo gruppo di opzioni che non dispone dell'opzione SPATIAL. È possibile identificare gli oggetti utilizzando le seguenti query:

   ```
   SELECT OWNER, SEGMENT_NAME, TABLESPACE_NAME, BYTES/1024/1024 mbytes
   FROM   DBA_SEGMENTS
   WHERE  SEGMENT_TYPE LIKE '%TABLE%'
   AND    (OWNER, SEGMENT_NAME) IN
          (SELECT DISTINCT OWNER, TABLE_NAME
           FROM   DBA_TAB_COLUMNS
           WHERE  DATA_TYPE='SDO_GEOMETRY'
           AND    OWNER <> 'MDSYS')
   ORDER BY 1,2,3,4;

   SELECT OWNER, TABLE_NAME, COLUMN_NAME
   FROM   DBA_TAB_COLUMNS
   WHERE  DATA_TYPE = 'SDO_GEOMETRY'
   AND    OWNER <> 'MDSYS'
   ORDER BY 1,2,3;
   ```

3. Eliminare gli oggetti che fanno riferimento ai tipi di dati o alle feature dell'opzione SPATIAL.
4. Scegliere una delle seguenti operazioni:

- Rimuovere l'opzione `SPATIAL` dal gruppo di opzioni a cui appartiene. Questa modifica coinvolge tutte le istanze database che usano il gruppo di opzioni. Per ulteriori informazioni, consulta Rimozione di un'opzione da un gruppo di opzioni.
- Modificare l'istanza database e specificare un diverso gruppo di opzioni che non comprenda l'opzione `SPATIAL`. Questa modifica coinvolge una singola istanza database. È possibile specificare il gruppo di opzioni predefinito (vuoto) o un gruppo di opzioni personalizzato diverso. Per ulteriori informazioni, consulta Modifica di un'istanza database Amazon RDS.

Oracle SQLT

Amazon RDS supporta Oracle SQLTXPLAIN (SQLT) attraverso l'utilizzo dell'opzione SQLT.

L'istruzione `EXPLAIN PLAN` di Oracle può stabilire il piano di esecuzione di un'istruzione SQL. Può verificare se l'ottimizzatore di Oracle sceglie un determinato piano di esecuzione, come un loop nidificato. Consente inoltre di comprendere le decisioni dell'ottimizzatore, ad esempio il motivo della scelta di loop nidificati rispetto a un hash join. Pertanto, `EXPLAIN PLAN` aiuta a comprendere le prestazioni dell'istruzione.

SQLT è un'utilità di Oracle che produce un rapporto. Tale rapporto include statistiche e metadati degli oggetti, parametri di inizializzazione correlati all'ottimizzatore e altre informazioni che possono essere utilizzate da un amministratore del database per modificare un'istruzione SQL e migliorarne le prestazioni. SQLT produce un rapporto HTML con collegamenti ipertestuali a tutte le relative sezioni.

A differenza dei rapporti di Automatic Workload Repository o Statspack, SQLT agisce sulle singole istruzioni SQL. SQLT è una raccolta di file SQL, PL/SQL e SQL*Plus che consente di raccogliere, archiviare e visualizzare i dati delle prestazioni.

Di seguito sono indicate le versioni Oracle supportate per ogni versione di SQLT.

Versione SQLT	Oracle Database 21c	Oracle Database 19c	Oracle Database 12c Release 2 (12.2)	Oracle Database 12c Release 1 (12.1)
2018-07-25.v1	Supportato	Supportato	Supportato	Supportato
2018-03-31.v1	Non supportato	Non supportato	Supportato	Supportato
2016-04-29.v1	Non supportato	Non supportato	Supportato	Supportato

Per il download e le istruzioni per l'accesso di SQLT:

- Accedere all'account My Oracle Support e aprire i documenti seguenti:
- Per scaricare SQLT: Document 215187.1
- Per istruzioni per l'utilizzo di SQLT: Document 1614107.1
- Per le domande frequenti su SQLT: Document 1454160.1
- Per informazioni sulla lettura dell'output di SQLT: Document 1456176.1

- Per l'interpretazione del report principale: Documento 1922234.1

Puoi utilizzare SQLT con qualsiasi edizione delle seguenti versioni di Oracle Database:

- Oracle Database 21c (21.0.0.0)
- Oracle Database 19c (19.0.0.0)
- Oracle Database 12c Release 2 (12.2.0.1)
- Oracle Database 12c Release 1 (12.1.0.2)

Amazon RDS non supporta i seguenti metodi SQLT:

- `XPLORE`
- `XHUME`

Prerequisiti di SQLT

Di seguito sono indicati i prerequisiti per l'utilizzo di SQLT:

- Devi rimuovere gli utenti e ruoli richiesti da SQLT, se esistenti.

  L'opzione SQLT crea i seguenti utenti e ruoli in un'istanza database:

  - `SQLTXPLAIN`Utente
  - `SQLTXADMIN`Utente
  - `SQLT_USER_ROLE`Ruolo

  Se nell'istanza database sono presenti uno o più di tali utenti o ruoli, accedi all'istanza database con un client SQL ed eliminali con le seguenti istruzioni:

```
DROP USER SQLTXPLAIN CASCADE;
DROP USER SQLTXADMIN CASCADE;
DROP ROLE SQLT_USER_ROLE CASCADE;
```

- Devi rimuovere gli spazi tabelle richiesti da SQLT, se esistenti.

  L'opzione SQLT crea i seguenti spazi tabelle in un'istanza database:

- `RDS_SQLT_TS`
- `RDS_TEMP_SQLT_TS`

Se nell'istanza database sono presenti tali spazi tabelle, accedi all'istanza database con un client SQL ed eliminali con le seguenti istruzioni:

Impostazioni dell'opzione SQLT

SQLT può essere utilizzata con caratteristiche con licenza fornite da Oracle Tuning Pack e da Oracle Diagnostics Pack. Oracle Tuning Pack include SQL Tuning Advisor, mentre Oracle Diagnostics Pack include Automatic Workload Repository. Le impostazioni SQLT consentono di abilitare o disabilitare l'accesso a tali caratteristiche da SQLT.

Amazon RDS supporta le seguenti impostazioni per l'opzione SQLT.

Impostazione opzioni	Valori validi	Valore predefinito	Descrizione
`LICENSE_PACK`	`T`, `D`, `N`	`N`	Gli Oracle Management Pack a cui desideri accedere con SQLT. Immetti uno dei seguenti valori: • `T` indica che possiedi una licenza per Oracle Tuning Pack e Oracle Diagnostics Pack e che desideri accedere a SQL Tuning Advisor e ad Automatic Workload Repository da SQLT. • `D` indica che possiedi una licenza per Oracle Tuning Pack e che desideri accedere ad Automatic Workload Repository da SQLT. • `N` indica che non possiedi una licenza per Oracle Tuning Pack e Oracle Diagnostics Pack o che possiedi una licenza per uno dei

Impostazione opzioni	Valori validi	Valore predefinito	Descrizione
			due, ma non desideri accedervi con SQLT. **Note** Amazon RDS non fornisce licenze per i seguenti Oracle Management Pack. Se specifichi che desideri utilizzare un pacchetto non incluso nell'istanza database, puoi utilizzare SQLT con tale istanza database. Tuttavia, SQLT non sarà in grado di accedere al pacchetto e il rapporto SQLT non includerà i relativi dati. Ad esempio, se specifichi T, ma l'istanza database non include Oracle Tuning Pack, SQLT opererà sull'istanza database, ma il rapporto generato non includerà i dati relativi a Oracle Tuning Pack.

Impostazione opzioni	Valori validi	Valore predefinito	Descrizione
VERSION	2016-04-29.v1 2018-03-31.v1 2018-07-25.v1	2016-04-29.v1	Versione di SQLT da installare. **Note** Per Oracle Database 19c e 21c, l'unica versione supportata è 2018-07-25.v1 . Questa è la versione predefinita per queste versioni.

Aggiunta dell'opzione SQLT

Di seguito è riportato il processo generale per aggiungere l'opzione SQLT a un'istanza database:

1. Creare un nuovo gruppo di opzioni oppure copiare o modificare un gruppo di opzioni esistente.
2. Aggiungere l'opzione SQLT al gruppo di opzioni.
3. Associare il gruppo di opzioni a questa istanza database.

Una volta aggiunta l'opzione SQLT, non appena il gruppo di opzioni sarà attivo, anche SQLT sarà attivo.

Per aggiungere l'opzione SQLT a un'istanza database

1. Determinare il gruppo di opzioni che si vuole usare. È possibile creare un nuovo gruppo di opzioni oppure usare un gruppo di opzioni esistente. Se si desidera usare un gruppo di opzioni esistente, puoi passare alla fase successiva. In caso contrario, creare un gruppo di opzioni database personalizzato con le seguenti impostazioni:
   a. Per Engine (Motore), scegliere l'edizione di Oracle da utilizzare. L'opzione SQLT è supportata in tutte le edizioni.
   b. In Major engine version (Versione principale del motore), scegliere la versione dell'istanza database.

Per ulteriori informazioni, consulta Creazione di un gruppo di opzioni.

2. Aggiungere l'opzione SQLT al gruppo di opzioni. Per ulteriori informazioni sull'aggiunta di opzioni, consulta Aggiunta di un'opzione a un gruppo di opzioni.
3. Applicare il gruppo di opzioni a un'istanza database nuova o esistente:

   - Per una nuova istanza database, si applica il gruppo di opzioni quando viene avviata l'istanza. Per ulteriori informazioni, consulta Creazione di un'istanza database Amazon RDS.

   - Per un'istanza database esistente, viene applicato il gruppo di opzioni modificando l'istanza e collegando il nuovo gruppo di opzioni. Per ulteriori informazioni, consulta Modifica di un'istanza database Amazon RDS.
4. (Opzionale) Verificare l'installazione SQLT in ciascuna istanza database dell'opzione SQLT.

   a. Utilizzare un client SQL per effettuare la connessione all'istanza database come utente master.

      Per informazioni sulla connessione a un'istanza database Oracle con un client SQL, consulta Connessione all'istanza database RDS per Oracle.

   b. Eseguire la seguente query:

      ```
      SELECT sqltxplain.sqlt$a.get_param('tool_version') sqlt_version FROM DUAL;
      ```

      La query restituisce la versione corrente dell'opzione SQLT in Amazon RDS. `12.1.160429` è un esempio di versione di SQLT disponibile in Amazon RDS.
5. Modificare le password degli utenti creati dall'opzione SQLT.

   a. Utilizzare un client SQL per effettuare la connessione all'istanza database come utente master.

   b. Eseguire la seguente istruzione SQL per modificare la password dell'utente `SQLTXADMIN`:

      ```
      ALTER USER SQLTXADMIN IDENTIFIED BY new_password ACCOUNT UNLOCK;
      ```

> **Note**
>
> Specifica una password diversa dal prompt mostrato qui come best practice per la sicurezza.

c. Eseguire la seguente istruzione SQL per modificare la password dell'utente `SQLTXPLAIN`:

```
ALTER USER SQLTXPLAIN IDENTIFIED BY new_password ACCOUNT UNLOCK;
```

> **Note**
>
> Specifica una password diversa dal prompt mostrato qui come best practice per la sicurezza.

> **Note**
>
> Per aggiornare SQLT, dovrai disinstallare la versione precedente di SQLT e installare quella nuova. Pertanto, durante l'aggiornamento di SQLT, tutti i metadati SQLT potrebbero andare perduti. Anche l'aggiornamento di una versione principale di un database la disinstallazione e la reinstallazione di SQLT. Un esempio di aggiornamento di una versione principale è quello da Oracle Database 12c Release 2 (12.2) a Oracle Database 19c.

Uso di SQLT

SQLT funziona con l'utilità Oracle SQL*Plus.

Per utilizzare SQLT

1. Scaricare il file .zip di SQLT da Document 215187.1 nel sito My Oracle Support.

> **Note**
>
> Non è possibile scaricare SQLT 12.1.160429 dal sito My Oracle Support. Oracle ha dichiarato obsoleta questa versione precedente.

2. Decomprimere il file .zip di SQLT.
3. Dal prompt dei comandi, portarsi sulla directory `sqlt/run` del file system.
4. Dal prompt dei comandi, aprire SQL*Plus ed effettuare la connessione all'istanza database come utente master.

   Per informazioni sulla connessione a un'istanza database Oracle con SQL*Plus, consulta Connessione all'istanza database RDS per Oracle.
5. Ottenere l'ID SQL di un'istruzione SQL:

   ```
   SELECT SQL_ID FROM V$SQL WHERE SQL_TEXT='sql_statement';
   ```

   L'output è simile a quello riportato di seguito:

   ```
   SQL_ID
   -------------
   chvsmttqjzjkn
   ```
6. Analizzare un'istruzione SQL con SQLT:

   ```
   START sqltxtract.sql sql_id sqltxplain_user_password
   ```

   Ad esempio, per l'ID SQL `chvsmttqjzjkn`, immettere quanto segue:

   ```
   START sqltxtract.sql chvsmttqjzjkn sqltxplain_user_password
   ```

SQLT genera un rapporto e le risorse correlate come un file. zip nella directory in cui è stato eseguito il comando SQLT.

7. (Opzionale) Per consentire agli utenti dell'applicazione di diagnosticare le istruzioni SQL con SQLT, concedere `SQLT_USER_ROLE` a ciascun utente con la seguente istruzione:

```
GRANT SQLT_USER_ROLE TO application_user_name;
```

> **Note**
>
> Oracle sconsiglia l'esecuzione di SQLT con l'utente `SYS` o con utenti che dispongono del ruolo `DBA`. La best practice prevede l'esecuzione della diagnostica SQLT con l'account dell'utente dell'applicazione, concedendo `SQLT_USER_ROLE` a tale utente.

Aggiornamento dell'opzione SQLT

Con Amazon RDS for Oracle puoi aggiornare l'opzione SQLT dalla versione attuale a una versione successiva. Per aggiornare l'opzione SQLT, completa le fasi 1–3 in Uso di SQLT per la nuova versione di SQLT. Inoltre, se hai concesso privilegi per la versione precedente di SQLT nella fase 7 di questa sezione, assegnali di nuovo per la nuova versione di SQLT.

L'aggiornamento dell'opzione SQLT causa la perdita dei metadati della versione precedente di SQLT. Lo schema e gli oggetti correlati della versione di SQLT precedente vengono eliminati e viene installata la versione più recente. Per ulteriori informazioni sulle modifiche nella versione più recente di SQLT, consulta il Documento 1614201.1 nel sito My Oracle Support.

> **Note**
>
> I downgrade di versione non sono supportati.

Modifica delle impostazioni SQLT

Dopo avere abilitato SQLT, puoi modificare le impostazioni `LICENSE_PACK` e `VERSION` dell'opzione.

Per ulteriori informazioni su come modificare le impostazioni dell'opzione, consulta Modifica di un'impostazione di un'opzione. Per ulteriori informazioni su ciascuna impostazione, consulta Impostazioni dell'opzione SQLT.

Rimozione dell'opzione SQLT

Puoi rimuovere SQLT da un'istanza database.

Per rimuovere SQLT da un'istanza database, esegui una delle seguenti procedure:

- Per rimuovere SQLT da più istanze database, rimuovi l'opzione SQLT dal gruppo di opzioni a cui appartengono le istanze database. Questa modifica coinvolge tutte le istanze database che usano il gruppo di opzioni. Per ulteriori informazioni, consulta Rimozione di un'opzione da un gruppo di opzioni.

- Per rimuovere SQLT da una singola istanza database, modifica l'istanza database e specifica un gruppo di opzioni diverso che non comprenda l'opzione SQLT. È possibile specificare il gruppo di opzioni predefinito (vuoto) o un gruppo di opzioni personalizzato diverso. Per ulteriori informazioni, consulta Modifica di un'istanza database Amazon RDS.

Oracle Statspack

L'opzione Oracle Statspack consente di installare e abilitare le statistiche sulle prestazioni di Oracle Statspack. Oracle Statspack è una raccolta di script SQL, PL/SQL e SQL*Plus che consente di raccogliere, archiviare e visualizzare i dati delle prestazioni. Per informazioni sull'utilizzo di Oracle Statspack, consulta la sezione Oracle Statspack della documentazione di Oracle.

> **Note**
>
> Oracle Statspack non è più supportato da Oracle ed è stato sostituito dal più avanzato Automatic Workload Repository (AWR). AWR è disponibile soltanto per i clienti con Oracle Enterprise Edition che abbiano acquistato il Diagnostics Pack. È possibile utilizzare Oracle Statspack con qualsiasi motore Oracle DB su Amazon RDS. Non è possibile eseguire Oracle Statspack sulle repliche di lettura Amazon RDS.

Impostazione di Oracle Statspack

Per eseguire gli script Statspack, è necessario aggiungere l'opzione Statspack.

Per impostare Oracle Statspack

1. In un client SQL, accedere al DB Oracle con un account amministrativo.
2. Eseguire una delle seguenti azioni, a seconda che Statspack sia installato o meno:
   - Se Statspack è installato e l'account `PERFSTAT` è associato a Statspack, andare al passaggio 4.
   - Se Statspack non è installato e l'account `PERFSTAT` esiste, eliminare l'account come segue:

     ```
     DROP USER PERFSTAT CASCADE;
     ```

     In caso contrario, il tentativo di aggiungere l'opzione Statspack genera un errore e `RDS-Event-0058`.
3. Aggiungere l'opzione Statspack a un gruppo di opzioni. Per informazioni, consulta Aggiunta di un'opzione a un gruppo di opzioni.

   Amazon RDS installa automaticamente gli script Statspack sull'istanza database e quindi imposta l'account `PERFSTAT`.

4. Reimpostare la password utilizzando la seguente istruzione SQL, sostituendo pwd con la nuova password:

```
ALTER USER PERFSTAT IDENTIFIED BY pwd ACCOUNT UNLOCK;
```

   È possibile accedere utilizzando l'account utente PERFSTAT ed eseguire gli script Statspack.

5. Effettuare una delle seguenti azioni, a seconda della versione del motore DB:

   - È possibile saltare questo passaggio se si utilizza Oracle Database 12c Release 2 (12.2) o inferiore.
   - Se si utilizza Oracle Database 19c o versione successiva, concedere il privilegio CREATE JOB all'account PERFSTAT utilizzando l'istruzione seguente:

```
GRANT CREATE JOB TO PERFSTAT;
```

6. Assicurarsi che gli eventi di attesa inattivi nella tabella PERFSTAT.STATS$IDLE_EVENT siano popolati.

   A causa del bug Oracle 28523746, gli eventi di attesa per inattività in PERFSTAT.STATS $IDLE_EVENT potrebbero non essere popolati. Per assicurarsi che tutti gli eventi inattivi siano disponibili, eseguire l'istruzione seguente:

```
INSERT INTO PERFSTAT.STATS$IDLE_EVENT (EVENT)
SELECT NAME FROM V$EVENT_NAME WHERE WAIT_CLASS='Idle'
MINUS
SELECT EVENT FROM PERFSTAT.STATS$IDLE_EVENT;
COMMIT;
```

Generazione di report Statspack

Un report Statspack confronta due snapshot.

Per generare report Statspack

1. In un client SQL, accedere al DB Oracle con l'account PERFSTAT.
2. Creare uno snapshot utilizzando una delle seguenti tecniche:

   - Creare manualmente uno snapshot Statspack.

- Creare un processo che accetti uno snapshot Statspack dopo un determinato intervallo di tempo. Ad esempio, il processo seguente crea uno snapshot Statspack ogni ora:

```
VARIABLE jn NUMBER;
exec dbms_job.submit(:jn, 'statspack.snap;',SYSDATE,'TRUNC(SYSDATE
+1/24,''HH24'')');
COMMIT;
```

3. Visualizzare gli snapshot utilizzando la seguente query:

```
SELECT SNAP_ID, SNAP_TIME FROM STATS$SNAPSHOT ORDER BY 1;
```

4. Eseguire la procedura `rdsadmin.rds_run_spreport` Amazon RDS, sostituendo begin_snap e end_snap con gli ID snapshot:

```
exec rdsadmin.rds_run_spreport(begin_snap,end_snap);
```

Il comando seguente, ad esempio, crea un report basato sull'intervallo compreso tra gli snapshot 1 e 2 di Statspack:

```
exec rdsadmin.rds_run_spreport(1,2);
```

Il nome del file del report Statspack include il numero dei due snapshot. Ad esempio, il file di report creato utilizzando gli snapshot 1 e 2 di Statspack sarà denominato `ORCL_spreport_1_2.lst`.

5. Monitorare l'output per eventuali errori.

Oracle Statspack esegue controlli prima di eseguire il report. Pertanto, è possibile visualizzare anche messaggi di errore nell'output del comando. Ad esempio, è possibile provare a generare un report basato su un intervallo non valido, in cui il valore iniziale dello snapshot Statspack è maggiore del valore finale. In questo caso, l'output mostra il messaggio di errore, ma il motore DB non genera un file di errore.

```
exec rdsadmin.rds_run_spreport(2,1);
*
ERROR at line 1:
ORA-20000: Invalid snapshot IDs. Find valid ones in perfstat.stats$snapshot.
```

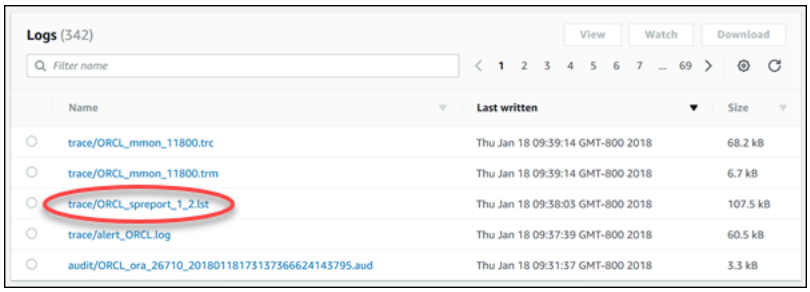
Se si utilizza un numero non valido per uno snapshot Statspack, l'output mostra un errore. Ad esempio, se si tenta di generare un report per gli snapshot 1 e 50, ma lo snapshot 50 non esiste, verrà visualizzato un errore.

```
exec rdsadmin.rds_run_spreport(1,50);
*
ERROR at line 1:
ORA-20000: Could not find both snapshot IDs
```

6. (Opzionale)

   Per recuperare il report, chiamare le procedure del file di traccia, come spiegato in Utilizzo di file di traccia Oracle.

   In alternativa, scaricare il report Statspack dalla console RDS. Vai alla sezione Log dei dettagli dell'istanza database e scegli Scarica:



   Se si verifica un errore durante la generazione di un report, il motore DB utilizza le stesse convenzioni di denominazione di un report ma con estensione err. Se, ad esempio, si verifica un errore durante la creazione di un report basato sugli snapshot 1 e 7 di Statspack, il file di report sarà denominato `ORCL_spreport_1_7.err`. È possibile scaricare il rapporto errori utilizzando le stesse tecniche di un report snapshot standard.

Rimozione delle istantanee Statspack

Utilizza il seguente comando per rimuovere un intervallo di istantanee Statspack:

```
exec statspack.purge(begin snap, end snap);
```

Fuso orario Oracle

Per modificare il fuso orario del sistema utilizzato dall'istanza database Oracle, utilizzare l'opsione del fuso orario. Ad esempio, potrebbe essere necessario modificare il fuso orario di un'istanza di database in modo che sia compatibile con un ambiente locale o con un'applicazione legacy. L'opzione del fuso orario modifica il fuso orario a livello di host. La modifica del fuso orario influisce su tutti i valori e su tutte le colonne della data, inclusi `SYSDATE` e `SYSTIMESTAMP`.

L'opzione del fuso orario è diversa dal comando `rdsadmin_util.alter_db_time_zone`. Il comando `alter_db_time_zone` modifica il fuso orario solo per determinati tipi di dati. L'opzione del fuso orario modifica il fuso orario per tutte le colonne e i valori della data. Per ulteriori informazioni su `alter_db_time_zone`, consulta Impostazione del fuso orario del database. Per ulteriori informazioni sulle considerazioni per l'aggiornamento, consulta Considerazioni sul fuso orario.

Considerazioni per l'impostazione del fuso orario

L'opzione del fuso orario è permanente e persistente. Pertanto, non è possibile completare le seguenti operazioni:

- Non è possibile rimuovere l'opzione da un gruppo di opzioni dopo averla aggiunta.
- Non è possibile eliminare il gruppo di opzioni da un'istanza database dopo averlo aggiunto.
- Non è possibile sostituire l'impostazione del fuso orario dell'opzione con un altro fuso orario.

Prima di aggiungere l'opzione del fuso orario al database di produzione, si consiglia vivamente di procedere come segue:

- Acquisisci uno snapshot dell'istanza database. Se imposti accidentalmente il fuso orario in modo errato, dovrai ripristinare l'istanza database alle impostazioni del fuso orario precedente. Per ulteriori informazioni, consulta Creazione di uno snapshot DB per un'istanza DB Single-AZ.
- Aggiungi l'opzione del fuso orario a un'istanza database di prova. L'aggiunta dell'opzione del fuso orario può causare problemi con le tabelle che utilizzano la data di sistema per aggiungere date o orari. Analizza i dati e le applicazioni dell'istanza di prova per valutare l'impatto della modifica del fuso orario nell'istanza di produzione.

Se la tua istanza DB utilizza il gruppo di opzioni predefinito, segui questi passaggi:

1. Acquisisci uno snapshot dell'istanza database.

2. Aggiungi l'opzione del fuso orario all'istanza database.

Se l'istanza DB attualmente utilizza un gruppo di opzioni non predefinito, segui questi passaggi:

1. Acquisisci uno snapshot dell'istanza database.
2. Crea un nuovo gruppo di opzioni.
3. Aggiungete l'opzione del fuso orario, insieme a tutte le altre opzioni attualmente associate al gruppo di opzioni esistente.

   Ciò impedisce la disinstallazione delle opzioni esistenti mentre si attiva l'opzione del fuso orario.
4. Aggiungi il gruppo di opzioni all'istanza database.

Impostazioni dell'opzione del fuso orario

Amazon RDS supporta le seguenti impostazioni per l'opzione del fuso orario.

Impostazione opzioni	Valori validi	Descrizione
`TIME_ZONE`	Uno dei fusi orari disponibili. Per l'elenco completo, consulta Fusi orari disponibili.	Il nuovo fuso orario per l'istanza di database.

Aggiunta dell'opzione del fuso orario

La procedura generale per aggiungere l'opzione del fuso orario a un'istanza di database è la seguente:

1. Creare un nuovo gruppo di opzioni oppure copiare o modificare un gruppo di opzioni esistente.
2. Aggiungere l'opzione al gruppo di opzioni.
3. Associare il gruppo di opzioni a questa istanza database.

Quando aggiungi l'opzione del fuso orario, si verifica una breve interruzione mentre l'istanza di database viene automaticamente riavviata.

Console

Per aggiungere l'opzione del fuso orario a un'istanza di database

1. Determinare il gruppo di opzioni che si desidera usare. È possibile creare un nuovo gruppo di opzioni oppure usare un gruppo di opzioni esistente. Se si desidera usare un gruppo di opzioni esistente, puoi passare alla fase successiva. In caso contrario, creare un gruppo di opzioni database personalizzato con le seguenti impostazioni:

   a. Per Engine (Motore) scegliere l'edizione Oracle per l'istanza database.

   b. In Major engine version (Versione principale del motore), scegliere la versione dell'istanza database.

   Per ulteriori informazioni, consulta Creazione di un gruppo di opzioni.

2. Aggiungere l'opzione Timezone (Fuso orario) al gruppo di opzioni e configurare le impostazioni di opzione.

   > **Important**
   >
   > Se aggiungi l'opzione del fuso orario a un gruppo di opzioni esistente già associato a una o più istanze di database, si verifica una breve interruzione mentre tutte le istanze di database vengono riavviate automaticamente.

   Per ulteriori informazioni sull'aggiunta di opzioni, consulta Aggiunta di un'opzione a un gruppo di opzioni. Per ulteriori informazioni su ciascuna impostazione, consulta Impostazioni dell'opzione del fuso orario.

3. Applicare il gruppo di opzioni a un'istanza database nuova o esistente:

   - Per una nuova istanza database, si applica il gruppo di opzioni quando viene avviata l'istanza. Per ulteriori informazioni, consulta Creazione di un'istanza database Amazon RDS.
   - Per un'istanza database esistente, viene applicato il gruppo di opzioni modificando l'istanza e collegando il nuovo gruppo di opzioni. Quando aggiungi l'opzione del fuso orario a un'istanza di database esistente, si verifica una breve interruzione mentre l'istanza di database viene automaticamente riavviata. Per ulteriori informazioni, consulta Modifica di un'istanza database Amazon RDS.

AWS CLI

L'esempio seguente utilizza il comando AWS CLI add-option-to-option-group per aggiungere l'`Timezone`opzione e l'impostazione dell'`TIME_ZONE`opzione a un gruppo di opzioni chiamato. `myoptiongroup` Il fuso orario è impostato su `Africa/Cairo`.

Per LinuxmacOS, oUnix:

```
aws rds add-option-to-option-group \
    --option-group-name "myoptiongroup" \
    --options "OptionName=Timezone,OptionSettings=[{Name=TIME_ZONE,Value=Africa/
Cairo}]" \
    --apply-immediately
```

Per Windows:

```
aws rds add-option-to-option-group ^
    --option-group-name "myoptiongroup" ^
    --options "OptionName=Timezone,OptionSettings=[{Name=TIME_ZONE,Value=Africa/
Cairo}]" ^
    --apply-immediately
```

Modifica delle impostazioni del fuso orario

L'opzione del fuso orario è permanente e persistente. Non è possibile rimuovere l'opzione da un gruppo di opzioni dopo averla aggiunta. Non è possibile eliminare il gruppo di opzioni da un'istanza di database dopo averlo aggiunto. Non è possibile sostituire l'impostazione del fuso orario dell'opzione con un altro fuso orario. Se imposti il fuso orario in modo errato, ripristina uno snapshot dell'istanza di database precedente all'aggiunta dell'opzione del fuso orario.

Rimozione dell'opzione del fuso orario

L'opzione del fuso orario è permanente e persistente. Non è possibile rimuovere l'opzione da un gruppo di opzioni dopo averla aggiunta. Non è possibile eliminare il gruppo di opzioni da un'istanza di database dopo averlo aggiunto. Per rimuovere l'opzione del fuso orario, ripristina uno snapshot dell'istanza di database precedente all'aggiunta dell'opzione del fuso orario.

Fusi orari disponibili

È possibile utilizzare i seguenti valori per l'opzione del fuso orario.

Zona	Time zone (Fuso orario)
Africa	Africa/Cairo, Africa/Casablanca, Africa/Harare, Africa/Lagos, Africa/Luanda, Africa/Monrovia, Africa/Nairobi, Africa/Tripoli, Africa/Windhoek
America	America/Araguaina, America/Argentina/Buenos_Aires, America/Asuncion, America/Bogota, America/Caracas, America/Chicago, America/Chihuahua, America/Cuiaba, America/Denver, America/Detroit, America/Fortaleza, America/Godthab, America/Guatemala, America/Halifax, America/Lima, America/Los_Angeles, America/Manaus, America/Matamoros, America/Mexico_City, America/Monterrey, America/Montevideo, America/New_York, America/Phoenix, America/Santiago, America/Sao_Paulo, America/Tijuana, America/Toronto
Asia	Asia/Amman, Asia/Ashgabat, Asia/Baghdad, Asia/Baku, Asia/Bangkok, Asia/Beirut, Asia/Calcutta, Asia/Damasco, Asia/Dhaka, Asia/Hong_Kong, Asia/Irkutsk, Asia/Giacarta, Asia/Gerusalemme, Asia/Kabul, Asia/Karachi, Asia/Kathmandu, Asia/Kolkata, Asia/Krasnoyarsk, Asia/Magadan, Asia/Manila, Asia/Muscat, Asia/Novosibirsk, Asia/Rangoon, Asia/Riyadh, Asia/Seul, Asia/Shanghai, Asia/Singapore, Asia/Taipei, Asia/Tehran, Asia/Tokyo, Asia/Ulaanbaatar, Asia/Vladivostok, Asia/Yakutsk, Asia/Yerevan
Atlantico	Atlantico/Azzorre, Atlantico/Cape_Verde
Australia	Australia/Adelaide, Australia/Brisbane, Australia/Darwin, Australia/Eucla, Australia/Hobart, Australia/Lord_Howe, Australia/Perth, Australia/Sydney
Brasile	Brasile/, Brasile/Est DeNoronha
Canada	Canada/Newfoundland, Canada/Saskatchewan
ecc	Ecc./GMT-3
Europa	Europa/Amsterdam, Europa/Atene, Europa/Berlino, Europa/Dublino, Europa/Helsinki, Europa/Kaliningrad, Europa/Londra, Europa/Madrid, Europa/Mosca, Europa/Parigi, Europa/Praga, Europa/Roma, Europa/Sarajevo

Zona	Time zone (Fuso orario)
Pacifico	Pacifico/Apia, Pacifico/Auckland, Pacifico/Chatham, Pacifico/Fiji, Pacifico/Guam, Pacifico/Honolulu, Pacifico/Kiritimati, Pacifico/Marquesas, Pacifico/Samoa, Pacifico/Tongatapu, Pacifico/Wake
USA	Stati Uniti/Alaska, Stati Uniti/Centrali, Stati Uniti/Est-Indiana, Stati Uniti/Orientali, Stati Uniti/Pacifico
UTC	UTC

Aggiornamento automatico dei file di fuso orario Oracle

Con l'`TIMEZONE_FILE_AUTOUPGRADE`opzione, puoi aggiornare il file del fuso orario corrente alla versione più recente sulla tua istanza DB RDS for Oracle.

Argomenti

Panoramica dei file di fuso orario di Oracle

Un file di fuso orario di Oracle Database include le seguenti informazioni:

- Differenza rispetto all'ora UTC (Coordinated Universal Time)
- Tempi di transizione per l'ora legale
- Abbreviazioni per ora standard e ora legale

Oracle Database fornisce più versioni dei file di fuso orario. Quando si crea un database Oracle in un ambiente On-Premise, si sceglie la versione del file di fuso orario. Per ulteriori informazioni, consulta Choosing a Time Zone File (Scelta di un file di fuso orario) in Oracle Database Globalization Support Guide (Guida al supporto per la globalizzazione di Oracle Database).

Se le regole per l'ora legale cambiano, Oracle pubblica nuovi file di fuso orario e Oracle rilascia questi nuovi file di fuso orario indipendentemente dalla pianificazione degli aggiornamenti trimestrali (Release Updates) e delle Release Update Revisions (RUR). I file del fuso orario si trovano sull'host del database nella directory `$ORACLE_HOME/oracore/zoneinfo/`. I nomi dei file di fuso orario utilizzano il formato DStV*versione*, come in DSTv35.

In che modo il file di fuso orario influisce sul trasferimento dei dati

In Oracle Database, il tipo di dati `TIMESTAMP WITH TIME ZONE` memorizza i dati di timestamp e fuso orario. Dati con il tipo di dati `TIMESTAMP WITH TIME ZONE` utilizzano le regole nella versione

del file di fuso orario associato. Pertanto, `TIMESTAMP WITH TIME ZONE` i dati esistenti vengono modificati quando si aggiorna il file del fuso orario.

Possono verificarsi problemi quando si trasferiscono i dati tra database che utilizzano versioni diverse del file del fuso orario. Ad esempio, se si importano dati da un database di origine con una versione del file con fuso orario superiore a quella del database di destinazione, il database genera l'`ORA-39405`errore. In precedenza si ovviava all'errore utilizzando una delle seguenti tecniche:

- Crea un'istanza database RDS per Oracle con il file di fuso orario desiderato, esporta i dati dal database di origine e quindi importali nel nuovo database.
- Utilizza DMS AWS o replica logica per eseguire la migrazione dei dati.

Aggiornamenti automatici con l'opzione TIMEZONE_FILE_AUTOUPGRADE

Quando il gruppo di opzioni collegato all'istanza DB di RDS for Oracle include l'`TIMEZONE_FILE_AUTOUPGRADE`opzione, RDS aggiorna automaticamente i file del fuso orario. Garantendo che i database Oracle utilizzino la stessa versione del file con fuso orario, si evitano procedure manuali dispendiose in termini di tempo quando si spostano i dati tra ambienti diversi. L'opzione `TIMEZONE_FILE_AUTOUPGRADE` supporta sia i database container (CDB) che quelli non CDB.

Quando aggiungi l'opzione `TIMEZONE_FILE_AUTOUPGRADE` al gruppo di opzioni, puoi scegliere se aggiungerla immediatamente o durante la finestra di manutenzione. *`Dopo che l'istanza DB ha applicato la nuova opzione, RDS verifica se è possibile installare un file di versione DStv più recente.`* L'impostazione di DSTv*`versione`* di destinazione dipende da quanto segue:

- La versione secondaria del motore attualmente in esecuzione sulla tua istanza database
- La versione secondaria del motore a cui desideri aggiornare la tua istanza database

Ad esempio, la versione corrente del file del fuso orario potrebbe essere DStv33. Quando RDS applica l'aggiornamento al gruppo di opzioni, potrebbe determinare che DStv34 è attualmente disponibile nel file system dell'istanza DB. RDS aggiornerà quindi automaticamente il file del fuso orario a DStv34.

Per trovare le versioni dell'ora legale disponibili negli aggiornamenti di rilascio RDS supportati, consulta le patch nelle Note di rilascio per Amazon Relational Database Service (Amazon RDS)

per Oracle. Ad esempio, la versione 19.0.0.0.ru-2022-10.rur-2022-10.r1 elenca la patch 34533061: RDBMS - DSTV39 UPDATE - TZDATA2022C.

Strategie per aggiornare il file del fuso orario

L'aggiornamento del motore DB e l'aggiunta dell'`TIMEZONE_FILE_AUTOUPGRADE`opzione a un gruppo di opzioni sono operazioni separate. L'aggiunta dell'`TIMEZONE_FILE_AUTOUPGRADE`opzione avvia l'aggiornamento del file del fuso orario se ne è disponibile uno più recente. I seguenti comandi vengono eseguiti (vengono visualizzate solo le opzioni pertinenti) immediatamente o nella finestra di manutenzione successiva:

- Aggiorna il tuo motore DB solo utilizzando il seguente comando CLI RDS:

```
modify-db-instance --engine-version name ...
```

- Aggiungi l'`TIMEZONE_FILE_AUTOUPGRADE`opzione solo utilizzando il seguente comando CLI:

```
add-option-to-option-group --option-group-name name --options
 OptionName=TIMEZONE_FILE_AUTOUPGRADE ...
```

- Aggiorna il tuo motore DB e aggiungi un nuovo gruppo di opzioni all'istanza utilizzando il seguente comando CLI:

```
modify-db-instance --engine-version name --option-group-name name ...
```

La strategia di aggiornamento dipende dal fatto che si desideri aggiornare il database e il file del fuso orario insieme o eseguire solo una di queste operazioni. Tieni presente che se aggiorni il tuo gruppo di opzioni e poi aggiorni il motore DB in operazioni API separate, è possibile che sia attualmente in corso un aggiornamento del file del fuso orario durante l'aggiornamento del motore DB.

Per gli esempi in questa sezione si assume quanto riportato di seguito:

- Non hai ancora aggiunto nulla `TIMEZONE_FILE_AUTOUPGRADE` al gruppo di opzioni attualmente associato alla tua istanza DB.
- L'istanza database utilizza il database versione 19.0.0.0.ru-2019-07.rur-2019-07.r1 e il file di fuso orario DSTv33.
- Il file di sistema dell'istanza database include il file DSTv34.
- L'aggiornamento di rilascio 19.0.0.0.ru-2022-10.rur-2022-10.r1 include DSTv35.

Per aggiornare il file di fuso orario, è possibile utilizzare le seguenti strategie.

Argomenti

Aggiornamento del file di fuso orario senza aggiornare il motore

In questo scenario, il database utilizza DSTv33, ma è disponibile DSTv34 nel file system dell'istanza database. Vuoi aggiornare il file del fuso orario utilizzato dalla tua istanza database da DSTv33 a DSTv34, ma non vuoi aggiornare il motore a una nuova versione secondaria, che include DSTv35.

In un `add-option-to-option-group` comando, aggiungi `TIMEZONE_FILE_AUTOUPGRADE` al gruppo di opzioni utilizzato dall'istanza DB. Specifica se vuoi aggiungere l'opzione immediatamente o durante la finestra di manutenzione. Dopo aver applicato l'`TIMEZONE_FILE_AUTOUPGRADE`opzione, RDS effettua le seguenti operazioni:

1. Verifica la presenza di una nuova versione dell'ora legale.
2. Determina che DStv34 è disponibile nel file system.
3. Aggiorna immediatamente il file del fuso orario.

Aggiornamento della versione del file del fuso e del motore di database

In questo scenario, il database utilizza DSTv33, ma è disponibile DSTv34 nel file system dell'istanza database. Desideri aggiornare il motore di database alla versione secondaria 19.0.0.0.ru-2022-10.rur-2022-10.r1, che include DSTv35, e il file del fuso orario a DSTv35 durante l'aggiornamento del motore. Pertanto, il tuo obiettivo è saltare DSTv34 e aggiornare i file del fuso orario direttamente a DSTv35.

Per aggiornare contemporaneamente il motore e il file del fuso orario, esegui `modify-db-instance` con le `--engine-version` opzioni `--option-group-name` and. È possibile eseguire il comando immediatamente o rimandarlo alla finestra di manutenzione. `In --option-group-name`, specifica un gruppo di opzioni che includa l'`TIMEZONE_FILE_AUTOUPGRADE`opzione. Per esempio:

```
aws rds modify-db-instance
    --db-instance-identifier my-instance \
```

```
    --engine-version new-version \
    ----option-group-name og-with-timezone-file-autoupgrade \
    --apply-immediately
```

RDS inizia ad aggiornare il motore alla versione 19.0.0.0.ru-2022-10.rur-2022-10.r1. Dopo aver applicato l'`TIMEZONE_FILE_AUTOUPGRADE`opzione, RDS verifica la presenza di una nuova versione DST, verifica che DStv35 sia disponibile in 19.0.0.0.ru-2022-10.rur-2022-10.r1 e avvia immediatamente l'aggiornamento a DStv35.

Per aggiornare immediatamente il motore e quindi aggiornare il file del fuso orario, esegui le operazioni in sequenza:

1. Aggiorna il tuo motore DB solo utilizzando il seguente comando CLI:

   ```
   aws rds modify-db-instance \
       --db-instance-identifier my-instance \
       --engine-version new-version \
       --apply-immediately
   ```

2. Aggiungi l'`TIMEZONE_FILE_AUTOUPGRADE`opzione al gruppo di opzioni collegato alla tua istanza utilizzando il seguente comando CLI:

   ```
   aws rds add-option-to-option-group \
       --option-group-name og-in-use-by-your-instance \
       --options OptionName=TIMEZONE_FILE_AUTOUPGRADE \
       --apply-immediately
   ```

Aggiornamento della versione del motore di database senza aggiornare il file del fuso orario

In questo scenario, il database utilizza DSTv33, ma è disponibile DSTv34 nel file system dell'istanza database. Vuoi aggiornare il motore di database alla versione 19.0.0.0.ru-2022-10.rur-2022-10.r1, che include DSTv35, ma mantenere DSTv33 per il file del fuso orario. Puoi scegliere questa strategia per i seguenti motivi:

- I tuoi dati non utilizzano il tipo di dati `TIMESTAMP WITH TIME ZONE`.
- I tuoi dati utilizzano il tipo di dati `TIMESTAMP WITH TIME ZONE`, ma i dati non sono interessati dalle modifiche del fuso orario.
- Si desidera posticipare l'aggiornamento del file di fuso orario perché non è possibile tollerare il tempo di inattività aggiuntivo.

La strategia dipende da quale delle seguenti condizioni sono vere:

- L'istanza database non è associata a un gruppo di opzioni che include `TIMEZONE_FILE_AUTOUPGRADE`. Nel `modify-db-instance` comando, non specificate un nuovo gruppo di opzioni in modo che RDS non aggiorni il file del fuso orario.
- L'istanza DB è attualmente associata a un gruppo di opzioni che include`TIMEZONE_FILE_AUTOUPGRADE`. Con un solo `modify-db-instance` comando, associa l'istanza DB a un gruppo di opzioni che non include `TIMEZONE_FILE_AUTOUPGRADE` e aggiorna il motore di database a 19.0.0.0.ru-2022-10.rur-2022-10.r1.

Tempo di inattività durante l'aggiornamento del file di fuso orario

Quando RDS aggiorna il file di fuso orario, i dati che utilizzano `TIMESTAMP WITH TIME ZONE` potrebbero cambiare e in questo caso, è opportuno tenere in considerazione il tempo di inattività.

> **Warning**
>
> Se aggiungi l'opzione `TIMEZONE_FILE_AUTOUPGRADE`, l'aggiornamento del motore potrebbe avere tempi di inattività prolungati. L'aggiornamento dei dati del fuso orario per un database di grandi dimensioni potrebbe richiedere ore o addirittura giorni.

La durata dell'aggiornamento del file di fuso orario dipende da fattori come i seguenti:

- La quantità di dati `TIMESTAMP WITH TIME ZONE` nel database
- La configurazione dell'istanza database
- La classe dell’istanza database
- La configurazione dell’archivio
- La configurazione del database
- Le impostazioni dei parametri di database

Possono verificarsi ulteriori tempi di inattività quando si eseguono le seguenti operazioni:

- Aggiungere l'opzione al gruppo di opzioni quando l'istanza database utilizza un file di fuso orario non aggiornato

- Aggiornare il modulo di gestione di database Oracle quando la nuova versione del motore contiene una nuova versione del file di fuso orario

> **Note**
>
> Durante l'aggiornamento del file di fuso orario, RDS per Oracle chiama `PURGE DBA_RECYCLEBIN`.

Preparazione all'aggiornamento del file di fuso orario

L'aggiornamento di un file di fuso orario prevede due fasi distinte: preparazione e aggiornamento. Anche se non richiesto, è fortemente consigliabile eseguire la fase di preparazione. In questo passaggio, è possibile scoprire quali dati saranno interessati dall'esecuzione della procedura PL/SQL `DBMS_DST.FIND_AFFECTED_TABLES`. Per ulteriori informazioni sulla finestra di preparazione, consulta Aggiornamento del file di fuso orario e del timestamp con dati di fuso orario nella documentazione di Oracle Database.

Per preparare l'aggiornamento del file di fuso orario

1. Connettersi al database Oracle usando un client SQL.
2. Determinare la versione corrente del file di fuso orario utilizzata.

```
SELECT * FROM V$TIMEZONE_FILE;
```

3. Determinare la versione più recente del file di fuso orario disponibile nell'istanza database. Questo passaggio è applicabile solo se si utilizza Oracle Database 12c Release 2 (12.2) o versioni successive.

```
SELECT DBMS_DST.GET_LATEST_TIMEZONE_VERSION FROM DUAL;
```

4. Determinare la dimensione totale delle tabelle con colonne di tipo `TIMESTAMP WITH LOCAL TIME ZONE` o `TIMESTAMP WITH TIME ZONE`.

```
SELECT SUM(BYTES)/1024/1024/1024 "Total_size_w_TSTZ_columns_GB"
FROM   DBA_SEGMENTS
WHERE  SEGMENT_TYPE LIKE 'TABLE%'
AND    (OWNER, SEGMENT_NAME) IN
         (SELECT OWNER, TABLE_NAME
```

```
          FROM   DBA_TAB_COLUMNS
          WHERE  DATA_TYPE LIKE 'TIMESTAMP%TIME ZONE');
```

5. Determinare i nomi e le dimensioni dei segmenti con colonne di tipo `TIMESTAMP WITH LOCAL TIME ZONE` o `TIMESTAMP WITH TIME ZONE`.

```
SELECT OWNER, SEGMENT_NAME, SUM(BYTES)/1024/1024/1024
 "SEGMENT_SIZE_W_TSTZ_COLUMNS_GB"
FROM   DBA_SEGMENTS
WHERE  SEGMENT_TYPE LIKE 'TABLE%'
AND    (OWNER, SEGMENT_NAME) IN
         (SELECT OWNER, TABLE_NAME
          FROM   DBA_TAB_COLUMNS
          WHERE  DATA_TYPE LIKE 'TIMESTAMP%TIME ZONE')
GROUP BY OWNER, SEGMENT_NAME;
```

6. Eseguire la fase di preparazione.

   - La procedura `DBMS_DST.CREATE_AFFECTED_TABLE` crea una tabella per l'archiviazione dei dati interessati. Passare il nome di questa tabella alla procedura `DBMS_DST.FIND_AFFECTED_TABLES`. Per ulteriori informazioni, consulta Procedura CREATE_AFFECTED_TABLE nella documentazione del database Oracle.
   - La procedura `CREATE_ERROR_TABLE` crea una tabella per registrare gli errori. Per ulteriori informazioni, consulta Procedura CREATE_ERROR_TABLE nella documentazione del database Oracle.

   Nell'esempio seguente vengono create le tabelle dei dati e degli errori interessati e vengono trovate tutte le tabelle interessate.

```
EXEC DBMS_DST.CREATE_ERROR_TABLE('my_error_table')
EXEC DBMS_DST.CREATE_AFFECTED_TABLE('my_affected_table')

EXEC DBMS_DST.BEGIN_PREPARE(new_version);
EXEC DBMS_DST.FIND_AFFECTED_TABLES('my_affected_table', TRUE, 'my_error_table');
EXEC DBMS_DST.END_PREPARE;

SELECT * FROM my_affected_table;
SELECT * FROM my_error_table;
```

7. Eseguire una query sulle tabelle interessate e di errore.

```
SELECT * FROM my_affected_table;
SELECT * FROM my_error_table;
```

Aggiunta dell'opzione di aggiornamento automatico del file di fuso orario

Quando aggiungi l'opzione, il gruppo di opzioni si trova in una delle seguenti condizioni:

- Un gruppo di opzioni esistente è attualmente collegato ad almeno un'istanza database. Quando aggiungi l'opzione, tutte le istanze database che utilizzano il gruppo di opzioni si riavviano automaticamente. Questo riavvio causa una breve interruzione.
- Un gruppo di opzioni esistente non è collegato a un'istanza database. Dovrai aggiungere l'opzione e quindi associare il gruppo di opzioni esistente alle istanze database esistenti o a una nuova istanza database.
- Crei un nuovo gruppo di opzioni e aggiungi l'opzione. Dovrai associare il nuovo gruppo di opzioni alle istanze database esistenti o a una nuova istanza database.

Console

Per aggiungere l'opzione del file di fuso orario a un'istanza di database

1. Accedi alla AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel pannello di navigazione scegliere Option groups (Gruppi di opzioni).
3. Determinare il gruppo di opzioni che si desidera usare. È possibile creare un nuovo gruppo di opzioni oppure usare un gruppo di opzioni esistente. Se si desidera usare un gruppo di opzioni esistente, puoi passare alla fase successiva. In caso contrario, creare un gruppo di opzioni database personalizzato con le seguenti impostazioni:

   a. Per Engine (Motore), scegli la versione Oracle Database per l'istanza database.
   b. In Major engine version (Versione principale del motore), scegliere la versione dell'istanza database.

   Per ulteriori informazioni, consulta Creazione di un gruppo di opzioni.
4. Selezionare il gruppo di opzioni che si vuole modificare, quindi scegliere Add Option (Aggiungi opzione).

5. Nella finestra Add option (Aggiungi opzione) eseguire queste operazioni:

    a. Scegli TIMEZONE_FILE_AUTOUPGRADE.

    b. Per abilitare l'opzione in tutte le istanze database associate non appena viene aggiunta, per Apply Immediately (Applica immediatamente) scegliere Yes (Sì). Se si sceglie No (impostazione predefinita), l'opzione viene abilitata per ogni istanza database associata durante la finestra di manutenzione successiva.

6. Dopo aver selezionato le impostazioni desiderate, selezionare Add Option (Aggiungi opzione).

AWS CLI

L'esempio seguente utilizza il comando -group per aggiungere l'opzione a un gruppo di opzioni chiamato. AWS CLI add-option-to-option `TIMEZONE_FILE_AUTOUPGRADE` `myoptiongroup`

Per LinuxmacOS, oUnix:

```
aws rds add-option-to-option-group \
    --option-group-name "myoptiongroup" \
    --options "OptionName=TIMEZONE_FILE_AUTOUPGRADE" \
    --apply-immediately
```

Per Windows:

```
aws rds add-option-to-option-group ^
    --option-group-name "myoptiongroup" ^
    --options "OptionName=TIMEZONE_FILE_AUTOUPGRADE" ^
    --apply-immediately
```

Controllo dei dati dopo l'aggiornamento del file di fuso orario

Si consiglia di controllare i dati dopo aver aggiornato il file del fuso orario. Durante la fase di preparazione, RDS per Oracle crea automaticamente le tabelle riportate di seguito:

- `rdsadmin.rds_dst_affected_tables` – Elenca le tabelle che contengono dati interessati dall'aggiornamento
- `rdsadmin.rds_dst_error_table` – Elenca gli errori generati durante l'aggiornamento

Queste tabelle sono indipendenti da tutte le tabelle create nella finestra di preparazione. Per visualizzare i risultati dell'aggiornamento, eseguire una query sulle tabelle come indicato di seguito.

```
SELECT * FROM rdsadmin.rds_dst_affected_tables;
SELECT * FROM rdsadmin.rds_dst_error_table;
```

Per ulteriori informazioni sullo schema per i dati interessati e le tabelle di errore, consulta Procedura FIND_AFFECTED_TABLE nella documentazione Oracle.

Oracle Transparent Data Encryption

Amazon RDS supporta Oracle Transparent Data Encryption (TDE), una caratteristica dell'opzione Oracle Advanced Security disponibile in Oracle Enterprise Edition. Tale caratteristica consente la crittografia automatica dei dati prima che vengano trascritti nello storage e la loro decriptazione automatica durante la lettura dallo storage. Questa opzione è supportata solo per il modello Bring Your Own License (BYOL).

TDE è utile negli scenari in cui è necessario crittografare i dati sensibili nel caso in cui i file di dati e i backup vengano ottenuti da terze parti. TDE è utile anche quando è necessario rispettare le normative relative alla sicurezza.

L'`TDE`opzione è persistente e permanente. Se associ l'istanza DB di RDS for Oracle a un gruppo di opzioni con l'`TDE`opzione abilitata, non puoi disabilitarla. È possibile modificare il gruppo di opzioni, ma il nuovo gruppo di opzioni deve includere l'`TDE`opzione. Per ulteriori informazioni sulle opzioni permanenti e permanenti, vedereOpzioni persistenti e permanenti.

> **Note**
>
> Non è possibile condividere un'istantanea del DB che utilizza l'opzione TDE. Per ulteriori informazioni sulla condivisione di snapshot DB, consulta Condivisione di uno snapshot del database.

Una spiegazione dettagliata su TDE in Oracle Database non rientra nell'ambito di questa guida. Per informazioni, consulta le seguenti risorse del database Oracle:

- Protezione dei dati archiviati utilizzando Transparent Data Encryption nella documentazione di Oracle Database
- Sicurezza avanzata di Oracle nella documentazione di Oracle Database
- Le migliori pratiche di crittografia dei dati trasparenti per la sicurezza avanzata di Oracle, un white paper di Oracle

Per ulteriori informazioni sull'utilizzo di TDE con RDS per Oracle, consulta i seguenti blog:

- Opzioni di crittografia del database Oracle su Amazon RDS
- Esegui la migrazione di un'istanza DB Amazon RDS for Oracle compatibile con TDE su più account con tempi di inattività ridotti utilizzando AWS DMS

Modalità di crittografia di TDE

Oracle Transparent Data Encryption supporta due modalità di crittografia: la crittografia TDE degli spazi tabelle e la crittografia TDE delle colonne. La crittografia TDE degli spazi tabelle si utilizza per crittografare intere tabelle di un'applicazione. La crittografia TDE delle colonne si utilizza per crittografare singole informazioni che contengono dati sensibili. Puoi anche applicare una soluzione di crittografia ibrida che utilizza sia la codifica TDE degli spazi tabelle che quella delle colonne.

> **Note**
>
> Amazon RDS gestisce Oracle Wallet e la chiave principale di TDE per l'istanza database. Non dovrai impostare la chiave crittografica con il comando `ALTER SYSTEM set encryption key`.

Dopo aver abilitato l'`TDE`opzione, puoi controllare lo stato di Oracle Wallet utilizzando il seguente comando:

```
SELECT * FROM v$encryption_wallet;
```

Per creare uno spazio tabelle crittografato, utilizza il comando seguente:

```
CREATE TABLESPACE encrypt_ts ENCRYPTION DEFAULT STORAGE (ENCRYPT);
```

Utilizza il comando seguente per specificare l'algoritmo di crittografia:

```
CREATE TABLESPACE encrypt_ts ENCRYPTION USING 'AES256' DEFAULT STORAGE (ENCRYPT);
```

Le istruzioni precedenti per la crittografia di un tablespace sono le stesse che si utilizzerebbero su un database Oracle locale.

Determinare se l'istanza DB utilizza TDE

Potresti voler determinare se la tua istanza DB è associata a un gruppo di opzioni con l'`TDE`opzione abilitata. Per visualizzare il gruppo di opzioni a cui è associata un'istanza DB, utilizzare la console RDS, il describe-db-instance AWS CLI comando o l'operazione API DescribedBInstances.

Aggiunta dell'opzione TDE

La procedura per utilizzare Oracle Transparent Data Encryption (TDE) con Amazon RDS è la seguente:

1. Se l'istanza DB non è associata a un gruppo di opzioni con l'`TDE`opzione abilitata, è necessario creare un gruppo di opzioni e aggiungere l'`TDE`opzione o modificare il gruppo di opzioni associato per aggiungere l'opzione. `TDE` Per informazioni sulla creazione o la modifica di un gruppo di opzioni, consulta Uso di gruppi di opzioni. Per informazioni sull'aggiunta di un'opzione a un gruppo di opzioni, consulta Aggiunta di un'opzione a un gruppo di opzioni.
2. Associa l'istanza database con il gruppo di opzioni contenente l'opzione TDE. Per informazioni su come associare un'istanza database con un gruppo di opzioni, consulta Modifica di un'istanza database Amazon RDS.

Copiare i dati in un'istanza DB che non include l'opzione TDE

Non è possibile rimuovere l'opzione TDE dall'istanza DB o associarla a un gruppo di opzioni che non include l'opzione TDE. Per migrare i dati su un'istanza che non include l'opzione TDE, procedi come segue:

1. Decrittografa i dati sulla tua istanza DB.
2. Copia i dati in una nuova istanza DB non associata a un gruppo di opzioni abilitato`TDE`.
3. Eliminare l'istanza DB originale.

Puoi rinominare la nuova istanza con lo stesso nome dell'istanza database precedente, se lo desideri.

Utilizzo di TDE con Oracle Data Pump

Puoi utilizzare Oracle Data Pump per importare o esportare file dump crittografati. Amazon RDS supporta la modalità di crittografia delle password `(ENCRYPTION_MODE=PASSWORD)` per Oracle Data Pump. Amazon RDS non supporta la modalità di crittografia trasparente `(ENCRYPTION_MODE=TRANSPARENT)` per Oracle Data Pump. Per ulteriori informazioni, consulta Importazione utilizzando Oracle Data Pump.

UTL_MAIL di Oracle

Amazon RDS supporta UTL_MAIL di Oracle attraverso l'uso dell'opzione UTL_MAIL e di server SMTP. Puoi inviare un messaggio e-mail direttamente dal database usando il pacchetto UTL_MAIL. Amazon RDS supporta UTL_MAIL per le seguenti versioni di Oracle:

- Oracle Database 21c (21.0.0.0), tutte le versioni
- Oracle Database 19c (19.0.0.0), tutte le versioni
- Oracle Database 12c Release 2 (12.2), tutte le versioni
- Oracle Database 12c Release 1 (12.1), versione 12.1.0.2.v5 e successive

Di seguito trovi alcune delle limitazioni all'utilizzo di UTL_MAIL:

- UTL_MAIL non supporta il protocollo Transport Layer Security (TLS) e le e-mail non vengono pertanto crittografate.

  Per connetterti in modo sicuro a risorse SSL/TLS remote creando e caricando wallet Oracle personalizzati, segui le istruzioni in Configurazione dell'accesso UTL_HTTP utilizzando certificati e un portafoglio Oracle.

  I certificati specifici necessari per il wallet variano in base al servizio. Per i servizi AWS, sono in genere disponibili nel repository di Amazon Trust Services.
- UTL_MAIL non supporta l'autenticazione con i server SMTP.
- È possibile inviare un solo allegato in una e-mail.
- Non è possibile inviare allegati di dimensioni maggiori di 32 K.
- È possibile utilizzare solo le codifiche di caratteri ASCII ed EBCDIC (Extended Binary Coded Decimal Interchange Code).
- La porta SMTP (25) è soggetta a throttling in base alle policy del proprietario dell'interfaccia di rete elastica.

Quando abiliti UTL_MAIL, il privilegio di esecuzione è concesso solo all'utente master dell'istanza database. Se necessario, l'utente master può concedere il privilegio di esecuzione ad altri utenti, per consentire loro di utilizzare UTL_MAIL.

> **Important**
>
> Ti consigliamo di abilitare la funzionalità di controllo integrata di Oracle per tenere traccia dell'utilizzo delle procedure di UTL_MAIL.

Prerequisiti per UTL_MAIL di Oracle

Di seguito sono indicati i prerequisiti per l'utilizzo di UTL_MAIL di Oracle:

- Uno o più server SMTP e i corrispondenti indirizzi IP o nomi DNS (Domain Name Server) pubblici o privati. Per ulteriori informazioni sui nomi DNS privati risolti tramite un server DNS personalizzato, consulta Impostazione di un server DNS personalizzato.
- Per le versioni precedenti a Oracle 12c, l'istanza database deve utilizzare anche l'opzione XML DB. Per ulteriori informazioni, consulta Oracle XML DB.

Aggiunta dell'opzione UTL_MAIL di Oracle

La procedura generale per aggiungere l'opzione UTL_MAIL di Oracle a un'istanza database è la seguente:

1. Creare un nuovo gruppo di opzioni oppure copiare o modificare un gruppo di opzioni esistente.
2. Aggiungere l'opzione al gruppo di opzioni.
3. Associare il gruppo di opzioni a questa istanza database.

Dopo essere stata aggiunta, l'opzione UTL_MAIL diventa attiva non appena il gruppo di opzioni è attivo.

Per aggiungere l'opzione UTL_MAIL a un'istanza database

1. Determinare il gruppo di opzioni che si desidera usare. È possibile creare un nuovo gruppo di opzioni oppure usare un gruppo di opzioni esistente. Se si desidera usare un gruppo di opzioni esistente, puoi passare alla fase successiva. In caso contrario, creare un gruppo di opzioni database personalizzato con le seguenti impostazioni:

   a. In Engine (Motore) scegliere l'edizione di Oracle che si desidera utilizzare.

b. In Major engine version (Versione principale del motore), scegliere la versione dell'istanza database.

Per ulteriori informazioni, consulta Creazione di un gruppo di opzioni.

2. Aggiungere l'opzione UTL_MAIL al gruppo di opzioni. Per ulteriori informazioni sull'aggiunta di opzioni, consulta Aggiunta di un'opzione a un gruppo di opzioni.
3. Applicare il gruppo di opzioni a un'istanza database nuova o esistente:

   - Per una nuova istanza database, si applica il gruppo di opzioni quando viene avviata l'istanza. Per ulteriori informazioni, consulta Creazione di un'istanza database Amazon RDS.
   - Per un'istanza database esistente, viene applicato il gruppo di opzioni modificando l'istanza e collegando il nuovo gruppo di opzioni. Per ulteriori informazioni, consulta Modifica di un'istanza database Amazon RDS.

Utilizzo di UTL_MAIL di Oracle

Dopo aver abilitato l'opzione UTL_MAIL, dovrai configurare il server SMTP per poter iniziare a utilizzarlo.

Per configurare il server SMTP, è necessario impostare il parametro SMTP_OUT_SERVER su un indirizzo IP o un nome DNS pubblico valido. Per il parametro SMTP_OUT_SERVER, puoi specificare un elenco separato da virgole di indirizzi di più server. Se il primo server non è disponibile, UTL_MAIL prova a utilizzare il server successivo e così via.

È possibile impostare il parametro SMTP_OUT_SERVER predefinito per un'istanza database utilizzando un gruppo di parametri database. Puoi impostare il parametro SMTP_OUT_SERVER per una sessione eseguendo il codice riportato di seguito sul database nell'istanza database.

```
ALTER SESSION SET smtp_out_server = mailserver.domain.com:25;
```

Quando l'opzione UTL_MAIL è abilitata e SMTP_OUT_SERVER è configurato, potrai inviare e-mail tramite la procedura `SEND`. Per ulteriori informazioni, consulta UTL_MAIL nella documentazione di Oracle.

Rimozione dell'opzione UTL_MAIL di Oracle

È possibile rimuovere l'opzione UTL_MAIL di Oracle da un'istanza database.

Per rimuovere UTL_MAIL da un'istanza database, procedi in uno dei seguenti modi:

- Per rimuovere l'opzione UTL_MAIL da più istanze database, rimuovila dal gruppo di opzioni a cui le istanze appartengono. Questa modifica coinvolge tutte le istanze database che usano il gruppo di opzioni. Per ulteriori informazioni, consulta Rimozione di un'opzione da un gruppo di opzioni.
- Per rimuovere l'opzione UTL_MAIL da una singola istanza database, modifica l'istanza database e specifica un gruppo di opzioni diverso che non comprenda l'opzione UTL_MAIL. È possibile specificare il gruppo di opzioni predefinito (vuoto) o un gruppo di opzioni personalizzato diverso. Per ulteriori informazioni, consulta Modifica di un'istanza database Amazon RDS.

Risoluzione dei problemi

Di seguito sono elencati i problemi che si potrebbero riscontrare quando si utilizza UTL_MAIL con Amazon RDS.

- Throttling. La porta SMTP (25) è soggetta a throttling in base alle policy del proprietario dell'interfaccia di rete elastica. Se riesci a inviare e-mail utilizzando UTL_MAIL e viene visualizzato l'errore `ORA-29278: SMTP transient error: 421 Service not available`, è possibile che sia in corso il throttling. Se la distribuzione di posta elettronica è soggetta a throttling, ti consigliamo di implementare un algoritmo di backoff. Per ulteriori informazioni sugli algoritmi di backoff, consulta Ripetizione dei tentativi in caso di errore e backoff esponenziale in AWS e il post Come gestire un errore di superamento della frequenza massima in uscita con conseguente throttling.

  È possibile richiedere che venga rimosso il throttling. Per ulteriori informazioni, leggi Come rimuovere il throttling sulla porta 25 per l'istanza EC2.

Oracle XML DB

Oracle XML DB aggiunge un supporto XML nativo all'istanza database. Con XML DB, è possibile archiviare e recuperare dati XML e relazionali strutturati o non strutturati. Il server del protocollo XML DB non è supportato su RDS for Oracle.

XML DB è preinstallato su Oracle Database 12c e versioni successive. Pertanto, non è necessario utilizzare un gruppo di opzioni per installare esplicitamente XML DB come funzionalità aggiuntiva.

Per informazioni su come configurare e utilizzare XML DB, vedere Oracle XML DB Developer's Guide nella documentazione di Oracle Database.

Aggiornamento del motore di database RDS per Oracle

Quando Amazon RDS supporta una nuova versione di Oracle Database, puoi effettuare l'aggiornamento delle istanze database alla nuova versione. Per informazioni sulle versioni di Oracle disponibili in Amazon RDS, consulta le Note di rilascio di Amazon RDS for Oracle.

> **Important**
>
> I database RDS per Oracle 11g, 12c e 18c non sono più supportati. Se sono stati conservati snapshot di database Oracle 11g, 12c o 18c, sarà possibile aggiornarli a una versione successiva. Per ulteriori informazioni, consulta Aggiornamento di uno shapshot DB Oracle.

Argomenti

Panoramica sugli aggiornamenti del motore di database Oracle

Prima di aggiornare l'istanza database RDS per Oracle, prova a familiarizzare con i seguenti concetti chiave.

Argomenti

- Aggiornamenti Oracle delle istanze di Micro DB

Aggiornamenti delle versioni principali e secondarie

Le versioni principali sono versioni di Oracle Database che vengono rilasciate ogni 1-2 anni. Esempi di versioni principali sono Oracle Database 19c e Oracle Database 21c.

Le versioni secondarie, chiamate anche Release Update (RU), vengono in genere rilasciate da Oracle ogni trimestre. Le versioni secondarie contengono piccoli miglioramenti alle funzioni e correzioni di bug. Esempi di versioni secondarie sono 21.0.0.0.ru-2023-10.rur-2023-10.r1 e 19.0.0.0.ru-2023-10.rur-2023-10.r1. Per ulteriori informazioni, consulta Note di rilascio per Amazon Relational Database Service (Amazon RDS) per Oracle.

RDS per Oracle supporta i seguenti aggiornamenti a un'istanza database:

Tipo di aggiornamento	Compatibilità delle applicazioni	Metodi di aggiornamento	Percorso di aggiornamento di esempio
Versione principale	Un aggiornamento della versione principale può introdurre modifiche che non sono compatibili con le applicazioni esistenti.	Solo manuale	Da Oracle Database 19c a Oracle Database 21c
Versione secondaria	Un aggiornamento della versione secondaria include solo modifiche compatibili con le versioni precedenti delle applicazioni esistenti.	Automatico o manuale	Da 21.0.0.0.ru-2023-07.rur-2022-07.r1 a 21.0.0.0.ru-2023-10.rur-2022-10.r1

Important

Quando aggiorni il motore database, si verifica un'interruzione. Il tempo di interruzione dipende dalla versione del motore e dalle dimensioni dell'istanza database.
Accertati di testare in modo approfondito qualsiasi aggiornamento per verificare che le tue applicazioni funzionino correttamente prima di applicare l'aggiornamento ai database di

produzione. Per ulteriori informazioni, consulta Verifica di un aggiornamento del database Oracle.

Date di supporto previste per le versioni principali di RDS per Oracle

Le versioni principali di RDS per Oracle restano disponibili almeno fino alla data di fine del supporto per la versione di rilascio di Oracle Database corrispondente. È possibile utilizzare le date seguenti per pianificare i cicli di test e aggiornamento. Queste date rappresentano la prima data in cui potrebbe essere richiesto un aggiornamento a una versione più recente. Se Amazon estende il supporto per una versione RDS per Oracle più a lungo di quanto inizialmente previsto, questa tabella verrà aggiornata in base alla nuova data.

Versione principale di Oracle Database	Data prevista per l'aggiornamento a una versione più recent
Oracle Database 19c	30 aprile 2026 con BYOL Premier Support (senza commissioni per Extended Support) 30 aprile 2027 con BYOL Extended Support (costo aggiuntivo) o un contratto di licenza illimitato 30 aprile 2027 con licenza inclusa (LI)
Oracle Database 21c	30 aprile 2025 (non disponibile per Extended Support)

Prima di chiederti di eseguire l'aggiornamento a una nuova versione principale, inviamo un promemoria con almeno 12 mesi di anticipo, dove descriviamo in dettaglio il processo di aggiornamento, inclusi la tempistica di alcune fasi cardine importanti, l'impatto sulle istanze database e le azioni consigliate. Esegui accuratamente i test delle applicazioni con le nuove versioni di RDS per Oracle prima di eseguire un aggiornamento della versione principale.

Dopo questo periodo di notifica preventiva, un aggiornamento automatico alla versione principale successiva potrebbe essere applicato a qualsiasi istanza database RDS per Oracle che esegue ancora la versione precedente. In tal caso, l'aggiornamento viene avviato durante le finestre di manutenzione pianificata.

Per ulteriori informazioni, vedere Pianificazione dei rilasci delle versioni correnti del database in My Oracle Support.

Gestione della versione del motore Oracle.

Con la gestione della versione del motore del database è possibile controllare quando e come applicare una patch o un aggiornamento al motore database. Grazie a questa funzionalità, si ottiene la flessibilità necessaria per mantenere la compatibilità con le versioni delle patch del motore database. Puoi inoltre testare nuove versioni delle patch di RDS per Oracle per assicurarti che funzionino con l'applicazione prima di implementarle in produzione. Inoltre, si aggiornano le versioni secondo le proprie condizioni e timeline.

> **Note**
>
> Amazon RDS aggrega periodicamente le patch ufficiali del database Oracle mediante una versione del motore del database specifica per Amazon RDS. Per visualizzare un elenco delle patch di Oracle contenute in una versione del motore Amazon RDS specifica di Oracle, visita le Note di rilascio di Amazon RDS for Oracle.

Snapshot automatici durante gli aggiornamenti del motore

Quando aggiorni un'istanza database Oracle, gli snapshot offrono protezione contro i problemi di aggiornamento. Se il periodo di retention dei backup per l'istanza database è maggiore di 0, durante l'aggiornamento Amazon RDS esegue i seguenti snapshot DB:

1. Uno snapshot DB relativo all'istanza database prima delle modifiche legate all'aggiornamento. Se l'aggiornamento non riesce, potrai ripristinare questa snapshot e creare un'istanza database che esegue la versione precedente.
2. Una copia snapshot dell'istanza database dopo il completamento dell'aggiornamento.

> **Note**
>
> Per cambiare il periodo di retention dei backup, consulta Modifica di un'istanza database Amazon RDS.

Dopo un aggiornamento, non è possibile ripristinare la versione precedente del motore. Tuttavia, è possibile creare una nuova istanza database Oracle ripristinando lo snapshot pre-aggiornamento.

Aggiornamenti Oracle in una implementazione multi-AZ

Se la tua istanza database è in un'implementazione Multi-AZ, Amazon RDS aggiorna sia le repliche principali sia le repliche standby. Se non sono necessari aggiornamenti del sistema operativo, gli aggiornamenti primari e standby vengono eseguiti contemporaneamente. Le istanze non sono disponibili fino al completamento dell'aggiornamento.

Se sono necessari aggiornamenti del sistema operativo in una distribuzione Multi-AZ, Amazon RDS applica gli aggiornamenti quando richiedi l'aggiornamento del database. Amazon RDS esegue le seguenti operazioni:

1. Aggiorna il sistema operativo sull'istanza DB in standby corrente.
2. Trasforma l'istanza DB principale nell'istanza DB in standby.
3. Aggiorna la versione del database sulla nuova istanza DB primaria, che in precedenza era l'istanza di standby. Il database primario non è disponibile durante l'aggiornamento.
4. Aggiorna il sistema operativo sulla nuova istanza DB in standby, che in precedenza era l'istanza DB principale.
5. Aggiorna la versione del database sulla nuova istanza DB in standby.
6. Effettua il failover della nuova istanza DB primaria sull'istanza DB primaria originale e la nuova istanza DB in standby sull'istanza DB in standby originale. Pertanto, Amazon RDS riporta la configurazione di replica allo stato originale.

Aggiornamenti Oracle delle repliche di lettura

La versione del motore di Oracle DB dell'istanza database di origine e tutte le relative repliche di lettura devono essere uguali. Amazon RDS esegue l'aggiornamento nelle seguenti fasi:

1. Aggiorna l'istanza database di origine. Le repliche di lettura sono disponibili in questa fase.
2. Aggiorna le repliche di lettura in parallelo, indipendentemente dalle finestre di manutenzione della replica. Il DB di origine è disponibile in questa fase.

Per gli aggiornamenti delle versioni principali delle repliche di lettura tra regioni, Amazon RDS esegue operazioni aggiuntive:

- Genera automaticamente un gruppo di opzioni per la versione di destinazione
- Copia tutte le opzioni e le impostazioni delle opzioni dal gruppo di opzioni originale al nuovo gruppo di opzioni
- Associa la replica aggiornata di lettura tra regioni al nuovo gruppo di opzioni

Aggiornamenti Oracle delle istanze di Micro DB

Non è consigliabile aggiornare i database in esecuzione su istanze micro DB. Poiché queste istanze hanno una CPU limitata, il completamento dell'aggiornamento può richiedere ore.

Puoi aggiornare istanze database micro con piccole quantità di storage (da 10 a 20 GiB) copiando i dati mediante Data Pump. Prima di eseguire la migrazione delle istanze database di produzione, ti consigliamo di eseguire il test copiando i dati mediante Data Pump.

Aggiornamenti a una versione principale Oracle

Per eseguire un aggiornamento a una versione principale, modificare l'istanza database manualmente. Gli aggiornamenti a una versione principale non si verificano in modo automatico.

> **Important**
>
> Accertati di testare in modo approfondito qualsiasi aggiornamento per verificare che le tue applicazioni funzionino correttamente prima di applicare l'aggiornamento ai database di produzione. Per ulteriori informazioni, consulta Verifica di un aggiornamento del database Oracle.

Argomenti

- Versioni supportate per gli aggiornamenti principali
- Classi di istanza supportate per gli aggiornamenti principali
- Raccolta delle statistiche prima degli aggiornamenti principali
- Consentire gli aggiornamenti principali

Versioni supportate per gli aggiornamenti principali

Amazon RDS supporta i seguenti aggiornamenti di una versione principale.

Versione corrente	Aggiornamenti supportati
19.0.0.0 utilizzando l'architettura CDB	2110,0

Un aggiornamento della versione principale di Oracle Database deve eseguire l'aggiornamento a un Release Update (RU) rilasciato nello stesso mese o successivamente. I downgrade delle versioni principali non sono supportati per le versioni di Oracle Database.

Classi di istanza supportate per gli aggiornamenti principali

In alcuni casi, l'istanza database Oracle corrente potrebbe essere in esecuzione su una classe di istanza database che non è supportata per la versione verso cui si sta eseguendo l'aggiornamento. In questo caso, prima di eseguire l'aggiornamento, esegui la migrazione dell'istanza database a una classe di istanza database supportata. Per ulteriori informazioni sulle classi di istanze database supportate per ogni versione ed edizione di Amazon RDS for Oracle, consulta Classi di istanze database.

Raccolta delle statistiche prima degli aggiornamenti principali

Prima di effettuare l'aggiornamento a una versione principale, Oracle consiglia di raccogliere statistiche di ottimizzazione sull'istanza database che si sta aggiornando. Questa azione può ridurre i tempi di inattività dell'istanza database durante l'aggiornamento.

Per raccogliere le statistiche dell'ottimizzatore, connettersi all'istanza database come utente principale ed eseguire la procedura `DBMS_STATS.GATHER_DICTIONARY_STATS`, come nel seguente esempio.

```
EXEC DBMS_STATS.GATHER_DICTIONARY_STATS;
```

Per ulteriori informazioni, consulta Raccogliere le statistiche dell'ottimizzatore per ridurre il tempo di inattività del database Oracle nella documentazione di Oracle.

Consentire gli aggiornamenti principali

Un aggiornamento della versione principale del motore potrebbe non essere compatibile con l'applicazione in uso. L'aggiornamento è irreversibile. Se si specifica una versione principale per il parametro EngineVersion diversa dalla versione principale corrente, è necessario consentire gli aggiornamenti delle versioni principali.

Se aggiorni una versione principale utilizzando il comando della CLI modify-db-instance, dovrai specificare `--allow-major-version-upgrade`. Questa impostazione non è definitiva, pertanto dovrai specificare `--allow-major-version-upgrade` ogni volta che esegui un aggiornamento importante. Questo parametro non ha alcun impatto sugli aggiornamenti delle versioni secondarie del motore. Per ulteriori informazioni, consulta Aggiornamento della versione del motore di un'istanza database.

Se si aggiorna una versione principale utilizzando la console, non è necessario scegliere un'opzione per consentire l'aggiornamento. Invece, la console visualizza un avviso che indica che gli aggiornamenti principali sono irreversibili.

Aggiornamenti a una versione secondaria Oracle

Un aggiornamento della versione secondaria applica un aggiornamento del set di patch del database Oracle (PSU, Patch Set Update) o un aggiornamento della versione (RU, Release Update) a una versione principale del motore. Ad esempio, se l'istanza database esegue la versione principale Oracle Database 21c e la versione secondaria 21.0.0.0.ru-2022-07.rur-2022-07.r1, puoi eseguire l'aggiornamento alla versione secondaria 21.0.0.0.ru-2022- 10.rur-2022-10.r1. In genere, una nuova versione secondaria è disponibile ogni trimestre.

> **Note**
>
> RDS per Oracle non supporta i downgrade a versioni precedenti.

Puoi eseguire l'aggiornamento del motore di database a una versione secondaria manualmente o automaticamente. Per informazioni su come eseguire l'aggiornamento manualmente, consulta Aggiornamento manuale della versione del motore. Per informazioni su come configurare gli aggiornamenti automatici, consulta Aggiornamento automatico della versione secondaria del motore. Indipendentemente dal fatto che sia manuale o automatico, un aggiornamento della versione secondaria comporta tempi di inattività. Tienilo in considerazione quando pianifichi gli aggiornamenti.

> **Important**
>
> Accertati di testare in modo approfondito qualsiasi aggiornamento per verificare che le tue applicazioni funzionino correttamente prima di applicare l'aggiornamento ai database di produzione. Per ulteriori informazioni, consulta Verifica di un aggiornamento del database Oracle.

Argomenti

Attivazione degli aggiornamenti a versioni secondarie automatiche per Oracle

In un aggiornamento automatico della versione secondaria, RDS applica l'ultima versione secondaria disponibile al database Oracle senza intervento manuale. Un'istanza database Amazon RDS per Oracle pianifica l'aggiornamento durante la successiva finestra di manutenzione nelle seguenti circostanze:

- L'opzione Aggiornamento automatico della versione secondaria è attivata per l'istanza database.
- L'istanza database non esegue già la versione secondaria più recente del motore.
- L'istanza database non ha già un aggiornamento in attesa pianificato.

Per informazioni su come attivare gli aggiornamenti automatici, consulta Aggiornamento automatico della versione secondaria del motore.

Prima di pianificare un aggiornamento automatico della versione secondaria per Oracle

RDS pubblica un preavviso prima di iniziare a pianificare gli aggiornamenti automatici. Puoi trovare la notifica nella scheda Manutenzione e backup della pagina dei dettagli del database. Il messaggio ha il formato seguente:

```
An automatic minor version upgrade to engine version will become available
 on availability-date and will be applied during a subsequent maintenance window.
```

La *availability-date* nel messaggio precedente è la data in cui RDS inizia a pianificare gli aggiornamenti per le istanze database della tua Regione AWS. Non è la data in cui è pianificata l'esecuzione dell'aggiornamento dell'istanza database.

È inoltre possibile ottenere la data di disponibilità dell'aggiornamento utilizzando il comando `describe-pending-maintenance-actions` nella AWS CLI, come mostrato nell'esempio seguente:

```
aws rds describe-pending-maintenance-actions

{
    "PendingMaintenanceActions": [
        {
            "ResourceIdentifier": "arn:aws:rds:us-east-1:123456789012:db:orclinst1",
            "PendingMaintenanceActionDetails": [
                {
                    "Action": "db-upgrade",
                    "Description": "Automatic minor version upgrade to
 21.0.0.0.ru-2022-10.rur-2022-10.r1",
                    "CurrentApplyDate": "2022-12-02T08:10:00Z",
                    "OptInStatus": "next-maintenance"
                }
            ]
        }, ...
```

La tabella seguente descrive le opzioni per ogni tipo di messaggio di operazione di manutenzione in sospeso.

Messaggio di operazione di manutenzione in sospeso	Quando viene visualizzato il messaggio	Idoneo per l'applicazione nella prossima finestra di manutenzione?	Idoneo per l'applicazione immediata?	Idoneo all'annullamento dell'adesione?
Un aggiornamento automatico della versione secondaria a *engine-version* sarà disponibile il *availability-date* e dovrebbe essere applicato durante una finestra di manutenzione successiva.	4-6 settimane prima della pianificazione degli aggiornamenti automatici.	Sì	Sì	Sì
Aggiornamento automatico della versione secondaria a *engine-version*	Il *availability-date* . o successivamente.	Sì	Sì	No

Messaggio di operazione e di manutenzione in sospeso	Quando viene visualizzato il messaggio	Idoneo per l'applicazione nella prossima finestra di manutenzione?	Idoneo per l'applicazione immediata?	Idoneo all'annullamento dell'adesione?
	RDS applica automaticamente questo aggiornamento nella finestra di manutenzione successiva dell'istanza database.			

Per ulteriori informazioni su describe-pending-maintenance-actions, consulta Riferimento ai comandi AWS CLI.

Quando RDS pianifica gli aggiornamenti automatici delle versioni secondarie per Oracle

Quando arriva la data di disponibilità degli aggiornamenti automatici, RDS inizia a pianificare gli aggiornamenti. Per la maggior parte delle Regioni AWS, RDS pianifica l'aggiornamento all'ultimo RU trimestrale, circa da quattro a sei settimane dopo la data di disponibilità. La data pianificata varia in base alla Regione AWS e ad altri fattori. Per ulteriori informazioni su RU e RUR, consulta le Note di rilascio di Amazon RDS per Oracle.

Quando RDS pianifica l'aggiornamento, viene visualizzata la seguente notifica nella scheda Manutenzione e backup della pagina dei dettagli del database:

```
Automatic minor version upgrade to engine-version
```

Il messaggio precedente indica che RDS ha pianificato l'aggiornamento del motore di database nella successiva finestra di manutenzione.

Gestione di un aggiornamento automatico della versione secondaria per Oracle

Quando diventa disponibile una nuova versione secondaria, puoi eseguire manualmente l'aggiornamento dell'istanza database a questa versione. L'esempio seguente aggiorna immediatamente l'istanza database denominata `orclinst1`:

```
aws rds apply-pending-maintenance-action \
    --resource-identifier arn:aws:rds:us-east-1:123456789012:db:orclinst1 \
    --apply-action db-upgrade \
    --opt-in-type immediate
```

Per annullare l'aggiornamento automatico di una versione secondaria che non è stato ancora pianificato, imposta `opt-in-type` su `undo-opt-in` come nell'esempio seguente:

```
aws rds apply-pending-maintenance-action \
    --resource-identifier arn:aws:rds:us-east-1:123456789012:db:orclinst1 \
    --apply-action db-upgrade \
    --opt-in-type undo-opt-in
```

Se RDS ha già pianificato un aggiornamento per l'istanza database, non puoi utilizzare `apply-pending-maintenance-action` per annullarlo. È tuttavia possibile modificare l'istanza database e disattivare la funzionalità di aggiornamento automatico della versione secondaria, che quindi annulla la pianificazione dell'aggiornamento.

Per informazioni su come disattivare gli aggiornamenti automatici della versione secondaria, consulta Aggiornamento automatico della versione secondaria del motore. Per ulteriori informazioni su apply-pending-maintenance-action, consulta Riferimento ai comandi AWS CLI.

Considerazioni sugli aggiornamenti di Oracle DB

Prima di aggiornare l'istanza Oracle, esamina le informazioni riportate di seguito.

Argomenti

- Considerazioni su Oracle Multitenant
- Considerazioni su gruppi di opzioni
- Considerazioni sui gruppi di parametri
- Considerazioni sul fuso orario

Considerazioni su Oracle Multitenant

La tabella seguente illustra le architetture supportate in diverse versioni.

Versione di Oracle Database	Stato del supporto RDS	Architettura
Oracle Database 21c	Supportato	Solo CDB
Oracle Database 19c	Supportato	CDB o non CDB
Oracle Database 12c Release 2 (12.2)	Non più supportato	Solo non CDB
Oracle Database 12c Release 1 (12.1)	Non più supportato	Solo non CDB

Nella tabella seguente vengono descritti i percorsi di aggiornamento supportati e non supportati.

Percorsi di aggiornamento	Supportata?
Da non CDB a non CDB	Sì
Da CDB a CDB	Sì
Da non CDB a CDB	No
Da CDB a non CDB	No

Per ulteriori informazioni su Oracle Multitenant in RDS per Oracle, consulta Configurazione a tenant singolo dell'architettura CDB.

Considerazioni su gruppi di opzioni

Se l'istanza DB utilizza un gruppo di opzioni personalizzato, a volte Amazon RDS non è in grado di assegnare automaticamente un nuovo gruppo di opzioni. Ad esempio, questo si verifica quando effettui l'aggiornamento a una nuova versione principale. In questi casi, quando esegui l'aggiornamento specifica un nuovo gruppo di opzioni. Ti consigliamo di creare un nuovo gruppo di opzioni e di aggiungere le stesse opzioni presenti nel gruppo di opzioni personalizzato esistente.

Per ulteriori informazioni, consulta Creazione di un gruppo di opzioni o Copia di un gruppo di opzioni.

Se l'istanza database utilizza un gruppo di opzioni personalizzato che contiene l'opzione APEX, a volte è possibile ridurre il tempo di aggiornamento. Per fare ciò, aggiornare la versione di APEX contemporaneamente all'istanza DB. Per ulteriori informazioni, consulta Aggiornamento della versione di APEX.

Considerazioni sui gruppi di parametri

Se l'istanza database utilizza un gruppo di parametri personalizzato, in alcuni casi Amazon RDS non può assegnare automaticamente all'istanza DB un nuovo gruppo di parametri. Ad esempio, questo si verifica quando effettui l'aggiornamento a una nuova versione principale. In questi casi, assicurati di specificare un nuovo gruppo di parametri quando esegui l'aggiornamento. Ti consigliamo di creare un nuovo gruppo di parametri e di configurare i parametri in modo analogo al gruppo di parametri personalizzato esistente.

Per ulteriori informazioni, consulta Creazione di un gruppo di parametri del database o Copia di un gruppo di parametri database.

Considerazioni sul fuso orario

L'opzione del fuso orario consente di modificare il fuso orario del sistema utilizzato dall'istanza database Oracle. Ad esempio, potrebbe essere necessario modificare il fuso orario di un'istanza di database in modo che sia compatibile con un ambiente locale o con un'applicazione legacy. L'opzione del fuso orario modifica il fuso orario a livello di host. Amazon RDS for Oracle aggiorna automaticamente il fuso orario del sistema per tutto l'anno. Per ulteriori informazioni sulla modifica del fuso orario di sistema, consulta Fuso orario Oracle.

Quando si crea un'istanza database Oracle, il database imposta automaticamente il fuso orario del database. Il fuso orario del database è noto anche come fuso orario ora legale (DST). Il fuso orario del database è distinto dal fuso orario del sistema.

Tra le release di Oracle Database, i set di patch o le singole patch possono includere nuove versioni dell'ora legale (DST). Queste patch riflettono le modifiche apportate alle regole di transizione per varie regioni di fuso orario. Ad esempio, un governo potrebbe cambiare il periodo in cui è in vigore l'ora legale. Le modifiche alle regole dell'ora legale possono influire sui dati di tipo di `TIMESTAMP WITH TIME ZONE` esistenti.

Se si aggiorna un'istanza database RDS for Oracle, Amazon RDS non aggiorna automaticamente il file del fuso orario del database. Per aggiornare automaticamente il file del fuso orario, è possibile

includere l'opzione `TIMEZONE_FILE_AUTOUPGRADE` nel gruppo di opzioni associato all'istanza DB durante o dopo l'aggiornamento della versione del motore. Per ulteriori informazioni, consulta Aggiornamento automatico dei file di fuso orario Oracle.

In alternativa, per aggiornare manualmente il fuso orario del database, creare una nuova istanza database Oracle con la patch dell'ora legale (DST) desiderata. Tuttavia, si consiglia di aggiornare il file del fuso orario del database utilizzando l'opzione `TIMEZONE_FILE_AUTOUPGRADE`.

Dopo aver aggiornato il file del fuso orario, migra i dati dall'istanza corrente alla nuova istanza. È possibile eseguire la migrazione dei dati utilizzando diverse tecniche, tra cui le seguenti:

- AWS Database Migration Service
- Oracle GoldenGate
- Oracle Data Pump
- Esportazione/Importazione originale (non supportata per uso generale)

> **Note**
>
> Quando si esegue la migrazione dei dati utilizzando Oracle Data Pump, l'utilità genera l'errore ORA-39405 quando la versione del fuso orario di destinazione è inferiore alla versione del fuso orario di origine.

Per ulteriori informazioni, consulta l'argomento relativo alle limitazioni TIMESTAMP con TimeZONE (TIMESTAMP WITH TIMEZONE Restrictions) nella documentazione Oracle.

Verifica di un aggiornamento del database Oracle

Prima di eseguire l'aggiornamento di una versione principale nell'istanza database, verifica a fondo il database e tutte le applicazioni che accedono a esso per verificarne la compatibilità con la nuova versione. È consigliabile utilizzare la procedura seguente.

Per testare un aggiornamento di una versione principale

1. Analizzare la documentazione dell'aggiornamento Oracle per la nuova versione del motore di database per verificare se sussistono problemi di compatibilità relativi al database o alle applicazioni. Per ulteriori informazioni, consultare la Guida all'aggiornamento del database nella documentazione di Oracle.

2. Se l'istanza database utilizza un gruppo di opzioni personalizzato, creare un nuovo gruppo di opzioni compatibile con la nuova versione a cui si sta eseguendo l'aggiornamento. Per ulteriori informazioni, consulta Considerazioni su gruppi di opzioni.
3. Se l'istanza database utilizza un gruppo di parametri personalizzato, creare un nuovo gruppo di parametri compatibile con la nuova versione a cui si sta eseguendo l'aggiornamento. Per ulteriori informazioni, consulta Considerazioni sui gruppi di parametri.
4. Creare uno snapshot DB dell'istanza database da aggiornare. Per ulteriori informazioni, consulta Creazione di uno snapshot DB per un'istanza DB Single-AZ.
5. Ripristinare lo snapshot DB per creare una nuova istanza database di test. Per ulteriori informazioni, consulta Ripristino da uno snapshot database.
6. Modificare la nuova istanza database di test per aggiornarla alla nuova versione, utilizzando uno dei metodi seguenti:

   - Console
   - AWS CLI
   - API RDS
7. Eseguire i test:

   - Eseguire quanti più test di controllo qualità possibili per l'istanza database aggiornata come necessario per assicurare che il database e l'applicazione funzionino correttamente con la nuova versione.
   - Implementare qualsiasi nuovo test necessario per valutare l'impatto di problemi di compatibilità identificati nella fase 1.
   - Testare tutte le stored procedure, le funzioni e i trigger.
   - Indirizzare le versioni di test delle applicazioni all'istanza database aggiornata. Verificare che le applicazioni funzionino correttamente con la nuova versione.
   - Valutare lo storage utilizzato dall'istanza aggiornata per determinare se l'aggiornamento richiede storage aggiuntivo. Potrebbe essere necessario scegliere una classe di istanza più grande per supportare la nuova versione in produzione. Per ulteriori informazioni, consulta Classi di istanze database.
8. Se tutti i test passano, aggiornare l'istanza DB di produzione. Si consiglia di confermare che l'istanza DB funziona correttamente prima di consentire operazioni di scrittura per l'istanza DB.

Aggiornamento della versione di un'istanza DB RDS for Oracle

Per aggiornare manualmente la versione del motore DB di un'istanza DB RDS for Oracle, utilizza l' AWS Management Console, the o l'API AWS CLI RDS. Per informazioni generali sugli aggiornamenti del database in RDS, vedere. Aggiornamento della versione di un'istanza DB RDS for Oracle Per ottenere obiettivi di aggiornamento validi, utilizzare il AWS CLI describe-db-engine-versionscomando.

Console

Per aggiornare la versione del motore di un'istanza DB RDS for Oracle utilizzando la console

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, scegliere Databases (Database) quindi selezionare l'istanza database da aggiornare.
3. Scegliere Modify (Modifica).
4. Per la versione del motore DB, scegli una versione del database superiore.
5. Scegliere Continue (Continua) e controllare il riepilogo delle modifiche. Assicurati di comprendere le implicazioni di un aggiornamento della versione del database. Non è possibile riconvertire un'istanza DB aggiornata alla versione precedente. Assicurati di aver testato sia il database che l'applicazione con la nuova versione prima di continuare.
6. Decidi quando pianificare l'aggiornamento dell'istanza DB. Per applicare immediatamente le modifiche, scegliere Apply immediately (Applica immediatamente). In alcuni casi, la chiusura di questa opzione può causare un'interruzione. Per ulteriori informazioni, consulta Impostazione delle modifiche alla pianificazione.
7. Nella pagina di conferma esaminare le modifiche. Se sono corrette, seleziona Modifica istanza database per salvare le modifiche.

   In alternativa, scegliere Back (Indietro) per cambiare le modifiche o Cancel (Annulla) per annullare le modifiche.

AWS CLI

Per aggiornare la versione del motore di un'istanza DB RDS for Oracle, puoi utilizzare il comando modify-db-instanceCLI. Specifica i seguenti parametri:

- `--db-instance-identifier`— il nome dell'istanza DB RDS for Oracle.

- `--engine-version` – Numero di versione del motore di database a cui effettuare l'aggiornamento.

  Per informazioni sulle versioni valide del motore, utilizzare il AWS CLI [describe-db-engine-versions](#)comando.
- `--allow-major-version-upgrade`— per aggiornare la versione del motore DB.
- `--no-apply-immediately` – Per applicare le modifiche durante la finestra di manutenzione successiva. Per applicare immediatamente le modifiche utilizzare `--apply-immediately`.

Example

L'esempio seguente aggiorna un'istanza CDB denominata `myorainst` dalla versione corrente di `19.0.0.0.ru-2024-01.rur-2024-01.r1` a version. `21.0.0.0.ru-2024-04.rur-2024-04.r1`

PerLinux, omacOS: Unix

```
aws rds modify-db-instance \
    --db-instance-identifier myorainst \
    --engine-version 21.0.0.0.ru-2024-04.rur-2024-04.r1 \
    --allow-major-version-upgrade \
    --no-apply-immediately
```

Per Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier myorainst ^
    --engine-version 21.0.0.0.ru-2024-04.rur-2024-04.r1 ^
    --allow-major-version-upgrade ^
    --no-apply-immediately
```

API RDS

Per aggiornare un'istanza RDS for Oracle DB, utilizzare l'azione ModifyDBInstance. Specifica i seguenti parametri:

- `DBInstanceIdentifier` – Nome dell'istanza database, ad esempio *`myorainst`*.

- `EngineVersion` – Numero di versione del motore di database a cui effettuare l'aggiornamento. Per informazioni sulle versioni valide del motore, utilizzare l'operazione DescribeDB. EngineVersions
- `AllowMajorVersionUpgrade` – Se consentire un aggiornamento della versione principale. A questo scopo, imposta il valore su `true`.
- `ApplyImmediately` – Indica se applicare le modifiche immediatamente o durante la finestra di manutenzione successiva. Per applicare le modifiche immediatamente, imposta il valore su `true`. Per applicare le modifiche durante la finestra di manutenzione successiva imposta il valore su `false`.

Aggiornamento di uno shapshot DB Oracle

Se si dispone di snapshot DB manuali esistenti, è possibile aggiornarli a una versione successiva del modulo di gestione di database Oracle.

Quando Oracle smette di fornire patch per una versione e, quindi, Amazon RDS dichiara obsoleta la versione, è possibile aggiornare gli snapshot corrispondenti alla versione considerata obsoleta. Per ulteriori informazioni, consulta Gestione della versione del motore Oracle..

Amazon RDS supporta l'aggiornamento degli snapshot in tutte le regioni AWS.

Console

Per aggiornare uno shapshot DB Oracle

1. Accedi alla AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione scegliere Snapshots (Snapshot) e selezionare lo snapshot DB da aggiornare.
3. Per Actions (Operazioni), scegliere Upgrade snapshot (Aggiorna snapshot). Viene visualizzata la pagina Upgrade snapshot (Aggiorna snapshot).
4. Scegliere la nuova versione del motore a cui aggiornare la copia istantanea.
5. (Facoltativo) Per Option group (Gruppo di opzioni), selezionare il gruppo di opzioni per lo snapshot DB aggiornato. Le stesse considerazioni sui gruppi di opzioni per quando si aggiorna un'istanza database si applicano quando si aggiorna uno snapshot DB. Per ulteriori informazioni, consulta Considerazioni su gruppi di opzioni.
6. Per salvare le modifiche, scegliere Salva modifiche.

Durante il processo di aggiornamento, tutte le operazioni dello snapshot sono disabilitate per lo snapshot database. Inoltre, lo stato dello snapshot DB cambia da available (disponibile) a upgrading (in aggiornamento), quindi diventa active (attivo) al completamento. Se lo snapshot DB non può essere aggiornato a causa di problemi di corruzione, lo stato diventa unavailable (non disponibile). Non è possibile recuperare lo snapshot quando è in questo stato.

> **Note**
>
> Se l'aggiornamento dello snapshot fallisce, lo snapshot viene riportato allo stato originario con la versione iniziale.

AWS CLI

Per aggiornare uno snapshot di Oracle DB utilizzando ilAWS CLI, chiamare il modify-db-snapshotcomando con i seguenti parametri:

- `--db-snapshot-identifier` – Nome dello snapshot DB.
- `--engine-version` – Versione a cui aggiornare lo snapshot.

Puoi anche includere il seguente parametro. Le stesse considerazioni sui gruppi di opzioni per quando si aggiorna un'istanza database si applicano quando si aggiorna uno snapshot DB. Per ulteriori informazioni, consulta Considerazioni su gruppi di opzioni.

- `--option-group-name` – Gruppo di opzioni per lo snapshot DB aggiornato.

Example

Il seguente esempio consente di aggiornare uno snapshot DB.

Per LinuxmacOS, oUnix:

```
aws rds modify-db-snapshot \
    --db-snapshot-identifier mydbsnapshot \
    --engine-version 19.0.0.0.ru-2020-10.rur-2020-10.r1 \
    --option-group-name default:oracle-se2-19
```

Per Windows:

```
aws rds modify-db-snapshot ^
    --db-snapshot-identifier mydbsnapshot ^
    --engine-version 19.0.0.0.ru-2020-10.rur-2020-10.r1 ^
    --option-group-name default:oracle-se2-19
```

API RDS

Per aggiornare uno snapshot DB Oracle tramite l'API Amazon RDS, chiamare l'operazione ModifyDBSnapshot con i parametri seguenti:

- `DBSnapshotIdentifier` – Nome dello snapshot DB.
- `EngineVersion` – Versione a cui aggiornare lo snapshot.

Potrebbe anche essere necessario includere il parametro `OptionGroupName`. Le stesse considerazioni sui gruppi di opzioni per quando si aggiorna un'istanza database si applicano quando si aggiorna uno snapshot DB. Per ulteriori informazioni, consulta Considerazioni su gruppi di opzioni.

Utilizzo di software di terze parti con l'istanza database RDS for Oracle

È possibile ospitare un'istanza DB RDS for Oracle che supporti strumenti e software di terze parti.

Argomenti

Utilizzo di Oracle GoldenGate con Amazon RDS per Oracle

Oracle GoldenGate raccoglie, replica e gestisce i dati transazionali tra database. È un pacchetto software con funzionalità Change Data Capture (CDC) e di replica basate su log che viene utilizzato con i database per i sistemi elaborazione di transazioni online (OLTP). Oracle GoldenGate crea file trail che contengono i dati modificati più recentemente dal database di origine. Quindi invia questi file al server, dove un processo converte il file trail in SQL standard da applicare al database di destinazione.

Oracle GoldenGate con RDS per Oracle supporta le seguenti funzionalità:

- Replica di database attivo-attivo
- Disaster recovery
- Protezione dei dati
- Replica in una regione e tra varie regioni
- Migrazione e upgrade senza tempi di inattività
- Replica dei dati tra un'istanza database RDS per Oracle e un database non Oracle

> **Note**
>
> Per l'elenco dei database supportati, consulta Oracle Fusion Middleware Supported System Configurations (Configurazioni di sistema supportate per Oracle Fusion Middleware) nella documentazione di Oracle.

È possibile utilizzare Oracle GoldenGate con RDS for Oracle per eseguire l'aggiornamento alle versioni principali di Oracle Database. Ad esempio, puoi utilizzare Oracle per GoldenGate eseguire l'aggiornamento da un database locale Oracle Database 11g a Oracle Database 19c su un'istanza database Amazon RDS.

Argomenti

- Monitoraggio Orac GoldenGate
- Risoluzione dei problemi relativi a GoldenGate

Versioni e opzioni di licenza supportate per Oracle GoldenGate

È possibile utilizzare Standard Edition 2 (SE2) o Enterprise Edition (EE) di RDS per Oracle con Oracle GoldenGate versione 12c e successive. È possibile utilizzare le seguenti funzionalità Oracle: GoldenGate

- Oracle GoldenGate Remote Capture (estrazione) è supportato.
- L'acquisizione (extract) è supportata nelle istanze database RDS per Oracle che utilizzano la tradizionale architettura di database non CDB. L'acquisizione Oracle GoldenGate Remote PDB è supportata sui database container (CDB) di Oracle Database 21c.
- Oracle GoldenGate Remote Delivery (replicat) è supportato su istanze RDS per Oracle DB che utilizzano architetture non CDB o CDB. Remote Delivery supporta Integrated Replicat, Parallel Replicat, Coordinated Replicat e Classic Replicat.
- RDS per Oracle supporta le architetture Classic e Microservices di Oracle. GoldenGate
- La replica dei valori Oracle GoldenGate DDL e Sequence è supportata quando si utilizza la modalità di acquisizione integrata.

Sei responsabile della gestione delle GoldenGate licenze Oracle (BYOL) da utilizzare con Amazon RDS in generale. Regioni AWS Per ulteriori informazioni, consulta Opzioni di licenza per RDS per Oracle.

Requisiti e limitazioni per Oracle GoldenGate

Quando lavori con Oracle GoldenGate e RDS per Oracle, considera i seguenti requisiti e limitazioni:

- Sei responsabile della configurazione e della gestione di Oracle GoldenGate per l'utilizzo con RDS for Oracle.
- Sei responsabile della configurazione di una GoldenGate versione Oracle certificata con i database di origine e di destinazione. Per ulteriori informazioni, consulta Oracle Fusion Middleware Supported System Configurations (Configurazioni di sistema supportate per Oracle Fusion Middleware) nella documentazione di Oracle.
- Puoi utilizzare Oracle GoldenGate in molti AWS ambienti diversi per molti casi d'uso diversi. Se hai un problema relativo al supporto relativo a Oracle GoldenGate, contatta Oracle Support Services.

- È possibile utilizzare Oracle GoldenGate su RDS per istanze Oracle DB che utilizzano Oracle Transparent Data Encryption (TDE). Per mantenere l'integrità dei dati replicati, configura la crittografia sull' GoldenGate hub Oracle utilizzando i volumi crittografati di Amazon EBS o la crittografia dei file trail. Configura anche la crittografia per i dati inviati tra l' GoldenGate hub Oracle e le istanze del database di origine e destinazione. Le istanze database RDS for Oracle supportano la crittografia con Oracle Secure Sockets Layer o Oracle native network encryption.

Architettura Oracle GoldenGate

L' GoldenGate architettura Oracle da utilizzare con Amazon RDS è costituita dai seguenti moduli disaccoppiati:

Database di origine

Il database di origine può essere un database Oracle che si trova in locale, un database Oracle su un'istanza Amazon EC2 oppure un database Oracle su un'istanza database Amazon RDS.

Hub GoldenGate Oracle

Un GoldenGate hub Oracle sposta le informazioni sulle transazioni dal database di origine al database di destinazione. L'hub può essere uno dei seguenti:

- Un'istanza Amazon EC2 con Oracle Database e Oracle installati GoldenGate
- Un'installazione Oracle on-premise

Puoi disporre di più hub Amazon EC2. Si consiglia di utilizzare due hub se si utilizza Oracle GoldenGate per la replica tra regioni.

Database di destinazione

Il database di destinazione può trovarsi in un'istanza database Amazon RDS, in un'istanza Amazon EC2 o in un percorso locale.

Le seguenti sezioni descrivono gli scenari comuni per Oracle GoldenGate su Amazon RDS.

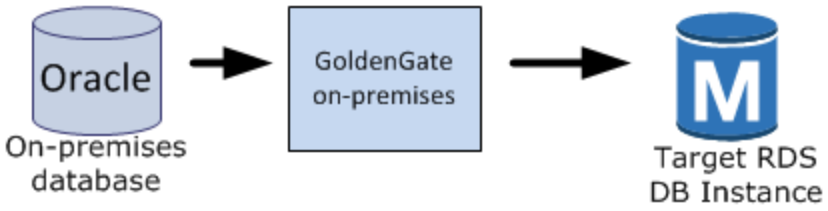
Argomenti

- Database di origine locale e hub Oracle GoldenGate
- Database di origine on-premise e hub Amazon EC2
- Database di origine Amazon RDS e hub Amazon EC2

- Database di origine Amazon EC2 e hub Amazon EC2
- Hub di Amazon EC2 in diverse regioni AWS

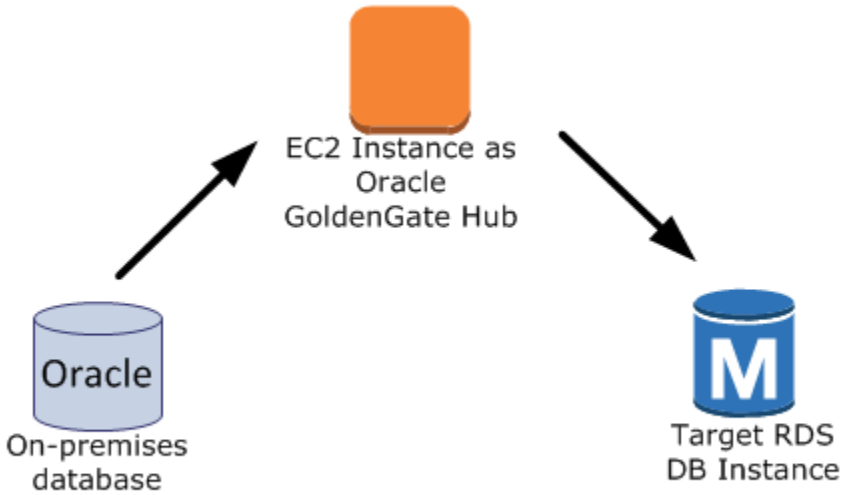
Database di origine locale e hub Oracle GoldenGate

In questo scenario, un database di origine Oracle e un GoldenGate hub Oracle locale forniscono dati a un'istanza database Amazon RDS di destinazione.



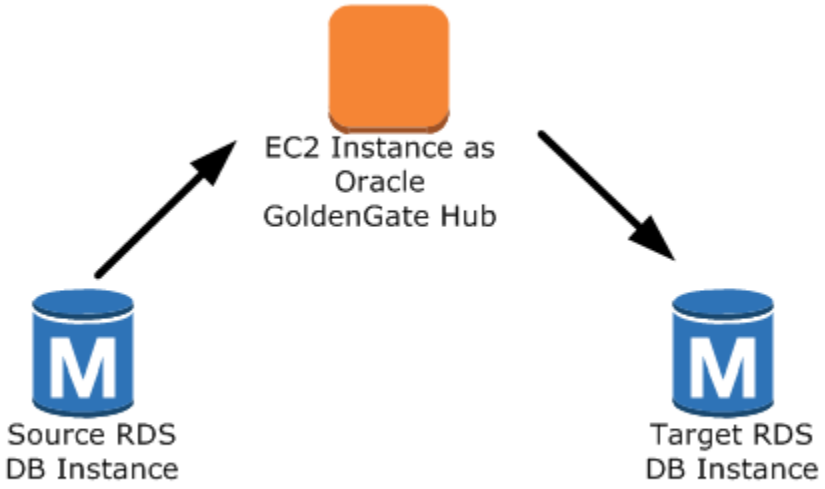
Database di origine on-premise e hub Amazon EC2

In questo scenario, un database Oracle on-premise funge da database di origine. È connesso a un hub di istanze Amazon EC2. Questo hub fornisce i dati a un'istanza database RDS di destinazione per Oracle.



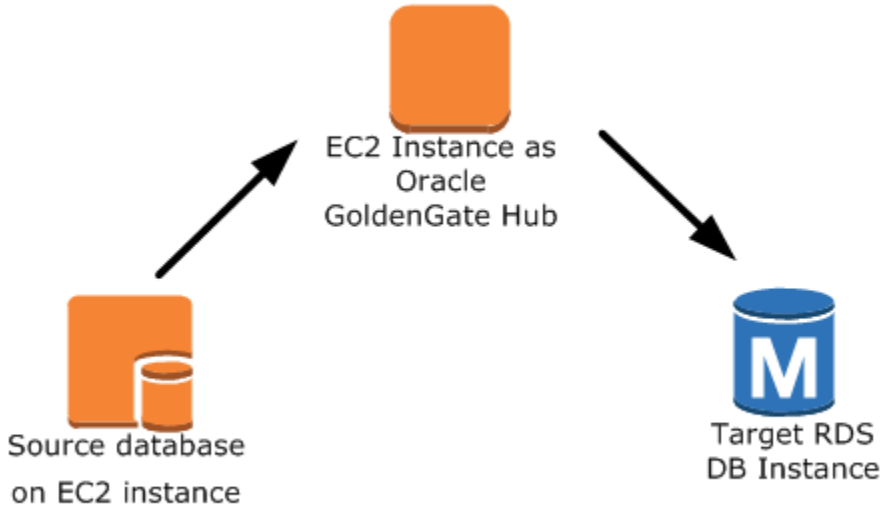
Database di origine Amazon RDS e hub Amazon EC2

In questo scenario, un'istanza database RDS for Oracle funge da database di origine. È connesso a un hub di istanze Amazon EC2. Questo hub fornisce i dati a un'istanza database RDS di destinazione per Oracle.



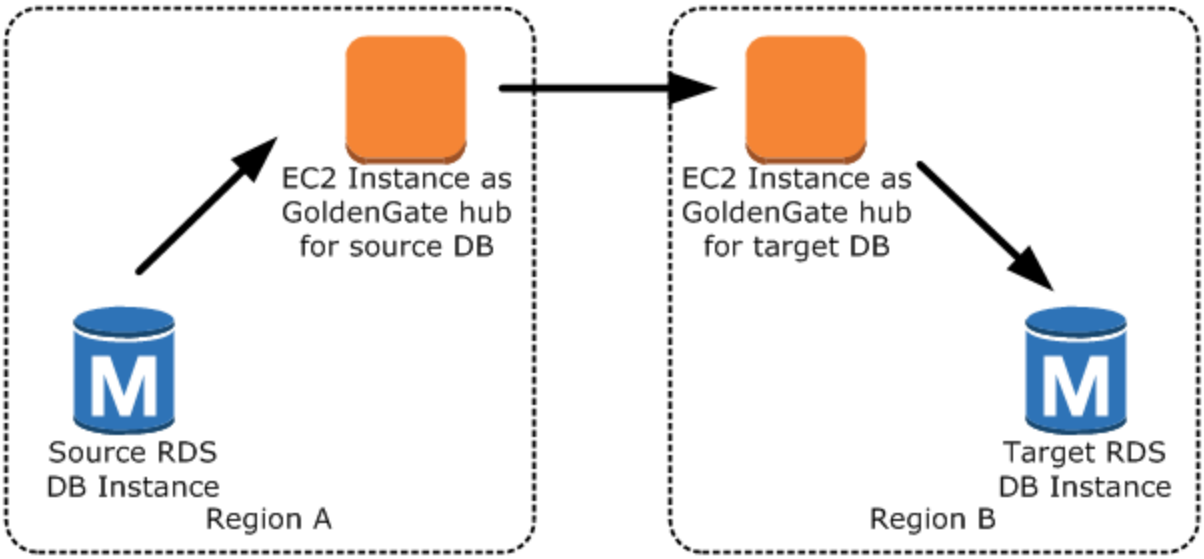
Database di origine Amazon EC2 e hub Amazon EC2

In questo scenario, un database Oracle su un'istanza Amazon EC2 funge da database di origine. È connesso a un hub di istanze Amazon EC2. Questo hub fornisce i dati a un'istanza database RDS di destinazione per Oracle.



Hub di Amazon EC2 in diverse regioni AWS

In questo scenario, un database Oracle in un'istanza database Amazon RDS è connesso a un hub di istanze Amazon EC2 nella stessa regione AWS. L'hub è connesso a un hub di istanze Amazon EC2 in una regione AWS diversa. Questo secondo hub fornisce i dati all'istanza database RDS for Oracle di destinazione nella stessa regione AWS del secondo hub di istanze Amazon EC2.



> **Note**
>
> Qualsiasi problema che influisce sull'esecuzione di Oracle GoldenGate in un ambiente locale influisce anche sull'esecuzione di Oracle GoldenGate su. AWS Si consiglia vivamente di monitorare l' GoldenGate hub Oracle per garantire che venga ripristinato `EXTRACT` e `REPLICAT` ripreso in caso di failover. Poiché l' GoldenGate hub Oracle viene eseguito su un'istanza Amazon EC2, Amazon RDS non gestisce l' GoldenGate hub Oracle e non può garantirne l'esecuzione.

Configurazione di Oracle GoldenGate

Per configurare Oracle GoldenGate utilizzando Amazon RDS, configura l'hub su un'istanza Amazon EC2, quindi configura i database di origine e di destinazione. Le sezioni seguenti forniscono un esempio di come configurare Oracle GoldenGate per l'utilizzo con Amazon RDS for Oracle.

Argomenti

- Configurazione di un GoldenGate hub Oracle su Amazon EC2
- Configurazione di un database di origine da utilizzare con Oracle GoldenGate su Amazon RDS
- Configurazione di un database di destinazione da utilizzare con Oracle GoldenGate su Amazon RDS

Configurazione di un GoldenGate hub Oracle su Amazon EC2

Per creare un GoldenGate hub Oracle su un'istanza Amazon EC2, devi prima creare un'istanza Amazon EC2 con un'installazione client completa di Oracle RDBMS. Sull'istanza Amazon EC2 deve essere installato anche il GoldenGate software Oracle. Le versioni del GoldenGate software Oracle dipendono dalle versioni del database di origine e di destinazione. Per ulteriori informazioni sull'installazione di Oracle GoldenGate, consulta la GoldenGatedocumentazione Oracle.

L'istanza Amazon EC2 che funge da GoldenGate hub Oracle archivia ed elabora le informazioni sulle transazioni dal database di origine in file trail. Per supportare questo processo, assicurati che siano soddisfatti i seguenti requisiti:

- Sia stato allocato uno spazio di archiviazione sufficiente per i file trail.
- L'istanza Amazon EC2 abbia una capacità di calcolo sufficiente per gestire la quantità di dati.
- Inoltre, assicurati che l'istanza EC2 disponga di memoria sufficiente per archiviare le informazioni sulle transazioni prima che vengano scritte nel file di trail.

Per configurare un hub di architettura GoldenGate classica Oracle su un'istanza Amazon EC2

1. Crea sottodirectory nella directory Oracle. GoldenGate

   Nella shell della riga di comando di Amazon EC2, start`ggsci`, l'interprete dei GoldenGate comandi Oracle. Il comando `CREATE SUBDIRS` crea le sottodirectory nella directory `/gg` per i file dei parametri, dei report e dei punti di controllo.

   ```
   prompt$ cd /gg
   prompt$ ./ggsci

   GGSCI> CREATE SUBDIRS
   ```

2. Configura il file `mgr.prm`.

   L'esempio seguente aggiunge le righe al file `$GGHOME/dirprm/mgr.prm`.

```
PORT 8199
PurgeOldExtracts ./dirdat/*, UseCheckpoints, MINKEEPDAYS 5
```

3. Avvia il manager.

   L'esempio seguente avvia `ggsci` ed esegue il comando `start mgr`.

```
GGSCI> start mgr
```

L' GoldenGate hub Oracle è ora pronto per l'uso.

Configurazione di un database di origine da utilizzare con Oracle GoldenGate su Amazon RDS

Quando il database di origine esegue Oracle Database 12c o versione successiva, completa le seguenti attività per configurare un database di origine da utilizzare con Oracle GoldenGate.

Passaggi di impostazione

- Passaggio 1: attivazione della registrazione supplementare nel database di origine
- Passaggio 2: impostazione del parametro di inizializzazione ENABLE_GOLDENGATE_REPLICATION su true
- Passaggio 3: impostazione del periodo di conservazione del log nel database di origine
- Fase 4: Creare un account GoldenGate utente Oracle nel database di origine
- Passaggio 5: concessione dei privilegi all'account utente per il database di origine
- Passaggio 6: aggiunta di un alias TNS per il database di origine

Passaggio 1: attivazione della registrazione supplementare nel database di origine

Per attivare la registrazione supplementare minima a livello di database, esegui la seguente procedura PL/SQL:

```
EXEC rdsadmin.rdsadmin_util.alter_supplemental_logging(p_action => 'ADD')
```

Passaggio 2: impostazione del parametro di inizializzazione ENABLE_GOLDENGATE_REPLICATION su true

Quando imposti il parametro di inizializzazione `ENABLE_GOLDENGATE_REPLICATION` su `true`, i servizi di database supportano la replica logica. Se il database di origine si trova in

un'istanza database Amazon RDS, assicurati di disporre di un gruppo di parametri assegnato all'istanza database con il parametro di inizializzazione `ENABLE_GOLDENGATE_REPLICATION` impostato su `true`. Per ulteriori informazioni sul parametro di inizializzazione `ENABLE_GOLDENGATE_REPLICATION`, consulta la documentazione di Oracle Database.

Passaggio 3: impostazione del periodo di conservazione del log nel database di origine

Assicurati di configurare il database di origine in modo che mantenga i log redo archiviati. Considera le linee guida seguenti:

- Specifica il periodo di retention dei log in ore. Il valore minimo è 1 ora.
- Imposta la durata su un valore superiore a eventuali potenziali tempi di inattività dell'istanza database di origine e a eventuali potenziali problemi relativi al periodo di comunicazione o di rete per l'istanza di origine. Tale durata consente a Oracle di GoldenGate recuperare i log dall'istanza di origine secondo necessità.
- Assicurarsi di disporre di spazio sufficiente sulla propria istanza per i file.

Ad esempio, impostare il periodo di conservazione per i redo log archiviati su 24 ore.

```
EXEC rdsadmin.rdsadmin_util.set_configuration('archivelog retention hours',24)
```

Se non è abilitata l'opzione di conservazione dei log o il suo valore è troppo basso, riceverai un messaggio di errore simile al seguente.

```
2022-03-06 06:17:27  ERROR   OGG-00446  error 2 (No such file or directory)
opening redo log /rdsdbdata/db/GGTEST3_A/onlinelog/o1_mf_2_9k4bp1n6_.log for sequence
 1306
Not able to establish initial position for begin time 2022-03-06 06:16:55.
```

Poiché l'istanza database mantiene i log redo archiviati, assicurati di disporre dello spazio sufficiente per i file. Per verificare quanto spazio hai usato nelle ultime *`num_hours`* ore, utilizza la query seguente, sostituendo *`num_hours`* con il numero di ore.

```
SELECT SUM(BLOCKS * BLOCK_SIZE) BYTES FROM V$ARCHIVED_LOG
   WHERE NEXT_TIME>=SYSDATE-num_hours/24 AND DEST_ID=1;
```

Fase 4: Creare un account GoldenGate utente Oracle nel database di origine

Oracle GoldenGate funziona come utente del database e richiede i privilegi di database appropriati per accedere ai redo log e ai redo log archiviati per il database di origine. A questo scopo, crea un account utente sul database di origine. Per ulteriori informazioni sulle autorizzazioni per un account GoldenGate utente Oracle, consulta la documentazione Oracle.

Le istruzioni seguenti creano un account utente denominato `oggadm1`.

```
CREATE TABLESPACE administrator;
CREATE USER oggadm1  IDENTIFIED BY "password"
   DEFAULT TABLESPACE ADMINISTRATOR TEMPORARY TABLESPACE TEMP;
ALTER USER oggadm1 QUOTA UNLIMITED ON administrator;
```

> **Note**
>
> Specifica una password diversa dal prompt mostrato qui come best practice per la sicurezza.

Passaggio 5: concessione dei privilegi all'account utente per il database di origine

In questa attività, concedi i privilegi dell'account necessari agli utenti per il database di origine.

Per concedere privilegi dell'account per il database di origine

1. Concedi i privilegi necessari all'account GoldenGate utente Oracle utilizzando il comando SQL `grant` e la `rdsadmin.rdsadmin_util` procedura. `grant_sys_object` Le istruzioni seguenti concedono i privilegi all'utente denominato `oggadm1`.

   ```
   GRANT CREATE SESSION, ALTER SESSION TO oggadm1;
   GRANT RESOURCE TO oggadm1;
   GRANT SELECT ANY DICTIONARY TO oggadm1;
   GRANT FLASHBACK ANY TABLE TO oggadm1;
   GRANT SELECT ANY TABLE TO oggadm1;
   GRANT SELECT_CATALOG_ROLE TO rds_master_user_name WITH ADMIN OPTION;
   EXEC rdsadmin.rdsadmin_util.grant_sys_object ('DBA_CLUSTERS', 'OGGADM1');
   GRANT EXECUTE ON DBMS_FLASHBACK TO oggadm1;
   GRANT SELECT ON SYS.V_$DATABASE TO oggadm1;
   GRANT ALTER ANY TABLE TO oggadm1;
   ```

2. Concedi i privilegi necessari a un account utente per essere amministratore Oracle GoldenGate. Il pacchetto utilizzato per eseguire la concessione `dbms_goldengate_auth` o `rdsadmin_dbms_goldengate_auth` dipende dalla versione del motore di Oracle DB.

   - Per le versioni di Oracle DB successive o uguali a Oracle Database 12c Release 2 (12.2), che richiedono il livello di patch 12.2.ru-12.2.0.1.rur-2019-04.12c o successivo, eseguire il seguente programma PL/SQL.

     ```
     EXEC rdsadmin.rdsadmin_dbms_goldengate_auth.grant_admin_privilege (
         grantee                 => 'OGGADM1',
         privilege_type          => 'capture',
         grant_select_privileges => true,
         do_grants               => TRUE);
     ```

   - Per le versioni di Oracle Database precedenti a Database 12c Release 2 (12.2), esegui il seguente programma PL/SQL.

     ```
     EXEC dbms_goldengate_auth.grant_admin_privilege (
         grantee                 => 'OGGADM1',
         privilege_type          => 'capture',
         grant_select_privileges => true,
         do_grants               => TRUE);
     ```

   Per revocare i privilegi, utilizzare la procedura `revoke_admin_privilege` nello stesso pacchetto.

Passaggio 6: aggiunta di un alias TNS per il database di origine

Aggiungi la seguente voce `$ORACLE_HOME/network/admin/tnsnames.ora` in Oracle Home che sarà utilizzata dal processo `EXTRACT`. Per ulteriori informazioni sul file `tnsnames.ora`, consulta la documentazione di Oracle.

```
OGGSOURCE=
   (DESCRIPTION=
        (ENABLE=BROKEN)
        (ADDRESS_LIST=
            (ADDRESS=(PROTOCOL=TCP)(HOST=goldengate-source.abcdef12345.us-
west-2.rds.amazonaws.com)(PORT=8200)))
        (CONNECT_DATA=(SERVICE_NAME=ORCL))
```

```
    )
```

Configurazione di un database di destinazione da utilizzare con Oracle GoldenGate su Amazon RDS

In questo task, si configura un'istanza DB di destinazione da utilizzare con Oracle GoldenGate.

Passaggi di impostazione

- Passaggio 1: impostazione del parametro di inizializzazione ENABLE_GOLDENGATE_REPLICATION su true
- Fase 2: Creare un account GoldenGate utente Oracle sul database di destinazione
- Passaggio 3: concessione dei privilegi all'account per il database di destinazione
- Passaggio 4: aggiunta di un alias TNS per il database di destinazione

Passaggio 1: impostazione del parametro di inizializzazione ENABLE_GOLDENGATE_REPLICATION su true

Quando imposti il parametro di inizializzazione `ENABLE_GOLDENGATE_REPLICATION` su `true`, i servizi di database supportano la replica logica. Se il database di origine si trova in un'istanza database Amazon RDS, assicurati di disporre di un gruppo di parametri assegnato all'istanza database con il parametro di inizializzazione `ENABLE_GOLDENGATE_REPLICATION` impostato su `true`. Per ulteriori informazioni sul parametro di inizializzazione `ENABLE_GOLDENGATE_REPLICATION`, consulta la documentazione di Oracle Database.

Fase 2: Creare un account GoldenGate utente Oracle sul database di destinazione

Oracle GoldenGate viene eseguito come utente del database e richiede i privilegi di database appropriati. A questo scopo, crea un account utente nel database di destinazione.

L'istruzione seguente crea un utente denominato `oggadm1`.

```
CREATE TABLESPSACE administrator;
CREATE USER oggadm1  IDENTIFIED BY "password"
   DEFAULT TABLESPACE administrator
   TEMPORARY TABLESPACE temp;
ALTER USER oggadm1 QUOTA UNLIMITED ON administrator;
```

> **Note**
>
> Specifica una password diversa dal prompt mostrato qui come best practice per la sicurezza.

Passaggio 3: concessione dei privilegi all'account per il database di destinazione

In questa attività, concedi i privilegi dell'account necessari agli utenti per il database di destinazione.

Per concedere i privilegi all'account per il database di destinazione

1. Concedi i privilegi necessari all'account GoldenGate utente Oracle sul database di destinazione. Nell'esempio seguente vengono concessi privilegi a `oggadm1`.

```
GRANT CREATE SESSION        TO oggadm1;
GRANT ALTER SESSION         TO oggadm1;
GRANT CREATE CLUSTER        TO oggadm1;
GRANT CREATE INDEXTYPE      TO oggadm1;
GRANT CREATE OPERATOR       TO oggadm1;
GRANT CREATE PROCEDURE      TO oggadm1;
GRANT CREATE SEQUENCE       TO oggadm1;
GRANT CREATE TABLE          TO oggadm1;
GRANT CREATE TRIGGER        TO oggadm1;
GRANT CREATE TYPE           TO oggadm1;
GRANT SELECT ANY DICTIONARY TO oggadm1;
GRANT CREATE ANY TABLE      TO oggadm1;
GRANT ALTER ANY TABLE       TO oggadm1;
GRANT LOCK ANY TABLE        TO oggadm1;
GRANT SELECT ANY TABLE      TO oggadm1;
GRANT INSERT ANY TABLE      TO oggadm1;
GRANT UPDATE ANY TABLE      TO oggadm1;
GRANT DELETE ANY TABLE      TO oggadm1;
```

2. Concedi i privilegi necessari a un account utente per essere amministratore Oracle GoldenGate. Il pacchetto utilizzato per eseguire la concessione `dbms_goldengate_auth` o `rdsadmin_dbms_goldengate_auth` dipende dalla versione del motore di Oracle DB.

   - Per le versioni di Oracle Database successive o uguali a Oracle Database 12c Release 2 (12.2), che richiedono il livello di patch 12.2.0.1.ru-2019-04.rur-2019-04.r1 o successivo, esegui il seguente programma PL/SQL.

```
EXEC rdsadmin.rdsadmin_dbms_goldengate_auth.grant_admin_privilege (
```

```
    grantee                 => 'OGGADM1',
    privilege_type          => 'apply',
    grant_select_privileges => true,
    do_grants               => TRUE);
```

- Per le versioni di Oracle Database precedenti a Oracle Database 12c Release 2 (12.2), esegui il seguente programma PL/SQL.

```
EXEC dbms_goldengate_auth.grant_admin_privilege (
    grantee                 => 'OGGADM1',
    privilege_type          => 'apply',
    grant_select_privileges => true,
    do_grants               => TRUE);
```

Per revocare i privilegi, utilizzare la procedura `revoke_admin_privilege` nello stesso pacchetto.

Passaggio 4: aggiunta di un alias TNS per il database di destinazione

Aggiungi la seguente voce `$ORACLE_HOME/network/admin/tnsnames.ora` in Oracle Home che sarà utilizzata dal processo `REPLICAT`. Per i database Oracle multitenant, assicurati che l'alias TNS punti al nome del servizio del PDB. Per ulteriori informazioni sul file `tnsnames.ora`, consulta la documentazione di Oracle.

```
OGGTARGET=
    (DESCRIPTION=
        (ENABLE=BROKEN)
        (ADDRESS_LIST=
            (ADDRESS=(PROTOCOL=TCP)(HOST=goldengate-target.abcdef12345.us-
west-2.rds.amazonaws.com)(PORT=8200)))
        (CONNECT_DATA=(SERVICE_NAME=ORCL))
    )
```

Utilizzo delle utilità EXTRACT e REPLICAT di Oracle GoldenGate

Le GoldenGate utilità Oracle `EXTRACT` e Oracle `REPLICAT` collaborano per mantenere sincronizzati i database di origine e di destinazione tramite la replica incrementale delle transazioni utilizzando file trail. Tutte le modifiche che si verificano nel database di origine vengono rilevate automaticamente`EXTRACT`, quindi formattate e trasferite in file trail sull'hub di istanze Oracle

GoldenGate on-premise o Amazon EC2. Dopo il completamento del carico iniziale, i dati vengono letti da questi file e replicati nel database di destinazione dall'utilità `REPLICAT`.

Eseguire l'utilità Oracle EXTRACT GoldenGate

L'utilità `EXTRACT` recupera, converte e restituisce dati dal database di origine nei file di trail. Di seguito è riportato il processo di base:

1. `EXTRACT` accoda i dettagli della transazione in memoria o nello spazio di archiviazione su disco temporaneo.
2. Il database di origine esegue il commit della transazione.
3. `EXTRACT` scrive i dettagli della transazione in un file trail.
4. Il file trail indirizza questi dettagli all'hub di istanze Oracle GoldenGate on-premise o Amazon EC2 e quindi al database di destinazione.

I passaggi seguenti avviano l'utilità `EXTRACT`, acquisiscono i dati da `EXAMPLE.TABLE` nel database di origine `OGGSOURCE` e creano i file trail.

Per eseguire l'utilità EXTRACT

1. Configura il file `EXTRACT` dei parametri sull' GoldenGate hub Oracle (locale o istanza Amazon EC2). L'elenco seguente mostra un esempio di file dei parametri `EXTRACT` denominato `$GGHOME/dirprm/eabc.prm`.

   ```
   EXTRACT EABC

   USERID oggadm1@OGGSOURCE, PASSWORD "my-password"
   EXTTRAIL /path/to/goldengate/dirdat/ab

   IGNOREREPLICATES
   GETAPPLOPS
   TRANLOGOPTIONS EXCLUDEUSER OGGADM1

   TABLE EXAMPLE.TABLE;
   ```

2. Nell' GoldenGate hub Oracle, accedi al database di origine e avvia l'interfaccia a riga di GoldenGate comando Oracle. `ggsci` L'esempio seguente mostra il formato per l'accesso.

   ```
   dblogin oggadm1@OGGSOURCE
   ```

3. Aggiungi i dati transazionali per attivare la registrazione supplementare per la tabella del database.

```
add trandata EXAMPLE.TABLE
```

4. Utilizzando la linea di comando `ggsci`, abilita l'utilità `EXTRACT` tramite i comandi seguenti.

```
add extract EABC tranlog, INTEGRATED tranlog, begin now
add exttrail /path/to/goldengate/dirdat/ab
   extract EABC,
   MEGABYTES 100
```

5. Registra l'utilità `EXTRACT` sul database in modo che i log archiviati non vengano eliminati. Questa attività ti consente di recuperare vecchie transazioni di cui non sia stato eseguito il commit, se necessario. Per registrare l'utilità `EXTRACT` nel database, utilizza il comando seguente.

```
register EXTRACT EABC, DATABASE
```

6. Avviare l'utilità `EXTRACT` con il comando seguente.

```
start EABC
```

Esecuzione dell'utilità Oracle GoldenGate REPLICAT

L'utilità `REPLICAT` immette le informazioni sulla transazione nei file di trail del database di destinazione.

I seguenti passaggi abilitano e avviano l'utilità `REPLICAT` in modo che possa replicare i dati acquisiti nella tabella `EXAMPLE.TABLE` del database di destinazione `OGGTARGET`.

Per eseguire l'utilità REPLICATE

1. Configura il file `REPLICAT` dei parametri sull' GoldenGate hub Oracle (locale o istanza EC2). L'elenco seguente mostra un esempio di file dei parametri `REPLICAT` denominato `$GGHOME/dirprm/rabc.prm`.

```
REPLICAT RABC

USERID oggadm1@OGGTARGET, password "my-password"
```

```
ASSUMETARGETDEFS
MAP EXAMPLE.TABLE, TARGET EXAMPLE.TABLE;
```

> **Note**
>
> Specifica una password diversa dal prompt mostrato qui come best practice per la sicurezza.

2. Accedi al database di destinazione e avvia l'interfaccia a riga di GoldenGate comando Oracle ()`ggsci`. L'esempio seguente mostra il formato per l'accesso.

```
dblogin userid oggadm1@OGGTARGET
```

3. Utilizzando la linea di comando `ggsci`, aggiungi una tabella dei punti di controllo. L'utente indicato deve essere l'account GoldenGate utente Oracle, non il proprietario dello schema della tabella di destinazione. L'esempio seguente crea una tabella dei punti di controllo denominata `gg_checkpoint`.

```
add checkpointtable oggadm1.oggchkpt
```

4. Per abilitare l'utilità `REPLICAT`, utilizza il comando seguente.

```
add replicat RABC EXTTRAIL /path/to/goldengate/dirdat/ab CHECKPOINTTABLE
 oggadm1.oggchkpt
```

5. Avvia l'utilità `REPLICAT` utilizzando il comando seguente.

```
start RABC
```

Monitoraggio Orac GoldenGate

Quando utilizzi Oracle GoldenGate per la replica, assicurati che il GoldenGate processo Oracle sia attivo e funzionante e che i database di origine e di destinazione siano sincronizzati. È possibile utilizzare i seguenti strumenti di monitoraggio:

- Amazon CloudWatch è un servizio di monitoraggio che viene utilizzato in questo schema per monitorare i log degli GoldenGate errori.

- Amazon SNS è un servizio di notifica dei messaggi utilizzato in questo modello per inviare notifiche e-mail.

Per istruzioni dettagliate, consulta Monitorare GoldenGate i log Oracle utilizzando Amazon CloudWatch.

Risoluzione dei problemi relativi a GoldenGate

Questa sezione spiega i problemi più comuni relativi all'utilizzo di Oracle GoldenGate con Amazon RDS for Oracle.

Argomenti

Errore durante l'apertura del log redo online

Assicurati di configurare i database per mantenere i log redo archiviati. Considera le linee guida seguenti:

- Specifica il periodo di retention dei log in ore. Il valore minimo è 1 ora.
- Imposta la durata su un valore superiore a eventuali potenziali tempi di inattività dell'istanza database di origine e a eventuali potenziali problemi relativi al periodo di comunicazione o di rete per l'istanza database di origine. Tale durata consente a Oracle di GoldenGate recuperare i log dall'istanza DB di origine secondo necessità.
- Assicurarsi di disporre di spazio sufficiente sulla propria istanza per i file.

Se non è abilitata l'opzione di conservazione dei log o il suo valore è troppo basso, riceverai un messaggio di errore simile al seguente.

```
2022-03-06 06:17:27  ERROR   OGG-00446  error 2 (No such file or directory)
opening redo log /rdsdbdata/db/GGTEST3_A/onlinelog/o1_mf_2_9k4bp1n6_.log for sequence
 1306
Not able to establish initial position for begin time 2022-03-06 06:16:55.
```

Oracle GoldenGate sembra essere configurato correttamente ma la replica non funziona

Per le tabelle preesistenti, è necessario specificare l'SCN da cui lavora GoldenGate Oracle.

Per risolvere questo problema

1. Accedere al database di origine e avviare l'interfaccia a riga di GoldenGate comando Oracle ()`ggsci`. L'esempio seguente mostra il formato per l'accesso.

   ```
   dblogin userid oggadm1@OGGSOURCE
   ```

2. Utilizzando la riga di comando `ggsci`, imposta l'SCN iniziale per il processo `EXTRACT`. Nell'esempio seguente, l'SCN viene impostato su 223274 per `EXTRACT`.

   ```
   ALTER EXTRACT EABC SCN 223274
   start EABC
   ```

3. Accedi al database di destinazione. L'esempio seguente mostra il formato per l'accesso.

   ```
   dblogin userid oggadm1@OGGTARGET
   ```

4. Utilizzando la riga di comando `ggsci`, imposta l'SCN iniziale per il processo `REPLICAT`. Nell'esempio seguente, l'SCN viene impostato su 223274 per `REPLICAT`.

   ```
   start RABC atcsn 223274
   ```

Integrated REPLICAT lento a causa della query su SYS."_DBA_APPLY_CDR_INFO"

Oracle GoldenGate Conflict Detection and Resolution (CDR) fornisce routine di risoluzione dei conflitti di base. Ad esempio, CDR può risolvere un conflitto univoco per un'istruzione `INSERT`.

Quando CDR risolve una collisione, è possibile inserire temporaneamente record nella tabella delle eccezioni `_DBA_APPLY_CDR_INFO`. Integrato `REPLICAT` elimina questi record in un secondo momento. In uno scenario raro, `REPLICAT` integrato può elaborare un gran numero di collisioni, ma un nuovo integrato `REPLICAT` non lo sostituisce. Invece di essere rimosse, le righe esistenti in `_DBA_APPLY_CDR_INFO` sono orfane. Tutti i nuovi processi `REPLICAT` integrati rallentano perché stanno eseguendo query su righe orfane in `_DBA_APPLY_CDR_INFO`.

Per rimuovere tutte le righe da `_DBA_APPLY_CDR_INFO`, attenersi alla Amazon RDS procedura `rdsadmin.rdsadmin_util.truncate_apply$_cdr_info`. Questa procedura viene rilasciata

nell'ambito della versione di ottobre 2020 e dell'aggiornamento delle patch. La procedura è disponibile nelle seguenti versioni del database:

- Versione 21.0.0.0.ru-2022-01.rur-2022-01.r1 e successive
- Versione 19.0.0.0.ru-2020-10.rur-2020-10.r1 e successive

Nell'esempio seguente viene troncata la tabella `_DBA_APPLY_CDR_INFO`.

```
SET SERVEROUTPUT ON SIZE 2000
EXEC rdsadmin.rdsadmin_util.truncate_apply$_cdr_info;
```

Utilizzo di Oracle Repository Creation Utility in RDS for Oracle

Puoi utilizzare Amazon RDS per ospitare un'istanza database RDS for Oracle che contiene gli schemi per supportare i componenti Oracle Fusion Middleware. Prima di poter utilizzare i componenti Fusion Middleware, crea e popola gli schemi per i componenti nel database. Puoi creare e popolare gli schemi tramite Oracle Repository Creation Utility (RCU).

Versioni supportate e opzioni di licenza per RCU

Amazon RDS supporta solo la versione 12c di Oracle Repository Creation Utility (RCU). Puoi utilizzare l'RCU nelle configurazioni seguenti:

- RCU 12c con Oracle Database 21c
- RCU 12c con Oracle Database 19c
- RCU 12c con Oracle Database 12c Release 2 (12.2)
- RCU 12c con Oracle Database 12c Release 1 (12.1) usando la versione 12.1.0.2.v4 o successive

Prima di poter utilizzare la utilità RCU, eseguire la seguente procedura:

- Ottieni una licenza per Oracle Fusion Middleware.
- Segui le linee guida Oracle per la gestione delle licenze relative al database Oracle che ospita il repository. Per ulteriori informazioni, consulta Oracle Fusion Middleware Licensing Information User Manual (Manuale utente riguardante le informazioni di licenza per Oracle Fusion Middleware nella documentazione di Oracle.

Fusion MiddleWare supporta i repository su Oracle Database Enterprise Edition e Standard Edition 2. Oracle consiglia Enterprise Edition per le installazioni di produzione che richiedono il partizionamento e le installazioni che richiedono la ricostruzione dell'indice online.

Prima di creare l'istanza database RDS for Oracle, verifica la versione del database di Oracle necessaria per supportare i componenti che vuoi distribuire. Utilizza la matrice di certificazione per trovare i requisiti per i componenti e le versioni di Fusion Middleware che desideri implementare. Per ulteriori informazioni, consulta Oracle Fusion Middleware Supported System Configurations (Configurazioni di sistema supportate per Oracle Fusion Middleware) nella documentazione di Oracle.

Amazon RDS supporta gli aggiornamenti della versione del database di Oracle secondo le necessità. Per ulteriori informazioni, consulta Aggiornamento della versione del motore di un'istanza database.

Requisiti e limitazioni per RCU

Per utilizzare RCU è necessario un Amazon VPC. L'istanza database Amazon RDS deve essere disponibile solo per i componenti Fusion Middleware e non per Internet pubblico. Pertanto, ospita l'istanza database Amazon RDS in una sottorete privata, che garantisce una maggiore sicurezza. È inoltre necessaria un'istanza database RDS for Oracle. Per ulteriori informazioni, consulta Creazione e connessione a un'istanza database Oracle.

Puoi archiviare gli schemi per qualsiasi componente Fusion Middleware nell'istanza database Amazon RDS. Gli schemi seguenti sono stati verificati per l'installazione corretta:

- Analytics (ATTIVITà)
- Servizi di audit (IAU)
- Servizi di audit Append (IAU_APPEND)
- Servizi di audit Viewer (IAU_VIEWER)
- Discussioni (DISCUSSIONI)
- Servizi dei metadati (MDS)
- Oracle Business Intelligence (BIPLATFORM)
- Oracle Platform Security Services (OPSS)
- Portale e servizi (WEBCENTER)
- Produttori Portlet (PORTLET)
- Tabella del servizio (STB)
- Infrastruttura SOA (SOAINFRA)
- Servizio di messaggistica dell'utente (UCSUMS)
- WebLogic Servizi (WLS)

Linee guida per l'utilizzo di RCU

Di seguito sono riportate alcune raccomandazioni quando si utilizza l'istanza database in questo scenario:

- Raccomandiamo di usare Multi-AZ per carichi di lavoro di produzione. Per altre informazioni sull'utilizzo di zone di disponibilità multiple, consulta Regioni, zone di disponibilità e Local Zones.
- Per una maggiore sicurezza, Oracle consiglia di utilizzare Transparent Data Encryption (TDE) per crittografare i dati in attesa. Se si dispone di una licenza Enterprise Edition che comprende

l'opzione di sicurezza avanzata, è possibile abilitare la crittografia in attesa utilizzando l'opzione TDE. Per ulteriori informazioni, consulta Oracle Transparent Data Encryption.

Amazon RDS fornisce anche un'opzione di crittografia in attesa per tutte le edizioni del database. Per ulteriori informazioni, consulta Crittografia delle risorse Amazon RDS.

- Configura il gruppo di sicurezza VPC per consentire la comunicazione tra i server dell'applicazione e l'istanza database Amazon RDS. I server dell'applicazione che ospitano i componenti Fusion Middleware possono essere su Amazon EC2 o in locale.

Esecuzione di RCU

Usa Oracle Repository Creation Utility (RCU) per creare e popolare gli schemi per supportare i componenti di Fusion Middleware. Puoi eseguire RCU in diversi modi diversi.

Argomenti

- Esecuzione di RCU usando la riga di comando in un passaggio
- Esecuzione di RCU usando la riga di comando in più passaggi
- Esecuzione di RCU in modalità interattiva

Esecuzione di RCU usando la riga di comando in un passaggio

Se non hai bisogno di modificare nessuno dei tuoi schemi prima di popolarli, puoi eseguire RCU in un unico passaggio. In caso contrario, consulta la seguente sezione per eseguire RCU in più passaggi.

Puoi eseguire RCU in modalità silenziosa usando il parametro della linea di comando `-silent`. Quando si esegue RCU in modalità silenziosa, è possibile evitare di digitare le password sulla riga di comando creando un file di testo contenente le password. Creare un file di testo con la password per `dbUser` sulla prima riga e la password per ogni componente sulle righe successive. Specifichi il nome del file della password come ultimo parametro al comando RCU.

Example

L'esempio seguente crea e popola gli schemi per il componente dell'infrastruttura SOA (e per le sue dipendenze) in un unico passaggio.

Per LinuxmacOS, oUnix:

```
export ORACLE_HOME=/u01/app/oracle/product/12.2.1.0/fmw
export JAVA_HOME=/usr/java/jdk1.8.0_65
```

```
${ORACLE_HOME}/oracle_common/bin/rcu \
-silent \
-createRepository \
-connectString ${dbhost}:${dbport}:${dbname} \
-dbUser ${dbuser} \
-dbRole Normal \
-honorOMF \
-schemaPrefix ${SCHEMA_PREFIX} \
-component MDS \
-component STB \
-component OPSS \
-component IAU \
-component IAU_APPEND \
-component IAU_VIEWER \
-component UCSUMS \
-component WLS \
-component SOAINFRA \
-f < /tmp/passwordfile.txt
```

Per ulteriori informazioni, consulta Running Repository Creation Utility from the Command Line (Esecuzione dell'utilità di creazione del repository dalla riga di comando) nella documentazione di Oracle.

Esecuzione di RCU usando la riga di comando in più passaggi

Per modificare manualmente gli script dello schema, esegui RCU in più passaggi:

1. Eseguire RCU in modalità Prepare Scripts for System Load (Preparare gli script per il caricamento del sistema) usando il parametro della riga di comando `-generateScript` per creare gli script per gli schemi.
2. Modificare manualmente ed eseguire lo script generato `script_systemLoad.sql`.
3. Eseguire RCU di nuovo in modalità Perform Product Load (Eseguire il caricamento del prodotto) usando il parametro della riga di comando `-dataLoad` per popolare gli schemi.
4. Esegui lo script di pulizia generato `script_postDataLoad.sql`.

Per eseguire RCU in modalità silenziosa, specifica il parametro della riga di comando `-silent`. Quando si esegue RCU in modalità silenziosa, è possibile evitare di digitare le password sulla riga di comando creando un file di testo contenente le password. Creare un file di testo con la password per `dbUser` sulla prima riga e la password per ogni componente sulle righe successive. Specifica il nome del file della password come ultimo parametro al comando RCU.

Example

L'esempio seguente crea degli script per lo schema per il componente dell'infrastruttura SOA e per le sue dipendenze.

Per LinuxmacOS, oUnix:

```
export ORACLE_HOME=/u01/app/oracle/product/12.2.1.0/fmw
export JAVA_HOME=/usr/java/jdk1.8.0_65
${ORACLE_HOME}/oracle_common/bin/rcu \
-silent \
-generateScript \
-connectString ${dbhost}:${dbport}:${dbname} \
-dbUser ${dbuser} \
-dbRole Normal \
-honorOMF \
[-encryptTablespace true] \
-schemaPrefix ${SCHEMA_PREFIX} \
-component MDS \
-component STB \
-component OPSS \
-component IAU \
-component IAU_APPEND \
-component IAU_VIEWER \
-component UCSUMS \
-component WLS \
-component SOAINFRA \
-scriptLocation /tmp/rcuscripts \
-f < /tmp/passwordfile.txt
```

Ora puoi modificare lo script generato, collegarti all'istanza database di Oracle ed eseguire lo script. Lo script generato è denominato `script_systemLoad.sql`. Per ulteriori informazioni sulla connessione alla tua istanza database di Oracle, consulta Fase 3: connessione del client SQL a un'istanza database Oracle..

L'esempio seguente popola gli schemi per il componente dell'infrastruttura SOA (e per le sue dipendenze).

Per LinuxmacOS, oUnix:

```
export JAVA_HOME=/usr/java/jdk1.8.0_65
${ORACLE_HOME}/oracle_common/bin/rcu \
```

```
-silent \
-dataLoad \
-connectString ${dbhost}:${dbport}:${dbname} \
-dbUser ${dbuser} \
-dbRole Normal \
-honorOMF \
-schemaPrefix ${SCHEMA_PREFIX} \
-component MDS \
-component STB \
-component OPSS \
-component IAU \
-component IAU_APPEND \
-component IAU_VIEWER \
-component UCSUMS \
-component WLS \
-component SOAINFRA \
-f < /tmp/passwordfile.txt
```

Per completare, collegati all'istanza database di Oracle ed esegui lo script di pulizia. Lo script è denominato `script_postDataLoad.sql`.

Per ulteriori informazioni, consulta Running Repository Creation Utility from the Command Line (Esecuzione dell'utilità di creazione del repository dalla riga di comando) nella documentazione di Oracle.

Esecuzione di RCU in modalità interattiva

Per utilizzare l'interfaccia utente grafica RCU, esegui RCU in modalità interattiva. Includi il parametro `-interactive` e ometti il parametro `-silent`. Per ulteriori informazioni, consulta Understanding Repository Creation Utility Screens (Comprensione delle schermate di utilità della creazione del repository) nella documentazione di Oracle.

Example

L'esempio seguente avvia RCU in modo interattivo e pre-popola le informazioni riguardanti la connessione.

Per LinuxmacOS, oUnix:

```
export ORACLE_HOME=/u01/app/oracle/product/12.2.1.0/fmw
export JAVA_HOME=/usr/java/jdk1.8.0_65
${ORACLE_HOME}/oracle_common/bin/rcu \
-interactive \
```

```
-createRepository \
-connectString ${dbhost}:${dbport}:${dbname} \
-dbUser ${dbuser} \
-dbRole Normal
```

Risoluzione dei problemi per RCU

Presta particolare attenzione ai seguenti problemi.

Argomenti

- Oracle Managed Files (OMF)
- Privilegi degli oggetti
- Enterprise Scheduler Service

Oracle Managed Files (OMF)

Amazon RDS usa file di dati OMF per semplificare la gestione dell'archiviazione. È possibile personalizzare gli attributi del tablespace, come ad esempio la dimensione e la gestione dell'estensione. Tuttavia, la specifica del nome di un file di dati quando si esegue la utilità RCU fa sì che il codice del tablespace restituisca l'errore `ORA-20900`. La RCU può essere utilizzata con OMF nei modi seguenti:

- In RCU 12.2.1.0 e versioni successive, usare il parametro della riga di comando `-honorOMF`.
- In RCU 12.1.0.3 e versioni successive, utilizzare più passaggi e modificare lo script generato. Per ulteriori informazioni, consulta Esecuzione di RCU usando la riga di comando in più passaggi.

Privilegi degli oggetti

Poiché Amazon RDS è un servizio gestito, non avrai l'accesso `SYSDBA` completo all'istanza database RDS per Oracle. Tuttavia, RCU 12c supporta utenti con privilegi inferiori. Nella maggior parte dei casi, il privilegio dell'utente master è sufficiente a creare repository.

L'account master può concedere direttamente i privilegi `WITH GRANT OPTION` già concessi. In alcuni casi, la utilità RCU potrebbe avere esito negativo con `ORA-01031` quando si cerca di concedere i privilegi oggetto `SYS`. Puoi riprovare ed eseguire la stored procedure `rdsadmin_util.grant_sys_object`, come mostrato nell'esempio seguente:

```
BEGIN
```

```
  rdsadmin.rdsadmin_util.grant_sys_object('GV_$SESSION','MY_DBA','SELECT');
END;
/
```

Se tenti di concedere i privilegi SYS sull'oggetto SCHEMA_VERSION_REGISTRY, l'operazione potrebbe avere esito negativo e restituire l'errore `ORA-20199: Error in rdsadmin_util.grant_sys_object`. Puoi qualificare la tabella SCHEMA_VERSION_REGISTRY$ e la vista SCHEMA_VERSION_REGISTRY con il nome del proprietario dello schema, ovvero SYSTEM, e riprovare l'operazione. In alternativa, puoi creare un sinonimo. Accedi come utente master ed esegui le seguenti istruzioni:

```
CREATE OR REPLACE VIEW SYSTEM.SCHEMA_VERSION_REGISTRY
  AS SELECT * FROM SYSTEM.SCHEMA_VERSION_REGISTRY$;
CREATE OR REPLACE PUBLIC SYNONYM SCHEMA_VERSION_REGISTRY FOR
 SYSTEM.SCHEMA_VERSION_REGISTRY;
CREATE OR REPLACE PUBLIC SYNONYM SCHEMA_VERSION_REGISTRY$ FOR SCHEMA_VERSION_REGISTRY;
```

Enterprise Scheduler Service

Quando usi la utilità RCU per eliminare un repository Enterprise Scheduler Service, la RCU può restituire il messaggio `Error: Component drop check failed`.

Configurazione di Oracle Connection Manager su un'istanza Amazon EC2

Oracle Connection Manager (CMAN) è un server proxy che inoltra le richieste di connessione ai server di database o ad altri server proxy. Puoi utilizzare CMAN per configurare quanto segue:

Controllo accessi

Puoi creare regole che filtrano le richieste client specificate dall'utente e accettarne altre.

Multiplexing della sessione

Puoi incanalare più sessioni client tramite una connessione di rete a una destinazione server condivisa.

In genere, CMAN risiede su un host separato dal server di database e dagli host client. Per ulteriori informazioni, consulta Configuring Oracle Connection Manager (Configurazione di Oracle Connection Manager) nella documentazione di Oracle Database.

Argomenti

- Versioni supportate e opzioni di licenza per CMAN
- Requisiti e limitazioni per CMAN
- Configurazione di CMAN

Versioni supportate e opzioni di licenza per CMAN

CMAN supporta l'Enterprise Edition di tutte le versioni di Oracle Database supportate da Amazon RDS. Per ulteriori informazioni, consultare Release di RDS per Oracle.

Puoi installare Oracle Connection Manager su un host separato dall'host in cui è installato Oracle Database. Non è necessaria una licenza separata per l'host che esegue CMAN.

Requisiti e limitazioni per CMAN

Per offrire un'esperienza completamente gestita, Amazon RDS limita l'accesso al sistema operativo. Non è possibile modificare i parametri del database che richiedono l'accesso al sistema operativo. Pertanto, Amazon RDS non supporta le caratteristiche di CMAN che richiedono l'accesso al sistema operativo.

Configurazione di CMAN

Quando si configura CMAN, si esegue la maggior parte del lavoro al di fuori del database RDS for Oracle.

Argomenti

Passaggio 1: configura CMAN in un'istanza Amazon EC2 nello stesso VPC dell'istanza RDS for Oracle

Per informazioni su come configurare CMAN, segui le istruzioni dettagliate nel post del blog Configuring and using Oracle Connection Manager on Amazon EC2 for Amazon RDS for Oracle (Configurazione e utilizzo di Oracle Connection Manager su Amazon EC2 per Amazon RDS for Oracle).

Passaggio 2: configura i parametri del database per CMAN

Per le caratteristiche CMAN come la modalità Traffic Director e il multiplexing di sessione, impostare`REMOTE_LISTENER`parametro all'indirizzo dell'istanza CMAN in un gruppo di parametri del database. Considera il seguente scenario:

- L'istanza CMAN risiede su un host con indirizzo IP `10.0.159.100` e utilizza la porta `1521`.
- I database `orcla`, `orclb` e `orclc` risiedono in istanze database RDS for Oracle separate.

La tabella seguente mostra come impostare il valore `REMOTE_LISTENER`. Il valore `LOCAL_LISTENER` viene impostato automaticamente da Amazon RDS.

Nome dell'istanza database	IP dell'istanza database	Valore del listener locale (impostato automaticamente)	Valore del listener remoto (impostato dall'utente)
`orcla`	`10.0.159.200`	`( address= (protocol=tcp)`	`10.0.159.100:1521`

Nome dell'istanza database	IP dell'istanza database	Valore del listener locale (impostato automaticamente)	Valore del listener remoto (impostato dall'utente)
		(host=10.0.159.200) (port=1521))	
orclb	10.0.159.300	(address= (protocol=tcp) (host=10.0.159.300) (port=1521))	10.0.159.100:1521
orclc	10.0.159.400	(address= (protocol=tcp) (host=10.0.159.400) (port=1521))	10.0.159.100:1521

Passaggio 3: associa l'istanza database al gruppo di parametri

Crea o modifica l'istanza database per utilizzare il gruppo di parametri configurato in Passaggio 2: configura i parametri del database per CMAN. Per ulteriori informazioni, consultare Associazione di un gruppo di parametri database a un'istanza database.

Installazione di un Database Siebel in Oracle in Amazon RDS

È possibile usare Amazon RDS per ospitare un database Siebel in un'istanza database di Oracle. Il database Siebel fa parte dell'architettura dell'applicazione Siebel Customer Relationship Management (CRM). Per un'illustrazione, consulta Architettura generica dell'applicazione Siebel Business.

Utilizzare il seguente argomento per facilitare la configurazione di un Database Siebel su un'istanza database Oracle su Amazon RDS. È inoltre possibile scoprire come utilizzare Amazon Web Services per supportare gli altri componenti richiesti dall'architettura dell'applicazione Siebel CRM.

> **Note**
>
> Per installare database Siebel in Oracle in Amazon RDS, è necessario utilizzare l'account utente master. Non è necessario il privilegio SYSDBA; è sufficiente il privilegio dell'utente master. Per ulteriori informazioni, consulta Privilegi dell'account utente master.

Licenze e versioni

Per installare un database Siebel in Amazon RDS, è necessario usare la propria licenza database di Oracle e la propria licenza Siebel. È necessario disporre della licenza Oracle Database (con Licenza di aggiornamento software e supporto) adatta alla classe dell'istanza database e all'edizione di Oracle Database. Per ulteriori informazioni, consulta Opzioni di licenza per RDS per Oracle.

Oracle Database Enterprise Edition è l'unica edizione certificata da Siebel per questo scenario. Amazon RDS supporta Siebel CRM versione 15.0 o 16.0. Utilizzare Oracle Database 12c Release 1 (12.1.0.2.0). Per le procedure seguenti, utilizziamo Siebel CRM versione 15.0 e Oracle Database Release 1 (12.1.0.2) o Oracle Database Release 2 (12.2.0.1). Per ulteriori informazioni, consulta Oracle Database 12c con Amazon RDS.

Amazon RDS supporta gli aggiornamenti della versione del database. Per ulteriori informazioni, consulta Aggiornamento della versione del motore di un'istanza database.

Prima di iniziare

Prima di iniziare è necessario un Amazon VPC. Poiché l'istanza database Amazon RDS deve essere disponibile solo per il server Siebel Enterprise e non nell'ambiente Internet pubblico, l'istanza database Amazon RDS viene ospitata in una sottorete privata, per garantire una maggiore sicurezza. Per informazioni su come creare un Amazon VPC da usare con Siebel CRM, consulta Creazione e connessione a un'istanza database Oracle.

Prima di iniziare, è necessaria anche un'istanza database di Oracle. Per informazioni su come creare un'istanza database Oracle da usare con Siebel CRM, consulta Creazione di un'istanza database Amazon RDS.

Installazione e configurazione di un Database Siebel

Dopo aver creato l'istanza database di Oracle, è possibile installare il database di Siebel. Si installa il database creando il proprietario della tabella e gli account dell'amministratore, installando procedure archiviate e funzioni e quindi eseguendo la Configurazione guidata del database Siebel. Per altri informazioni, consulta Installing the Siebel Database on the RDBMS (Installazione del database di Siebel in RDBMS).

Per eseguire la configurazione guidata del database di Siebel, è necessario utilizzare l'account utente master. Non è necessario il privilegio `SYSDBA`; è sufficiente il privilegio dell'utente master. Per ulteriori informazioni, consulta Privilegi dell'account utente master.

Utilizzo di altre caratteristiche Amazon RDS con un Database di Siebel

Dopo aver creato l'istanza database di Oracle, è possibile utilizzare delle caratteristiche aggiuntive di Amazon RDS per aiutare a personalizzare il database di Siebel.

Raccolta di statistiche con l'opzione Oracle Statspack

Puoi aggiungere delle caratteristiche all'istanza database tramite l'uso di opzioni nei gruppi di opzioni database. Quando hai creato l'istanza database di Oracle, hai utilizzato il gruppo di opzioni predefinite database. Se si desidera aggiungere caratteristiche al database, è possibile creare un nuovo gruppo di opzioni per l'istanza database.

Se si desidera raccogliere statistiche sulle prestazioni nel database Siebel, è possibile aggiungere la caratteristica Oracle Statspack Per ulteriori informazioni, consulta Oracle Statspack.

Alcune modifiche dell'opzione vengono applicate immediatamente e alcune modifiche dell'opzione vengono applicate durante la finestra di manutenzione successiva per l'istanza database. Per ulteriori informazioni, consulta Uso di gruppi di opzioni. Dopo aver creato un gruppo di opzioni personalizzate, modificare l'istanza database per il collegamento. Per ulteriori informazioni, consulta Modifica di un'istanza database Amazon RDS.

Tuning Performance con i parametri

Puoi gestire la configurazione del motore database attraverso l'uso di parametri in un gruppo di parametri database. Quando hai creato l'istanza database di Oracle, hai utilizzato il gruppo di

parametri predefiniti database. Se si desidera personalizzare le caratteristiche del database, è possibile creare un nuovo gruppo di parametri per l'istanza database.

Quando modifichi un parametro, in base al tipo di parametro, le modifiche vengono applicate o immediatamente o dopo i riavvio manuale dell'istanza database. Per ulteriori informazioni, consulta Utilizzo di gruppi di parametri. Dopo aver creato un gruppo di parametri personalizzati, modificare l'istanza database per il collegamento. Per ulteriori informazioni, consulta Modifica di un'istanza database Amazon RDS.

Per ottimizzare l'istanza database di Oracle per Siebel CRM, è possibile personalizzare determinati parametri. La tabella riportata di seguito mostra alcune impostazioni consigliate per il parametro. Per ulteriori informazioni sulla regolazione delle prestazioni di Siebel CRM, consulta la Guida alla regolazione delle prestazioni di Siebel CRM.

Nome del parametro	Valore predefinito	Guida per prestazioni ottimali di Siebel CRM
`_always_semi_join`	`CHOOSE`	`OFF`
`_b_tree_bitmap_plans`	`TRUE`	`FALSE`
`_like_with_bind_as_equality`	`FALSE`	`TRUE`
`_no_or_expansion`	`FALSE`	`FALSE`
`_optimizer_join_sel_sanity_check`	`TRUE`	`TRUE`
`_optimizer_max_permutations`	2000	100

Nome del parametro	Valore predefinito	Guida per prestazioni ottimali di Siebel CRM
`_optimizer_sortmerge_join_enabled`	`TRUE`	`FALSE`
`_partition_view_enabled`	`TRUE`	`FALSE`
`open_cursors`	`300`	Almeno **`2000`**.

Creazione di snapshot

Dopo aver creato il database di Siebel, è possibile copiare il database usando le caratteristiche di snapshot di Amazon RDS. Per ulteriori informazioni, consulta Creazione di uno snapshot DB per un'istanza DB Single-AZ e Ripristino da uno snapshot database.

Supporto per altri Componenti Siebel CRM

Oltre al database Siebel, è anche possibile utilizzare Amazon Web Services per supportare gli altri componenti dall'architettura dell'applicazione Siebel CRM. È possibile trovare altre informazioni sul supporto fornite da Amazon AWS per i componenti aggiuntivi di Siebel CRM nella tabella seguente.

Componente Siebel CRM	AWS Assistenza Amazon
Siebel Enterprise (con uno o più server Siebel)	Puoi ospitare i server Siebel in istanze Amazon Elastic Compute Cloud (Amazon EC2). Puoi utilizzare Amazon EC2 per avviare il numero di server virtuali necessari. Usando Amazon EC2, puoi ridurre o aumentare facilmente per gestire i cambiamenti nei requisiti. Per ulteriori informazioni, consulta Che cos'è Amazon EC2? Puoi inserire i server nello stesso VPC con l'istanza database e usare il gruppo di sicurezza VPC per

Componente Siebel CRM	AWS Assistenza Amazon
	accedere al database. Per ulteriori informazioni, consulta Uso di un'istanza database in un VPC.
Server Web (con Siebel Web Server Extensions)	È possibile installare più server Web su più istanze EC2. Per utilizzare Elastic Load Balancing per distribuire il traffico in ingresso tra le istanze. Per ulteriori informazioni, consulta Che cos'è Elastic Load Balancing?
Siebel Gateway Name Server	Puoi ospitare i server Siebel Gateway Name Server nell'istanza EC2. Puoi quindi inserire i server nello stesso VPC con l'istanza database e usare il gruppo di sicurezza VPC per accedere al database. Per ulteriori informazioni, consulta Uso di un'istanza database in un VPC.

Note di rilascio del motore di database Oracle

Aggiorna le istanze database Amazon RDS for Oracle per mantenerle aggiornate. Se applichi gli aggiornamenti, puoi essere certo che l'istanza database esegue una versione del software di database che è stata testata da Oracle e Amazon. Non supportiamo l'applicazione di patch occasionali a singole istanze database RDS per Oracle.

Quando crei una nuova istanza database, puoi specificare qualsiasi versione di Oracle Database attualmente supportata. Si può specificare la versione principale, ad esempio Oracle Database 19c, e qualsiasi versione secondaria supportata per la versione principale specificata. Se non viene specificata alcuna versione, Amazon RDS utilizza per impostazione predefinita una versione supportata, in genere la più recente. Se viene specificata una versione principale ma non una secondaria, per impostazione predefinita Amazon RDS utilizza una release recente della versione principale specificata. Per visualizzare un elenco di versioni supportate e di valori predefiniti per le nuove istanze database create, utilizza il comando della AWS CLI `describe-db-engine-versions`.

Per informazioni dettagliate sulle versioni di Oracle Database supportate da Amazon RDS, consulta le Note di rilascio di Amazon RDS per Oracle.

Amazon RDS per PostgreSQL

Amazon RDS supporta istanze database che eseguono diverse versioni di PostgreSQL. Per un elenco dettagliato delle versioni disponibili, consulta Versioni del database PostgreSQL disponibili.

> **Note**
>
> L'obsolescenza di PostgreSQL 9.6 è prevista per il 26 aprile 2022. Per ulteriori informazioni, consulta Obsolescenza di PostgreSQL versione 9.6.

È possibile creare istanze DB e snapshot DB, point-in-time ripristini e backup. Le istanze database che eseguono PostgreSQL supportano implementazioni Multi-AZ, repliche di lettura e IOPS con provisioning e possono essere create all'interno di un cloud privato virtuale (VPC). Puoi inoltre utilizzare il protocollo SSL (Secure Socket Layer) per connetterti a un'istanza database che esegue PostgreSQL.

Prima di creare un'istanza database, assicurati di completare i passaggi indicati in Configurazione di Amazon RDS.

Puoi utilizzare qualsiasi applicazione client SQL standard per eseguire i comandi per l'istanza dal tuo computer client. Queste applicazioni includono pgAdmin, uno strumento di amministrazione e sviluppo open source per PostgreSQL ampiamente diffuso, oppure psql, utilità a riga di comando che fa parte di un'installazione di PostgreSQL. Per offrire un'esperienza di servizio gestito, Amazon RDS non fornisce accesso host alle istanze database. Limita anche l'accesso ad alcune procedure di sistema e tabelle che richiedono privilegi avanzati. Amazon RDS supporta l'accesso ai database in un'istanza database con qualsiasi applicazione client SQL standard. Amazon RDS non consente l'accesso host diretto a un'istanza database utilizzando Telnet o Secure Shell (SSH).

Amazon RDS for PostgreSQL è conforme a molti standard di settore. Ad esempio, è possibile utilizzare i database Amazon RDS for PostgreSQL per creare applicazioni conformi a HIPAA e archiviare informazioni sanitarie. Ciò include l'archiviazione per informazioni sanitarie protette (PHI) in base a un Contratto di società in affari (BAA) completo stipulato con AWS. Amazon RDS for PostgreSQL soddisfa inoltre i requisiti di sicurezza Federal Risk and Authorization Management Program (FedRAMP). Amazon RDS for PostgreSQL ha ricevuto una Provisional Authority to Operate (P-ATO) del FedRAMP Joint Authorization Board (JAB) presso la FedRAMP HIGH Baseline all'interno delle regioni. AWS GovCloud (US) Per ulteriori informazioni sugli standard di conformità supportati, consulta Conformità di AWS Cloud.

Per importare i dati PostgreSQL in un'istanza database, segui le informazioni contenute nella sezione Importazione di dati in PostgreSQL su Amazon RDS.

Argomenti

Attività di gestione frequenti per Amazon RDS for PostgreSQL

Di seguito sono riportate le attività di gestione più frequenti che puoi eseguire con un'istanza database Amazon RDS for PostgreSQL;, con collegamenti alla documentazione rilevante per ciascuna attività.

Area attività	Documentazione di riferimento
Configurazione di Amazon RDS per il primo utilizzo Prima di poter creare l'istanza database, devi completare alcuni prerequisiti. Ad esempio, le istanze database vengono create per impostazione predefinita con un firewall che ne impedisce l'accesso. Quindi è necessario creare un gruppo di sicurezza con gli indirizzi IP e la configurazione di rete corretti per accedere all'istanza database.	Configurazione di Amazon RDS
Informazioni sulle istanze database Amazon RDS Se stai creando un'istanza database per la produzione, è necessario comprendere come funzionano in Amazon RDS le classi di istanze, i tipi di storage e le Provisioned IOPS.	Classi di istanze database Tipi di storage Amazon RDS Storage SSD Provisioned IOPS
Ricerca delle versioni PostgreSQL disponibili Amazon RDS supporta diverse versioni di PostgreSQL.	Versioni del database PostgreSQL disponibili
Configurazione di elevata disponibilità e supporto per il failover Un'istanza database in produzione deve utilizzare implement azioni Multi-AZ. Le implementazioni Multi-AZ forniscono alle istanze database maggior disponibilità, longevità dei dati e tolleranza ai guasti.	Configurazione e gestione di un'implementazione multi-AZ
Informazioni sulla rete Amazon Virtual Private Cloud (VPC) Se il tuo AWS account ha un VPC predefinito, l'istanza DB viene creata automaticamente all'interno del VPC predefinito. In alcuni casi, l'account potrebbe non avere un VPC predefinito e può	Uso di un'istanza database in un VPC

Area attività	Documentazione di riferimento
essere opportuno avere l'istanza database in un VPC. In questi casi, creare il VPC e i gruppi di sottoreti prima di creare l'istanza database.	
Importazione di dati in PostgreSQL Amazon RDS Puoi utilizzare diversi strumenti per importare i dati nella tua istanza database PostgreSQL su Amazon RDS.	Importazione di dati in PostgreSQL su Amazon RDS
Configurazione di repliche di lettura di sola lettura (master e standby) RDS per PostgreSQL supporta le repliche di lettura sia nella AWS stessa regione che in una regione diversa dall'istanza principale. AWS	Uso delle repliche di lettura dell'istanza database Utilizzo delle repliche di lettura per Amazon RDS per PostgreSQL Creazione di una replica di lettura in un altro Regione AWS
Comprendere i gruppi di sicurezza Per impostazione predefinita, le istanze database vengono create con un firewall che ne impedisce l'accesso. Per fornire l'accesso tramite tale firewall, è necessario modificare le regole in entrata per il gruppo di sicurezza VPC associato al VPC che ospita l'istanza database.	Controllo dell'accesso con i gruppi di sicurezza
Configurazione dei gruppi di parametri e delle relative caratteri stiche Per modificare i parametri predefiniti per l'istanza database, crea un gruppo parametri del database personalizzato e modifica le relative impostazioni. Se esegui questa operazione prima di creare l'istanza database, puoi scegliere il gruppo parametri del database personalizzato quando crei l'istanza.	Utilizzo di gruppi di parametri

Area attività	Documentazione di riferimento
Connessione a un'istanza database PostgreSQL Dopo aver creato un gruppo di sicurezza e averlo associato a un'istanza database, è possibile effettuare la connessione all'istanza database mediante un'applicazione cliente SQL standard, come `psql` o `pgAdmin`.	Connessione a un'istanza database che esegua il motore di database di PostgreSQL Utilizzo del protocollo SSL con un'istanza database PostgreSQL
Backup e ripristino dell'istanza database È possibile configurare l'istanza database affinché effettui backup automatici o acquisisca snapshot manuali e poi esegua il ripristino istanze da backup o snapshot.	Backup, ripristino ed esportazione dei dati
Monitoraggio dell'attività e delle prestazioni dell'istanza database Puoi monitorare un'istanza DB PostgreSQL utilizzando metriche, eventi e monitoraggio avanzato di CloudWatch Amazon RDS.	Visualizzazione dei parametri nella console Amazon RDS Visualizzazione di eventi Amazon RDS
Aggiornamento della versione del database PostgreSQL. Puoi effettuare aggiornamenti a versioni principali e secondarie per la tua istanza database PostgreSQL.	Aggiornamento del motore del database PostgreSQL per Amazon RDS Scelta di un aggiornamento di versione principale per PostgreSQL
Utilizzo dei file di log Puoi accedere ai file di log per le tue istanze database PostgreSQL.	File di log del database RDS per PostgreSQL
Comprendere le best practice per le istanze database PostgreSQL Scopri alcune delle best practice per lavorare con PostgreSQL su Amazon RDS.	Best practice per l'utilizzo di PostgreSQL

Di seguito è riportato un elenco di altre sezioni di questa guida che possono aiutarti a comprendere e utilizzare importanti caratteristiche di RDS for PostgreSQL:

- Informazioni su ruoli e autorizzazioni di PostgreSQL
- Controllo dell'accesso utente al database PostgreSQL
- Utilizzo dei parametri sull'istanza database RDS for PostgreSQL
- Comprensione dei meccanismi di registrazione supportati da RDS for PostgreSQL
- Utilizzo della funzione di autovacuum di PostgreSQL in Amazon RDS for PostgreSQL
- Utilizzo di un Server DNS personalizzato per Outbound Network Access.

Utilizzo dell'ambiente di anteprima del database

La community PostgreSQL rilascia di continuo nuove versioni ed estensioni di PostgreSQL, incluse le versioni beta. In tal modo gli utenti PostgreSQL hanno l'opportunità di provare una nuova versione di PostgreSQL in anticipo. Per ulteriori informazioni sul processo di rilascio della versione beta della community PostgreSQL, consulta Beta Information (Informazioni sulla versione beta) nella documentazione di PostgreSQL. Analogamente, Amazon RDS rende disponibili alcune versioni beta di PostgreSQL come versioni di anteprima. In tal modo puoi creare istanze database utilizzando la versione di anteprima e testarne le funzionalità nell'ambiente di anteprima del database.

Le istanze database RDS per PostgreSQL nell'ambiente di anteprima del database sono funzionalmente simili alle altre istanze database per PostgreSQL. Tuttavia, una versione di anteprima non può essere utilizzata in produzione.

Tieni presenti le importanti limitazioni riportate di seguito:

- Tutte le istanze database vengono eliminate 60 giorni dopo la creazione, insieme a eventuali backup e snapshot.
- Puoi creare un'istanza database solo in un virtual private cloud (VPC) basato sul servizio Amazon VPC.
- Puoi utilizzare solo lo storage General Purpose (SSD) e Provisioned IOPS (SSD).
- Non puoi ricevere assistenza dal AWS Supporto per le istanze DB. Puoi invece pubblicare le tue domande nella community di domande e risposte AWS gestita, re:POST.AWS
- Non puoi copiare uno snapshot di un'istanza database in un ambiente di produzione.

Le seguenti opzioni sono supportate dall'anteprima.

- È possibile creare istanze database usando solo i tipi di istanza M6i, R6i, M6g, M5, T3, R6g e R5. Per ulteriori informazioni sulle classi delle istanze RDS, consulta Classi di istanze database.
- È possibile utilizzare distribuzioni AZ singola e Multi-AZ.
- È possibile utilizzare le funzioni di dump e caricamento standard di PostgreSQL per esportare database da o importare database nell'ambiente di anteprima database.

Funzionalità non supportate nell'ambiente di anteprima del database

Le seguenti funzionalità non sono disponibili nell'ambiente di anteprima del database:

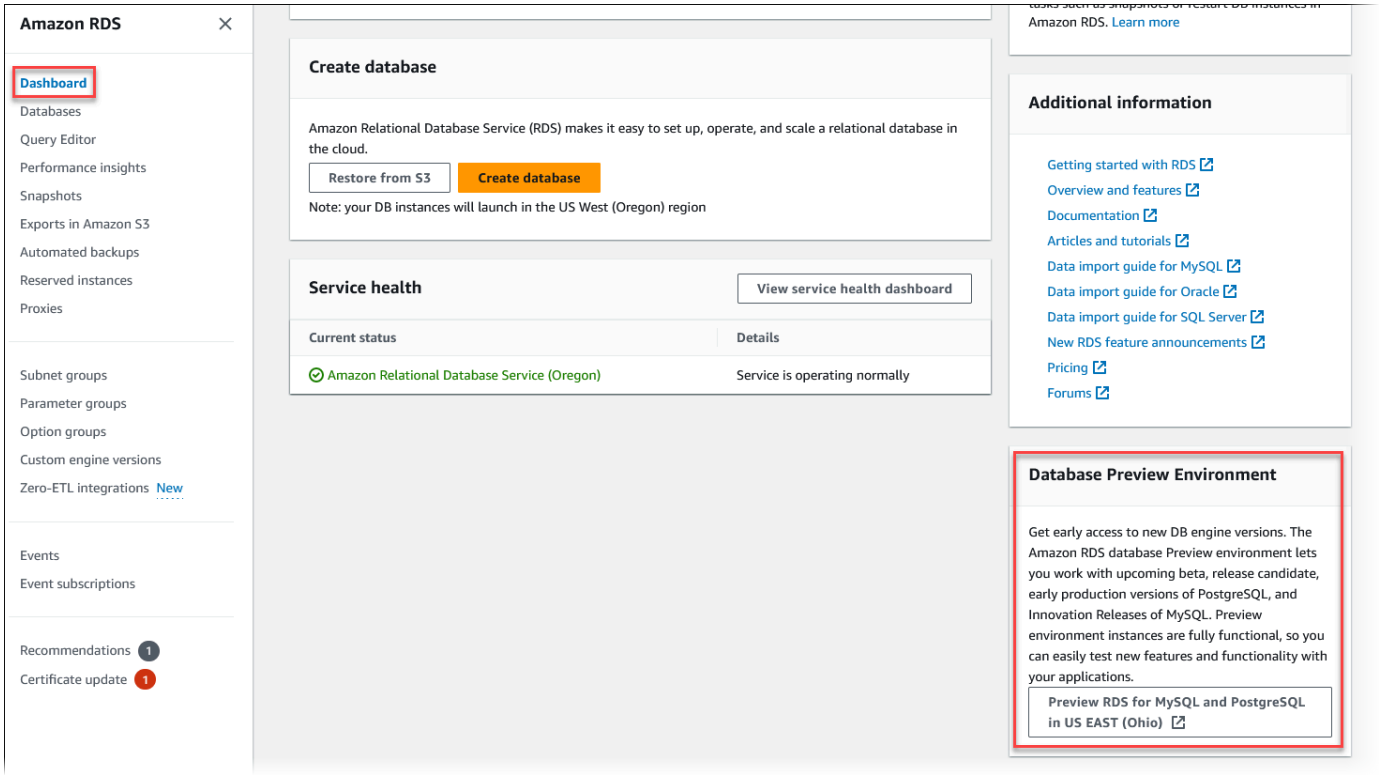
- Copia di snapshot tra regioni diverse
- Repliche di lettura tra regioni diverse

Creazione di una nuova istanza database nell'ambiente di anteprima del database

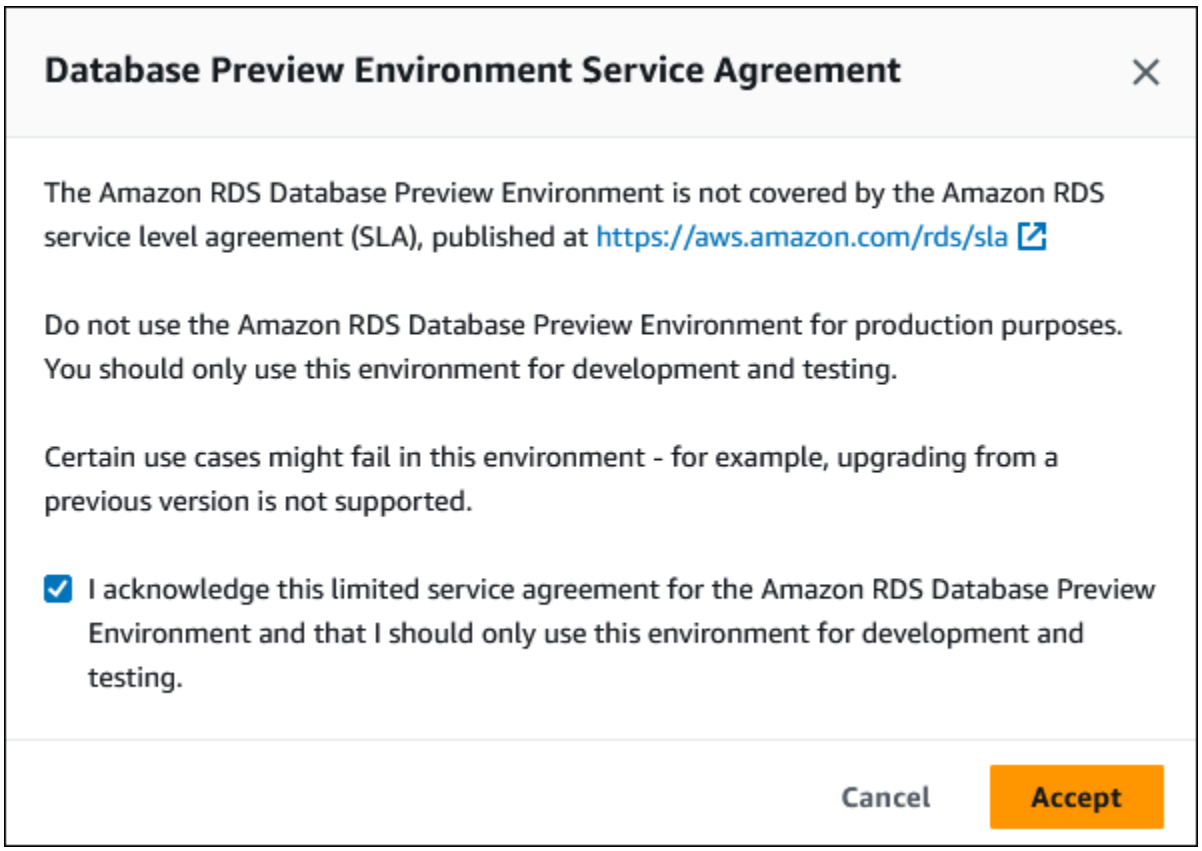
Utilizza la seguente procedura per creare un'istanza database nell'ambiente di anteprima.

Per creare un'istanza database nell'ambiente di anteprima del database

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Scegliere Dashboard (Pannello di controllo) nel pannello di navigazione.
3. Nella pagina Dashboard (Pannello di controllo), individua la sezione Database Preview Environment (Ambiente di anteprima del database), come mostrato nell'immagine seguente.



L'ambiente di anteprima del database è accessibile direttamente. Prima di poter procedere, è necessario capire e accettare le limitazioni.



4. Per creare l'istanza database RDS per PostgreSQL, segui la stessa procedura utilizzata per creare qualsiasi istanza database Amazon RDS. Per ulteriori informazioni, consulta la procedura Console in Creazione di un'istanza database.

Per creare un'istanza nel Database Preview Environment utilizzando l'API RDS o il AWS CLI, utilizza il seguente endpoint.

```
rds-preview.us-east-2.amazonaws.com
```

PostgreSQL versione 17 nell'ambiente Database Preview

⚠ Questa è la documentazione di anteprima per Amazon RDS PostgreSQL versione 17. ed è soggetta a modifiche.

La versione 17 Beta 1 di PostgreSQL è ora disponibile nell'ambiente Amazon RDS Database Preview. La versione 17 Beta 1 di PostgreSQL contiene diversi miglioramenti descritti nella seguente documentazione di PostgreSQL: PostgreSQL 17 Beta 1 rilasciato!

Per informazioni sull'ambiente di anteprima del database, consulta the section called “ Ambiente di anteprima del database”. Per accedere all'ambiente di anteprima dalla console, selezionare https://console.aws.amazon.com/rds-preview/.

PostgreSQL versione 16 nell'ambiente di anteprima del database

⚠ Questa è la documentazione di anteprima per Amazon RDS PostgreSQL versione 16. ed è soggetta a modifiche.

Note

Le versioni 16 RC1, 16 Beta 3, 16 Beta 2 e 16 Beta 1 di RDS per PostgreSQL non saranno supportate dopo il rilascio della versione 16.0 di RDS per PostgreSQL nell'ambiente di anteprima del database.

PostgreSQL versione 16.0 è ora disponibile nell'ambiente di anteprima del database Amazon RDS. PostgreSQL versione 16 include vari miglioramenti, descritti nella seguente documentazione PostgreSQL:

- Rilascio di PostgreSQL 16
- Rilascio di PostgreSQL 16 RC1
- Rilascio di PostgreSQL 16 Beta 3
- Rilascio di PostgreSQL 16 Beta 2
- PostgreSQL 16 Beta 1 rilasciata

Per informazioni sull'ambiente di anteprima del database, consulta the section called “ Ambiente di anteprima del database”. Per accedere all'ambiente di anteprima dalla console, selezionare https://console.aws.amazon.com/rds-preview/.

Versioni del database PostgreSQL disponibili

Amazon RDS supporta le istanze database che eseguono diverse edizioni di PostgreSQL. Quando crei una nuova istanza database, puoi specificare qualsiasi versione di MySQL attualmente disponibile. Puoi specificare la versione principale (come PostgreSQL 14) e qualsiasi versione secondaria disponibile per la versione principale specificata. Se non viene specificata alcuna versione, Amazon RDS utilizza per impostazione predefinita una versione disponibile, in genere la più recente. Se viene specificata una versione principale ma non una secondaria, per impostazione predefinita Amazon RDS utilizza una release recente della versione principale specificata.

Per visualizzare un elenco delle versioni disponibili, nonché i valori predefiniti per le istanze DB appena create, usa il comando. `describe-db-engine-versions` AWS CLI Ad esempio, per visualizzare la versione predefinita del motore PostgreSQL, utilizza il seguente comando:

```
aws rds describe-db-engine-versions --default-only --engine postgres
```

Per informazioni dettagliate sulle versioni di PostgreSQL supportate su Amazon RDS, consultare la sezione Amazon RDS for PostgreSQL Release Notes.

Se non sei pronto per l'aggiornamento manuale a una nuova versione principale del motore prima della data di fine del supporto standard RDS, Amazon RDS registrerà automaticamente i tuoi database in Amazon RDS Extended Support dopo la data di fine del supporto standard RDS. Quindi, puoi continuare a eseguire RDS per PostgreSQL versione 11 e successive. Per ulteriori informazioni, consulta Utilizzo dell'estensione del supporto per Amazon RDS and Prezzi di Amazon RDS.

Definizione come obsoleto di PostgreSQL versione 10

Il 17 aprile 2023 Amazon RDS prevede di rendere obsoleto PostgreSQL 10 in base alla seguente pianificazione. Ti consigliamo di agire e aggiornare i database PostgreSQL in esecuzione sulla versione principale 10 a una versione successiva, come PostgreSQL versione 14. Per aggiornare l'istanza database della versione principale 10 di RDS per PostgreSQL da una versione di PostgreSQL precedente alla 10.19, consigliamo di eseguire prima l'aggiornamento alla versione 10.19 e quindi quello alla versione 14. Per ulteriori informazioni, consulta Aggiornamento del motore del database PostgreSQL per Amazon RDS.

Azione o raccomandazione	Date:
La community di PostgreSQL prevede di rendere obsoleto PostgreSQL 10 e non fornirà alcuna patch di sicurezza dopo questa data.	10 novembre 2022
Inizia ad aggiornare le istanze database RDS per PostgreSQL 10 a una versione principale successiv a, come PostgreSQL 14. Sebbene sia possibile continuare a ripristinare gli snapshot PostgreSQL 10 e creare repliche di lettura con la versione 10, tieni presente le altre date critiche di questa pianificazione della definizione di obsoleto e del loro impatto.	Fino al 14 febbraio 2023
Dopo questa data, non puoi creare nuove istanze Amazon RDS con la versione principale di PostgreSQL 10 da o da. AWS Management Console AWS CLI	14 febbraio 2023
Dopo questa data, Amazon RDS aggiorna automatic amente le istanze PostgreSQL 10 alla versione 14. Se si ripristina uno snapshot del database PostgreSQL 10, Amazon RDS aggiorna automatic amente il database ripristinato a PostgreSQL 14.	17 aprile 2023

Per ulteriori informazioni sulla deprecazione di RDS per PostgreSQL versione 10, vedere [Annuncio]: deprecazione di RDS per PostgreSQL 10 in re:POST. AWS

Obsolescenza di PostgreSQL versione 9.6

Il 31 marzo 2022 Amazon RDS prevede di rendere obsoleto PostgreSQL 9.6 in base alla seguente pianificazione. Ciò estende la data precedentemente annunciata dal 18 gennaio 2022 al 26 aprile 2022. Ti consigliamo di aggiornare tutte le istanze database di PostgreSQL 9.6 a PostgreSQL 12 o versione successiva il prima possibile. Ti consigliamo di eseguire prima l'aggiornamento alla versione secondaria 9.6.20 o successiva e quindi eseguire l'aggiornamento direttamente a PostgreSQL 12 anziché eseguire l'aggiornamento a una versione principale intermedia. Per ulteriori informazioni, consulta Aggiornamento del motore del database PostgreSQL per Amazon RDS.

Azione o raccomandazione	Date:
La community PostgreSQL ha interrotto il supporto per PostgreSQL 9.6 e non fornirà più correzioni di bug o patch di sicurezza per questa versione.	11 novembre 2021
Inizia ad aggiornare le istanze database di RDS for PostgreSQL 9.6 a PostgreSQL 12 o versione successiva il prima possibile. Sebbene sia possibile continuare a ripristinare gli snapshot PostgreSQL 9.6 e creare repliche di lettura con la versione 9.6, tieni presente le altre date critiche di questa pianificazione di obsolescenza e del loro impatto.	Fino al 31 marzo 2022
Dopo questa data, non puoi creare nuove istanze Amazon RDS con la versione principale di PostgreSQL 9.6 né dalla. AWS Management Console AWS CLI	31 marzo 2022
Dopo tale data, Amazon RDS aggiorna automatic amente le istanze PostgreSQL 9.6 alla versione 12. Se si ripristina uno snapshot del database PostgreSQL 9.6, Amazon RDS aggiorna automatic amente il database ripristinato a PostgreSQL 12.	26 Aprile 2022

Versioni obsolete per Amazon RDS for PostgreSQL

RDS for PostgreSQL 9.5 è stato reso obsoleto a partire da marzo 2021. Per ulteriori informazioni sulla deprecazione di RDS for PostgreSQL 9.5, vedere Aggiornamento dalla versione 9.5. Amazon RDS for PostgreSQL

Per ulteriori informazioni sulla policy di deprecazione per RDS for PostgreSQL, consulta Domande frequenti su Amazon RDS. Per ulteriori informazioni sulle versioni di PostgreSQL, consulta Versioning Policy (Policy di controllo delle versioni) nella documentazione di PostgreSQL.

Versioni con estensione PostgreSQL supportate

RDS per PostgreSQL supporta numerose estensioni PostgreSQL. La community PostgreSQL a volte si riferisce a questi come moduli. Le estensioni espandono la funzionalità fornita dal motore PostgreSQL. Trovi l'elenco delle estensioni supportate da Amazon RDS nel gruppo di parametri database predefinito per quella versione di PostgreSQL. Puoi inoltre vedere l'elenco delle estensioni correnti che utilizzano `psql` mostrando il parametro `rds.extensions` come nel seguente esempio.

```
SHOW rds.extensions;
```

> **Note**
>
> I parametri aggiunti in una versione minore possono essere visualizzati in modo non corretto quando si utilizza il parametro `rds.extensions` in `psql`.

A partire da RDS per PostgreSQL 13, alcune estensioni possono essere installate da utenti del database diversi da `rds_superuser`. Tali estensioni sono da considerare estensioni attendibili. Per ulteriori informazioni, consulta Estensioni attendibili di PostgreSQL.

Alcune versioni di RDS per PostgreSQL supportano il parametro `rds.allowed_extensions`. Questo parametro consente a un utente `rds_superuser` di limitare le estensioni installabili nell'istanza database RDS per PostgreSQL. Per ulteriori informazioni, consulta Limitazione dell'installazione delle estensioni PostgreSQL.

Per un elenco delle estensioni e versioni di PostgreSQL supportate da ogni versione disponibile di RDS per PostgreSQL, consulta Estensioni PostgreSQL supportate su Amazon RDS nelle Note di rilascio di Amazon RDS per PostgreSQL.

Limitazione dell'installazione delle estensioni PostgreSQL

È possibile limitare quali estensioni possono essere installate su un'istanza database PostgreSQL. Per impostazione predefinita, questo parametro non è impostato e pertanto è possibile aggiungere qualsiasi estensione supportata se l'utente dispone delle autorizzazioni corrispondenti. A tale scopo, impostare il parametro `rds.allowed_extensions` su una stringa di nomi di estensione separati da virgole. L'aggiunta di un elenco di estensioni a questo parametro consente di identificare esplicitamente le estensioni che l'istanza database RDS per PostgreSQL può utilizzare. Solo queste estensioni possono quindi essere installate nell'istanza database di PostgreSQL.

La stringa predefinita per il parametro `rds.allowed_extensions` è '*', il che significa che qualsiasi estensione disponibile per la versione del motore può essere installata. La modifica del parametro `rds.allowed_extensions` non richiede il riavvio del database perché si tratta di un parametro dinamico.

Il motore di istanze database PostgreSQL deve essere una delle seguenti versioni per poter utilizzare il parametro `rds.allowed_extensions`:

- Tutte le versioni di PostgreSQL 16
- PostgreSQL 15 e tutte le versioni successive
- Aurora PostgreSQL 14 e tutte le versioni successive
- PostgreSQL 13.3 e versioni secondarie successive
- PostgreSQL 12.7 e versioni secondarie successive

Per vedere quali installazioni di estensione sono consentite, utilizzare il seguente comando psql.

```
postgres=> SHOW rds.allowed_extensions;
 rds.allowed_extensions
------------------------
 *
```

Se un'estensione è stata installata prima di essere lasciata fuori dall'elenco nel `rds.allowed_extensions` parametro, l'estensione può comunque essere utilizzata normalmente e comandi come `ALTER EXTENSION` e `DROP EXTENSION` continueranno a funzionare. Tuttavia, dopo che un'estensione è stata limitata, i comandi `CREATE EXTENSION` per l'estensione con restrizioni avranno esito negativo.

Anche l'installazione di dipendenze di estensione con `CREATE EXTENSION CASCADE` sono limitate. L'estensione e le relative dipendenze devono essere specificate in `rds.allowed_extensions`. Se un'installazione delle dipendenze di estensione non riesce, l'intera istruzione `CREATE EXTENSION CASCADE` avrà esito negativo.

Se un'estensione non è inclusa nel `rds.allowed_extensions` parametro, verrà visualizzato un errore come il seguente se si tenta di installarla.

```
ERROR: permission denied to create extension "extension-name"
HINT: This extension is not specified in "rds.allowed_extensions".
```

Estensioni attendibili di PostgreSQL

L'installazione della maggior parte delle estensioni PostgreSQL richiede privilegi `rds_superuser`. PostgreSQL 13 ha introdotto le estensioni attendibili, che riducono la necessità di concedere privilegi `rds_superuser` agli utenti regolari. Con questa funzione, gli utenti possono installare molte estensioni se dispongono del privilegio `CREATE` sul database corrente invece di richiedere il ruolo `rds_superuser`. Per ulteriori informazioni, consulta il comando CREATE EXTENSION SQL nella documentazione di PostgreSQL.

Di seguito sono elencate le estensioni che possono essere installate da un utente che dispone del privilegio `CREATE` sul database corrente e non richiedono il ruolo `rds_superuser`:

- bool_plperl
- btree_gin
- btree_gist
- citext
- cube
- dict_int
- fuzzystrmatch
- hstore
- intarray
- isn
- jsonb_plperl
- ltree
- pg_trgm
- pgcrypto
- plperl
- plpgsql
- pltcl
- tablefunc
- tsm_system_rows
- tsm_system_time
- unaccent

- uuid-ossp

Per un elenco delle estensioni e versioni di PostgreSQL supportate da ogni versione disponibile di RDS per PostgreSQL, consulta PostgreSQL extensions supported on Amazon RDS nelle Note di rilascio di Amazon RDS per PostgreSQL.

Utilizzo delle caratteristiche di PostgreSQL supportate da Amazon RDS for PostgreSQL

Amazon RDS per PostgreSQL supporta molte delle funzioni PostgreSQL più comuni. Ad esempio, PostgreSQL ha una funzione di pulizia automatica che esegue la manutenzione di routine sul database. La funzione autovacuum è attivata per impostazione predefinita. Sebbene sia possibile disattivare questa caratteristica, è consigliabile tenerla attiva. Comprendere questa funzionalità e cosa puoi fare per assicurarti che funzioni come dovrebbe è un compito di base di qualsiasi DBA. Per ulteriori informazioni sulla caratteristica autovacuum, consulta Utilizzo della funzione di autovacuum di PostgreSQL in Amazon RDS for PostgreSQL. Per ulteriori informazioni su altre attività DBA comuni, Attività DBA comuni per Amazon RDS for PostgreSQL.

RDS for PostgreSQL supporta anche estensioni che aggiungono funzionalità importanti all'istanza database. Ad esempio, è possibile utilizzare l'estensione PostGIS per lavorare con i dati spaziali o utilizzare l'estensione pg_cron per pianificare la manutenzione dall'istanza. Per ulteriori informazioni sulle estensioni PostgreSQL, consulta Utilizzo delle estensioni PostgreSQL con Amazon RDS for PostgreSQL.

I wrapper di dati esterni sono un tipo specifico di estensione progettato per consentire all'istanza database RDS for PostgreSQL di lavorare con altri tipi di dati o database commerciali. Per ulteriori informazioni sui wrapper di dati esterni supportati da RDS for PostgreSQL, consulta Utilizzo dei wrapper di dati esterni supportati per Amazon RDS for PostgreSQL.

Di seguito sono riportate informazioni su alcune altre caratteristiche supportate da RDS per PostgreSQL.

Argomenti

- Tipi di dati personalizzati ed enumerazioni con RDS for PostgreSQL
- Trigger di eventi per RDS for PostgreSQL
- Pagine di grandi dimensioni per RDS for PostgreSQL
- Esecuzione della replica logica per Amazon RDS for PostgreSQL
- Disco RAM per stats_temp_directory
- Spazi tabelle per RDS for PostgreSQL
- Regole di confronto RDS per PostgreSQL per EBCDIC e altre migrazioni di mainframe

Tipi di dati personalizzati ed enumerazioni con RDS for PostgreSQL

PostgreSQL supporta la creazione di tipi di dati personalizzati e l'utilizzo delle enumerazioni. Per ulteriori informazioni sulla creazione e sull'utilizzo di enumerazioni e altri tipi di dati, consulta Tipi enumerati nella documentazione di PostgreSQL.

Di seguito è riportato un esempio di creazione di un tipo come enumerazione e quindi di inserimento di valori in una tabella.

```
CREATE TYPE rainbow AS ENUM ('red', 'orange', 'yellow', 'green', 'blue', 'purple');
CREATE TYPE
CREATE TABLE t1 (colors rainbow);
CREATE TABLE
INSERT INTO t1 VALUES ('red'), ( 'orange');
INSERT 0 2
SELECT * from t1;
colors
--------
red
orange
(2 rows)
postgres=> ALTER TYPE rainbow RENAME VALUE 'red' TO 'crimson';
ALTER TYPE
postgres=> SELECT * from t1;
colors
---------
crimson
orange
(2 rows)
```

Trigger di eventi per RDS for PostgreSQL

Tutte le versioni attuali di PostgreSQL supportano i trigger di eventi, così come tutte le versioni disponibili di RDS for PostgreSQL. Puoi utilizzare l'account utente principale (di default `postgres`) per creare, modificare, rinominare ed eliminare i trigger degli eventi. I trigger di eventi sono a livello di istanza database, quindi possono essere applicati a tutti i database in un'istanza.

Ad esempio, il seguente codice crea un trigger di eventi che stampa l'utente corrente alla fine di ogni comando DDL (Data Definition Language).

```
CREATE OR REPLACE FUNCTION raise_notice_func()
```

```
    RETURNS event_trigger
    LANGUAGE plpgsql AS
$$
BEGIN
    RAISE NOTICE 'In trigger function: %', current_user;
END;
$$;

CREATE EVENT TRIGGER event_trigger_1
    ON ddl_command_end
EXECUTE PROCEDURE raise_notice_func();
```

Per ulteriori informazioni sui trigger di eventi PostgreSQL, consulta Trigger di eventi nella documentazione di PostgreSQL.

L'utilizzo dei trigger di eventi PostgreSQL su Amazon RDS prevede diverse limitazioni. Questi sono i seguenti:

- Non puoi creare trigger di eventi su repliche di lettura. Tuttavia, puoi creare trigger di eventi su una replica di lettura master. I trigger di eventi vengono quindi copiati nella replica di lettura. I trigger di eventi sulla replica di lettura non vengono attivati sulla replica di lettura in caso di modifiche da parte del master. Tuttavia, se viene promossa la replica di lettura, i trigger di eventi esistenti si attivano in caso di operazioni del database.
- Per eseguire un aggiornamento principale della versione a un'istanza database PostgreSQL che utilizza i trigger di eventi, assicurati di eliminare i trigger di eventi prima dell'aggiornamento dell'istanza.

Pagine di grandi dimensioni per RDS for PostgreSQL

Le pagine di grandi dimensioni sono una caratteristica di gestione della memoria che riduce il sovraccarico quando un'istanza database lavora con grandi blocchi di memoria contigui, come quelli utilizzati dai buffer condivisi. Questa caratteristica di PostgreSQL è supportata da tutte le versioni RDS for PostgreSQL attualmente disponibili. Allocate enormi pagine per la vostra applicazione utilizzando chiamate `mmap` o `SYSV` memoria condivisa. RDS for PostgreSQL supporta pagine da 4 KB e 2 MB.

È possibile attivare o disattivare pagine di grandi dimensioni modificando il valore del parametro `huge_pages`. La caratteristica è attivata per impostazione predefinita per tutte le classi di istanze database diverse dalle classi di istanza micro, small e medium.

RDS for PostgreSQL usa le pagine di grandi dimensioni in base alla memoria condivisa disponibile. Se l'istanza database non è in grado di utilizzare le pagine di grandi dimensioni a causa dei limiti della memoria condivisa, Amazon RDS impedisce l'avvio dell'istanza database. In questo caso, Amazon RDS imposta lo stato dell'istanza database su uno stato dei parametri incompatibile. In questo caso, puoi impostare il parametro `huge_pages` su `off` per permettere a Amazon RDS di avviare l'istanza database.

Il parametro `shared_buffers` è essenziale per impostare il pool della memoria condivisa richiesto per l'utilizzo delle pagine di grandi dimensioni. Il valore predefinito per il parametro `shared_buffers` utilizza una macro dei parametri del database. Questa macro imposta una percentuale di 8 KB in totale disponibili per la memoria dell'istanza database. Quando si utilizzano pagine di dimensioni enormi, queste pagine vengono allocate insieme alle pagine enormi. Amazon RDS inserisce un'istanza database in uno stato di parametri non compatibile se i parametri della memoria condivisa sono impostati in modo da richiedere più del 90% della memoria dell'istanza database.

Per ulteriori informazioni sulla gestione della memoria PostgreSQL, consulta Consumo di risorse nella documentazione di PostgreSQL.

Esecuzione della replica logica per Amazon RDS for PostgreSQL

A partire dalla versione 10.4, RDS per PostgreSQL supporta la pubblicazione e la sottoscrizione della sintassi SQL introdotta in PostgreSQL 10. Per ulteriori informazioni, consulta Replica logica nella documentazione di PostgreSQL.

> **Note**
>
> Oltre alla funzionalità di replica logica nativa di PostgreSQL introdotta in PostgreSQL 10, RDS per PostgreSQL supporta anche l'estensione `pglogical`. Per ulteriori informazioni, consulta Utilizzo di pglogical per sincronizzare i dati tra le istanze.

Di seguito sono riportate informazioni sull'impostazione della replica logica per un'istanza database RDS for PostgreSQL.

Argomenti

- Comprendere la replica logica e la decodifica logica
- Lavorare con gli slot di replica logica

Comprendere la replica logica e la decodifica logica

RDS for PostgreSQL supporta lo streaming delle modifiche write-ahead log (WAL) utilizzando slot di replica logica di PostgreSQL. Supporta inoltre l'utilizzo della decodifica logica. Puoi configurare gli slot di replica logica nell'istanza ed effettuare lo streaming delle modifiche del database tramite questi slot in un client come `pg_recvlogical`. Gli slot di replica logica sono creati a livello di database e supportano le connessioni di replica a un singolo database.

I client più comuni per la replica logica PostgreSQL sono AWS Database Migration Service o un host gestito personalizzato su un'istanza Amazon EC2. Lo slot di replica logica non contiene informazioni sul ricevitore del flusso. Inoltre, non è necessario che il target sia un database di replica. Se configuri uno slot di replica logica e non leggi lo slot, i dati possono venire scritti e riempire rapidamente lo storage dell'istanza database.

La replica logica e la decodifica logica PostgreSQL in Amazon RDS vengono attivate con un parametro, un tipo di connessione di replica e un ruolo di sicurezza. Il client per la decodifica logica può essere qualsiasi client in grado di stabilire una connessione di replica a un database in un'istanza database PostgreSQL.

Per attivare la decodifica logica per un'istanza database RDS for PostgreSQL

1. Assicurati che l'account utente che stai utilizzando abbia i seguenti ruoli:
   - Il ruolo `rds_superuser` in modo da poter attivare la replica logica
   - Il ruolo `rds_replication` per concedere le autorizzazioni per gestire gli slot logici e per eseguire lo streaming dei dati utilizzando gli slot logici
2. Impostare il parametro statico `rds.logical_replication` su 1. Come parte dell'applicazione di questo parametro, imposta anche i parametri `wal_level`, `max_wal_senders`, `max_replication_slots` e `max_connections`. Le modifiche di questi parametri possono aumentare la generazione dei WAL, quindi imposta il parametro `rds.logical_replication` solo quando si utilizzano gli slot logici.
3. Affinché il parametro statico `rds.logical_replication` abbia effetto, riavviare l'istanza database.
4. Creare uno slot di replica logica come spiegato nella sezione successiva. Questo processo richiede che venga specificato un plug-in di decodifica. Attualmente RDS for PostgreSQL supporta i plugin di output test_decoding e wal2json forniti con PostgreSQL.

Per ulteriori informazioni sulla decodifica logica di PostgreSQL, consulta la documentazione di PostgreSQL.

Lavorare con gli slot di replica logica

Puoi utilizzare i comandi SQL per lavorare con gli slot logici. Ad esempio, il seguente comando crea uno slot logico denominato `test_slot` che utilizza il plugin di output PostgreSQL predefinito test_decoding `test_decoding`.

```
SELECT * FROM pg_create_logical_replication_slot('test_slot', 'test_decoding');
slot_name    | xlog_position
-----------------+---------------
regression_slot | 0/16B1970
(1 row)
```

Per elencare gli slot logici, utilizza il seguente comando:

```
SELECT * FROM pg_replication_slots;
```

Per eliminare uno slot logico, utilizza il seguente comando:

```
SELECT pg_drop_replication_slot('test_slot');
pg_drop_replication_slot
-----------------------
(1 row)
```

Per ulteriori esempi su come lavorare con gli slot di replica logica, consulta gli esempi di decodifica logica nella documentazione PostgreSQL.

Una volta creato lo slot di replica logica, puoi iniziare lo streaming. Nell'esempio seguente viene illustrato il modo in cui la decodifica logica viene controllata tramite il protocollo di replica in streaming. Questo esempio utilizza il programma pg_recvlogical, incluso nella distribuzione PostgreSQL. Questa operazione richiede che l'autenticazione del client sia configurata per permettere le connessioni alla replica.

```
pg_recvlogical -d postgres --slot test_slot -U postgres
    --host -instance-name.111122223333.aws-region.rds.amazonaws.com
    -f -  --start
```

Per visualizzare il contenuto della vista `pg_replication_origin_status`, eseguire una query sulla funzione `pg_show_replication_origin_status`.

```
SELECT * FROM pg_show_replication_origin_status();
local_id | external_id | remote_lsn | local_lsn
----------+-------------+------------+-----------
(0 rows)
```

Disco RAM per stats_temp_directory

Puoi utilizzare il parametro `rds.pg_stat_ramdisk_size` di RDS for PostgreSQL per specificare la memoria di sistema allocata a un disco RAM per l'archiviazione di `stats_temp_directory` PostgreSQL. Il parametro del disco RAM è disponibile per tutte le versioni PostgreSQL su Amazon RDS.

Per alcuni carichi di lavoro, l'impostazione di questo parametro può migliorare le prestazioni e ridurre i requisiti I/O. Per ulteriori informazioni su `stats_temp_directory`, consulta la documentazione di PostgreSQL..

Per impostare un disco RAM per `stats_temp_directory`, imposta il parametro `rds.pg_stat_ramdisk_size` su un valore diverso da zero nel gruppo di parametri utilizzato dalla tua istanza database. Questo parametro utilizza MB, quindi è necessario specificare un valore intero. Espressioni, formule e funzioni non sono valide per il parametro `rds.pg_stat_ramdisk_size`. Assicurati di riavviare l'istanza database in modo da applicare il nuovo valore. Per informazioni sull'estensione dei parametri consulta Utilizzo di gruppi di parametri.

Ad esempio, il seguente comando AWS CLI imposta il parametro del disco RAM su 256 MB.

```
aws rds modify-db-parameter-group \
    --db-parameter-group-name pg-95-ramdisk-testing \
    --parameters "ParameterName=rds.pg_stat_ramdisk_size, ParameterValue=256,
 ApplyMethod=pending-reboot"
```

Dopo il riavvio, esegui il seguente comando per visualizzare lo stato di `stats_temp_directory`:

```
postgres=> SHOW stats_temp_directory;
```

Il comando deve restituire quanto segue.

```
stats_temp_directory
```

```
---------------------------
/rdsdbramdisk/pg_stat_tmp
(1 row)
```

Spazi tabelle per RDS for PostgreSQL

RDS for PostgreSQL supporta gli spazi tabelle per la compatibilità. Poiché tutto lo storage si trova su un singolo volume logico, non è possibile utilizzare gli spazi tabelle per la suddivisione o l'isolamento di I/O. I nostri benchmark e la nostra esperienza indicano che un singolo volume logico è la configurazione migliore per la maggior parte dei casi d'uso.

Per creare e utilizzare spazi tabelle con l'istanza database di RDS for PostgreSQL occorre il ruolo `rds_superuser`. L'account utente principale dell'istanza database RDS for PostgreSQL (nome di default, `postgres`) è un membro di questo ruolo. Per ulteriori informazioni, consulta Informazioni su ruoli e autorizzazioni di PostgreSQL.

Se si specifica un nome di file quando si crea uno spazio tabelle, il prefisso del percorso è `/rdsdbdata/db/base/tablespace`. Nell'esempio seguente i file dello spazio tabelle vengono posizionati in `/rdsdbdata/db/base/tablespace/data`. Questo esempio presuppone l'esistenza di un utente (ruolo) `dbadmin` e la concessione del ruolo `rds_superuser` necessario per lavorare con gli spazi tabella.

```
postgres=> CREATE TABLESPACE act_data
  OWNER dbadmin
  LOCATION '/data';
CREATE TABLESPACE
```

Per ulteriori informazioni sugli spazi tabella PostgreSQL, consulta Spazi tabella nella documentazione di PostgreSQL.

Regole di confronto RDS per PostgreSQL per EBCDIC e altre migrazioni di mainframe

RDS per PostgreSQL versione 10 e successive include la versione 60.2 di ICU, basata su Unicode 10.0 e include le regole di confronto tratte da Unicode Common Locale Data Repository, CLDR 32. Queste librerie di internazionalizzazione del software garantiscono la coerenza a livello di presentazione delle codifiche dei caratteri, indipendentemente dal sistema operativo o dalla piattaforma. Per ulteriori informazioni su Unicode CLDR-32, consulta la Nota di rilascio di CLDR 32 sul sito Web Unicode CLDR. Puoi saperne di più sui componenti di internazionalizzazione per

Unicode (ICU) nel sito Web ICU Technical Committee (ICU-TC). Per informazioni su ICU-60, consulta la pagina Download ICU 60.

A partire dalla versione 14.3, RDS per PostgreSQL include anche regole di confronto che facilitano l'integrazione e la conversione dei dati dai sistemi basati su EBCDIC. Il codice di interscambio decimale con codice binario esteso codifica EBCDIC (Extended Binary Coded Decimal Interchange Code) è comunemente utilizzata dai sistemi operativi mainframe. Queste regole di confronto fornite da Amazon RDS sono definite in modo restrittivo per ordinare solo i caratteri Unicode mappati direttamente alle tabelle codici EBCDIC. I caratteri sono ordinati in base all'ordine dei punti di codice EBCDIC per consentire la convalida dei dati dopo la conversione. Queste regole di confronto non includono i moduli denormalizzati, né i caratteri Unicode non associati direttamente a un carattere nella tabella codici EBCDIC di origine.

Le mappature dei caratteri tra le tabelle codici EBCDIC e i punti di codice Unicode si basano su tabelle pubblicate da IBM. Il set completo è disponibile presso IBM come file compresso da scaricare. RDS per PostgreSQL ha utilizzato queste mappature con gli strumenti forniti da ICU per creare le regole di confronto elencate nelle tabelle di questa sezione. I nomi delle regole di confronto includono la lingua e il paese richiesti da ICU. Tuttavia, le tabelle codici EBCDIC non specificano le lingue e alcune tabelle codici EBCDIC coprono più paesi. Ciò significa che la parte relativa alla lingua e al paese dei nomi delle regole di confronto nella tabella è arbitraria e non deve necessariamente corrispondere alla lingua corrente. In altre parole, il numero della tabella codici è la parte più importante del nome della regola di confronto in questa tabella. È possibile utilizzare una qualsiasi delle regole di confronto elencate nelle tabelle seguenti in qualsiasi database RDS per PostgreSQL.

- Unicode to EBCDIC collations table: alcuni strumenti di migrazione dei dati mainframe utilizzano internamente LATIN1 o LATIN9 per codificare ed elaborare i dati. Tali strumenti utilizzano schemi round trip per preservare l'integrità dei dati e supportare la conversione inversa. Le regole di confronto in questa tabella possono essere utilizzate da strumenti che elaborano i dati utilizzando la codifica LATIN1, che non richiede una gestione speciale.
- Unicode to LATIN9 collations table: è possibile utilizzare queste regole di confronto in qualsiasi database RDS per PostgreSQL.

Nella tabella seguente sono disponibili le regole di confronto disponibili in RDS per PostgreSQL che mappano le tabelle codici EBCDIC ai punti di codice Unicode. Si consiglia di utilizzare le regole di confronto contenute in questa tabella per lo sviluppo di applicazioni che richiedono l'ordinamento in base all'ordine delle tabelle codici IBM.

Nome della regola di confronto PostgreSQL	Descrizione della mappatura e dell'ordinamento delle tabelle codici
da-DK-cp277-x-icu	I caratteri Unicode mappati direttamente alla tabella codici IBM EBCDIC 277 (per tabelle di conversione) sono ordinati in base all'ordine dei punti di codice IBM CP 277
de-DE-cp273-x-icu	I caratteri Unicode mappati direttamente alla tabella codici IBM EBCDIC 273 (per tabelle di conversione) sono ordinati in base all'ordine dei punti di codice IBM CP 273
en-GB-cp285-x-icu	I caratteri Unicode mappati direttamente alla tabella codici IBM EBCDIC 285 (per tabelle di conversione) sono ordinati in base all'ordine dei punti di codice IBM CP 285
en-US-cp037-x-icu	I caratteri Unicode mappati direttamente alla tabella codici IBM EBCDIC 037 (per tabelle di conversione) sono ordinati in base all'ordine dei punti di codice IBM CP 37
es-ES-cp284-x-icu	I caratteri Unicode mappati direttamente alla tabella codici IBM EBCDIC 284 (per tabelle di conversione) sono ordinati in base all'ordine dei punti di codice IBM CP 284
fi-FI-cp278-x-icu	I caratteri Unicode mappati direttamente alla tabella codici IBM EBCDIC 278 (per tabelle di conversione) sono ordinati in base all'ordine dei punti di codice IBM CP 278
fr-FR-cp297-x-icu	I caratteri Unicode mappati direttamente alla tabella codici IBM EBCDIC 297 (per tabelle di conversione) sono ordinati in base all'ordine dei punti di codice IBM CP 297

Nome della regola di confronto PostgreSQL	Descrizione della mappatura e dell'ordinamento delle tabelle codici
it-IT-cp280-x-icu	I caratteri Unicode mappati direttamente alla tabella codici IBM EBCDIC 280 (per tabelle di conversione) sono ordinati in base all'ordine dei punti di codice IBM CP 280
nl-BE-cp500-x-icu	I caratteri Unicode mappati direttamente alla tabella codici IBM EBCDIC 500 (per tabelle di conversione) sono ordinati in base all'ordine dei punti di codice IBM CP 500

Amazon RDS fornisce una serie di regole di confronto aggiuntive che ordinano i punti di codice Unicode mappati ai caratteri LATIN9 utilizzando le tabelle pubblicate da IBM, nell'ordine dei punti di codice originali in base alla tabella codici EBCDIC dei dati di origine.

Nome della regola di confronto PostgreSQL	Descrizione della mappatura e dell'ordinamento delle tabelle codici
da-DK-cp1142m-x-icu	I caratteri Unicode mappati direttamente ai caratteri LATIN9 originariamente convertiti dalla tabella codici IBM EBCDIC 1142 (per tabelle di conversione) sono ordinati in base all'ordine dei punti di codice IBM CP 1142
de-DE-cp1141m-x-icu	I caratteri Unicode mappati direttamente ai caratteri LATIN9 originariamente convertiti dalla tabella codici IBM EBCDIC 1141 (per tabelle di conversione) sono ordinati in base all'ordine dei punti di codice IBM CP 1141
en-GB-cp1146m-x-icu	I caratteri Unicode mappati direttamente ai caratteri LATIN9 originariamente convertiti dalla tabella codici IBM EBCDIC 1146 (per tabelle di

Nome della regola di confronto PostgreSQL	Descrizione della mappatura e dell'ordinamento delle tabelle codici
	conversione) sono ordinati in base all'ordine dei punti di codice IBM CP 1146
en-US-cp1140m-x-icu	I caratteri Unicode mappati direttamente ai caratteri LATIN9 originariamente convertiti dalla tabella codici IBM EBCDIC 1140 (per tabelle di conversione) sono ordinati in base all'ordine dei punti di codice IBM CP 1140
es-ES-cp1145b-x-icu	I caratteri Unicode mappati direttamente ai caratteri LATIN9 originariamente convertiti dalla tabella codici IBM EBCDIC 1145 (per tabelle di conversione) sono ordinati in base all'ordine dei punti di codice IBM CP 1145
fi-FI-cp1143m-x-icu	I caratteri Unicode mappati direttamente ai caratteri LATIN9 originariamente convertiti dalla tabella codici IBM EBCDIC 1143 (per tabelle di conversione) sono ordinati in base all'ordine dei punti di codice IBM CP 1143
fr-FR-cp1147m-x-icu	I caratteri Unicode mappati direttamente ai caratteri LATIN9 originariamente convertiti dalla tabella codici IBM EBCDIC 1147 (per tabelle di conversione) sono ordinati in base all'ordine dei punti di codice IBM CP 1147
it-IT-cp1144b-x-icu	I caratteri Unicode mappati direttamente ai caratteri LATIN9 originariamente convertiti dalla tabella codici IBM EBCDIC 1144 (per tabelle di conversione) sono ordinati in base all'ordine dei punti di codice IBM CP 1144

Nome della regola di confronto PostgreSQL	Descrizione della mappatura e dell'ordinamento delle tabelle codici
nl-BE-cp1148m-x-icu	I caratteri Unicode mappati direttamente ai caratteri LATIN9 originariamente convertiti dalla tabella codici IBM EBCDIC 1148 (per tabelle di conversione) sono ordinati in base all'ordine dei punti di codice IBM CP 1148

Di seguito viene fornito un esempio di come utilizzare una regola di confronto RDS per PostgreSQL.

```
db1=> SELECT pg_import_system_collations('pg_catalog');
 pg_import_system_collations
----------------------------
                          36
db1=> SELECT '¤' < 'a' col1;
 col1
------
 t
db1=> SELECT '¤' < 'a' COLLATE "da-DK-cp277-x-icu" col1;
 col1
------
 f
```

Si consiglia di utilizzare le regole di confronto contenute nella Unicode to EBCDIC collations table e nella Unicode to LATIN9 collations table per lo sviluppo di applicazioni che richiedono l'ordinamento in base all'ordine delle tabelle codici IBM. Le seguenti regole di confronto (con il suffisso "b") sono visibili anche in `pg_collation`, ma sono destinate all'uso da parte degli strumenti di integrazione e migrazione dei dati mainframe in AWS che mappano le tabelle codici con spostamenti specifici dei punti di codice e richiedono una gestione speciale a livello di regola di confronto. In altre parole, si sconsiglia l'utilizzo delle regole di confronto elencate di seguito.

- da-DK-277b-x-icu
- da-DK-1142b-x-icu
- de-DE-cp273b-x-icu
- de-DE-cp1141b-x-icu
- en-GB-cp1146b-x-icu

- en-GB-cp285b-x-icu
- en-US-cp037b-x-icu
- en-US-cp1140b-x-icu
- es-ES-cp1145b-x-icu
- es-ES-cp284b-x-icu
- fi-FI-cp1143b-x-icu
- fr-FR-cp1147b-x-icu
- fr-FR-cp297b-x-icu
- it-IT-cp1144b-x-icu
- it-IT-cp280b-x-icu
- nl-BE-cp1148b-x-icu
- nl-BE-cp500b-x-icu

Per saperne di più sulla migrazione delle applicazioni da ambienti mainframe ad AWS, consulta la pagina relativa alla modernizzazione dei mainframe AWS.

Per ulteriori informazioni sulla gestione delle regole di confronto in PostgreSQL, consulta la pagina relativa al supporto delle regole di confronto nella documentazione di PostgreSQL.

Connessione a un'istanza database che esegua il motore di database di PostgreSQL

Dopo che Amazon RDS effettua il provisioning dell'istanza database, è possibile utilizzare qualsiasi applicazione client SQL standard per la connessione all'istanza. Prima di poterti connettere, l'istanza database deve essere disponibile e accessibile. Se è possibile o meno connettersi all'istanza dall'esterno del VPC dipende da come hai creato l'istanza database Amazon RDS:

- Se hai creato la tua istanza database come pubblica, i dispositivi e le istanze Amazon EC2 al di fuori del VPC possono connettersi al database.
- Se hai creato la tua istanza database come privata, solo i dispositivi e le istanze Amazon EC2 all'interno di Amazon VPC possono connettersi al database.

Per verificare se la tua istanza DB è pubblica o privata, usa AWS Management Console per visualizzare la scheda Connettività e sicurezza per la tua istanza. Sotto Security (Sicurezza), puoi trovare il valore "Accessibile pubblicamente", con No per privato, Sì per pubblico.

Per ulteriori informazioni sulle diverse configurazioni Amazon RDS e Amazon VPC e su come influiscono sull'accessibilità, consultare Scenari per accedere a un'istanza database in un VPC.

Indice

Installazione del client psql

Per connetterti alla tua istanza DB da un'istanza EC2, puoi installare un client PostgreSQL sull'istanza EC2. Per installare il client psql su Amazon Linux 2023, esegui il seguente comando:

```
sudo dnf install postgresql15
```

Per installare il client psql su Amazon Linux 2, esegui il seguente comando:

```
sudo amazon-linux-extras install postgresql14
```

Per installare il client psql su Ubuntu, esegui il seguente comando:

```
sudo apt-get install -y postgresql14
```

Ricerca delle informazioni di connessione per un'istanza DB RDS per PostgreSQL

Se l'istanza database è disponibile e accessibile, è possibile connettersi fornendo le seguenti informazioni all'applicazione client SQL:

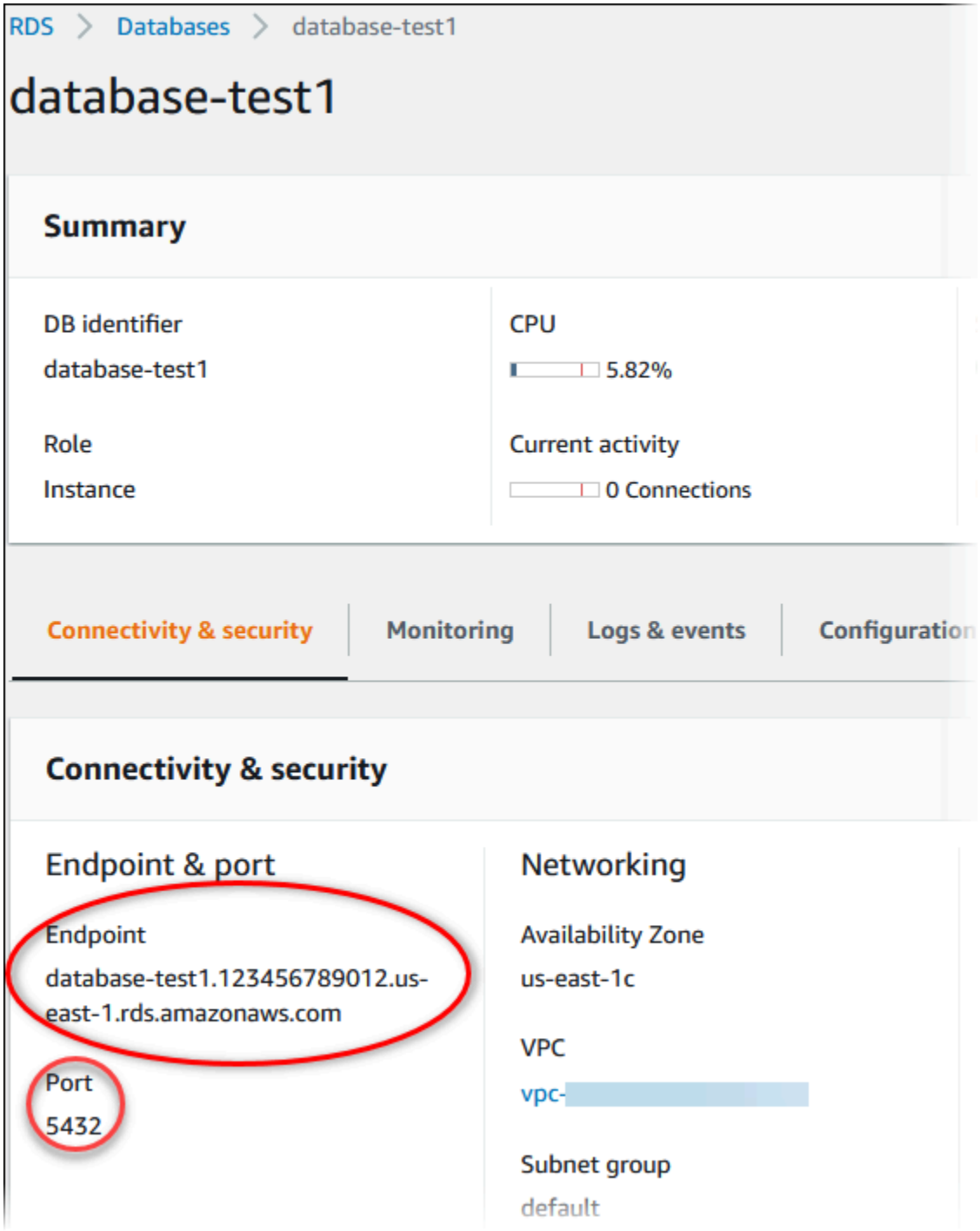
- L'endpoint dell'istanza database, che funge da nome host (nome DNS) per l'istanza.
- Porta di ascolto dell'istanza database. La porta predefinita per PostgreSQL è la 5432.
- Nome utente e password per l'istanza database. Il «nome utente principale» predefinito per PostgreSQL è `postgres`.
- Il nome e la password del database (nome DB).

Puoi ottenere questi dettagli utilizzando il AWS CLI describe-db-instancescomando AWS Management Console, o l'operazione DescribedBInstances dell'API Amazon RDS.

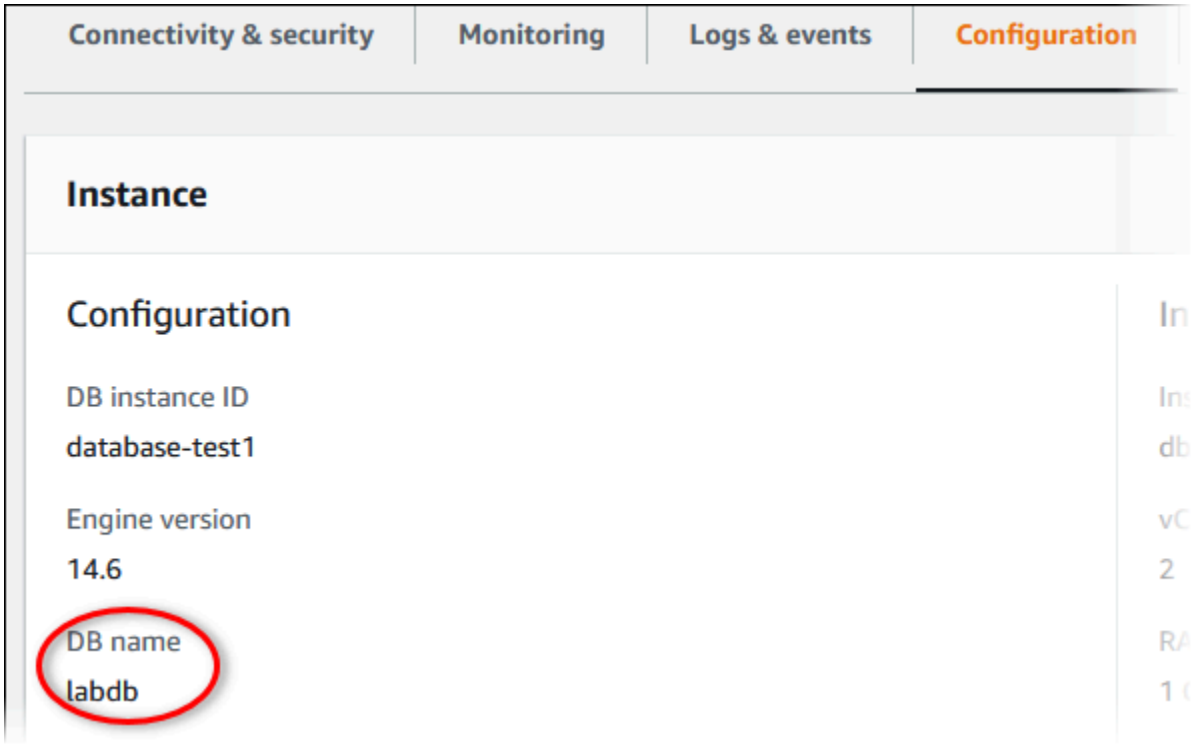
Per trovare l'endpoint, il numero di porta e il nome del DB, utilizza il AWS Management Console

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Aprire la console RDS e scegliere Databases (Database) per visualizzare un elenco delle istanze database.
3. Scegliere il nome dell'istanza database PostgreSQL per visualizzarne i dettagli.

4. Nella scheda Connectivity & security (Connettività e sicurezza), copiare l'endpoint. Annotare anche il numero di porta. L'endpoint e il numero di porta sono necessari per la connessione all'istanza database.



5. Nella scheda Configuration (Configurazione), annotare il nome del database. Se hai creato un database quando hai creato l'istanza RDS for PostgreSQL, viene visualizzato il nome elencato sotto nome DB. Se non è stato creato un database, il nome DB visualizza un trattino (-).



Di seguito vengono indicati due modi per connettersi a un'istanza database PostgreSQL. Il primo esempio utilizza pgAdmin, un popolare strumento di amministrazione e sviluppo open source per PostgreSQL. Il secondo esempio utilizza psql, una utility a riga di comando che fa parte di un'installazione di PostgreSQL.

Utilizzo di pgAdmin per connettersi a un'istanza database RDS for PostgreSQL

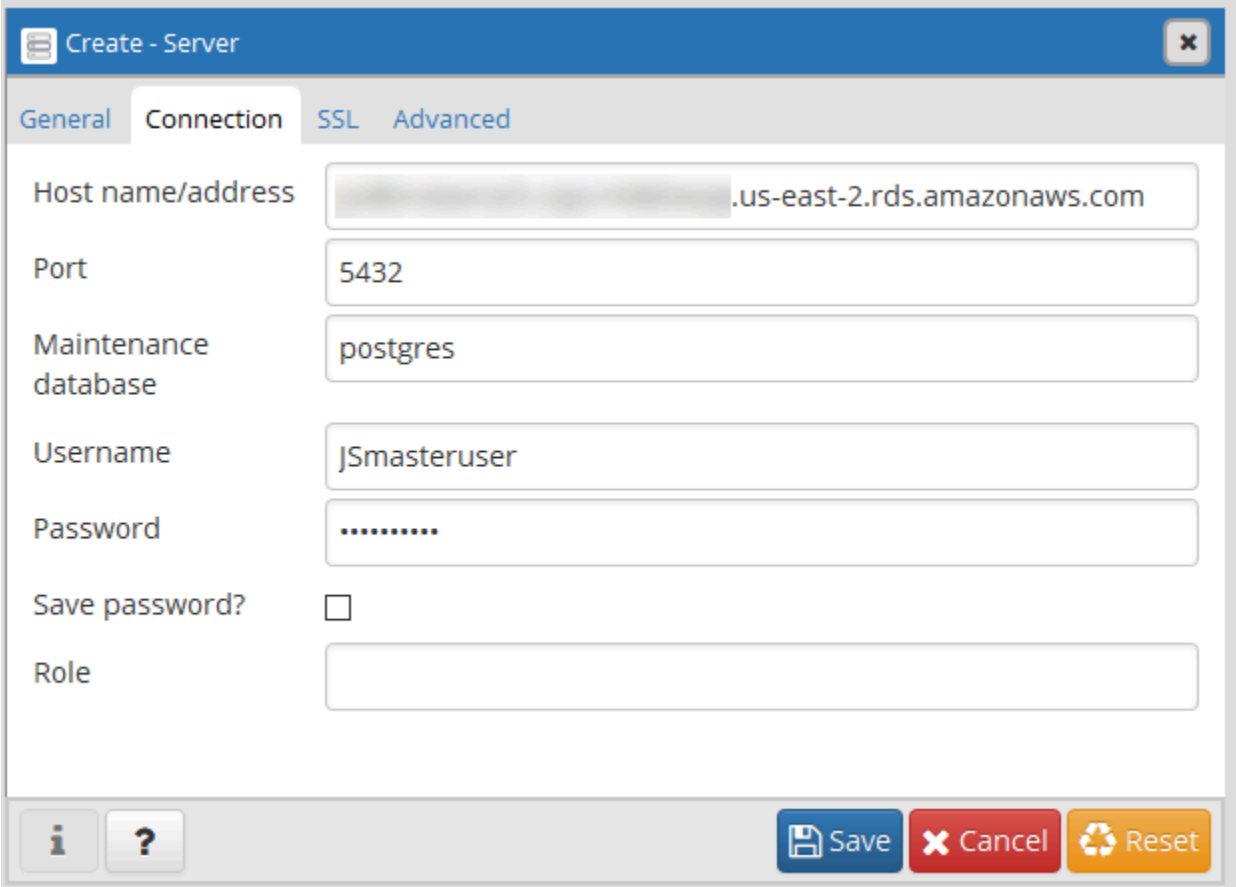
È possibile utilizzare lo strumento open source pgAdmin per connettersi a un'istanza database RDS for PostgreSQL. È possibile scaricare e utilizzare pgAdmin da http://www.pgadmin.org/ senza disporre di un'istanza locale di PostgreSQL sul computer client.

Per connettere l'istanza database RDS per PostgreSQL utilizzando pgAdmin

1. Avviare l'applicazione pgAdmin sul computer client.
2. Nella scheda Dashboard (Pannello di controllo) selezionare Add New Server (Aggiungi nuovo server).
3. Nella finestra di dialogo Create - Server (Crea - Server) digitare un nome nella scheda General (Generale) per identificare il server in pgAdmin.

4. Nella scheda Connection (Connessione) digitare le informazioni seguenti relative all'istanza database:

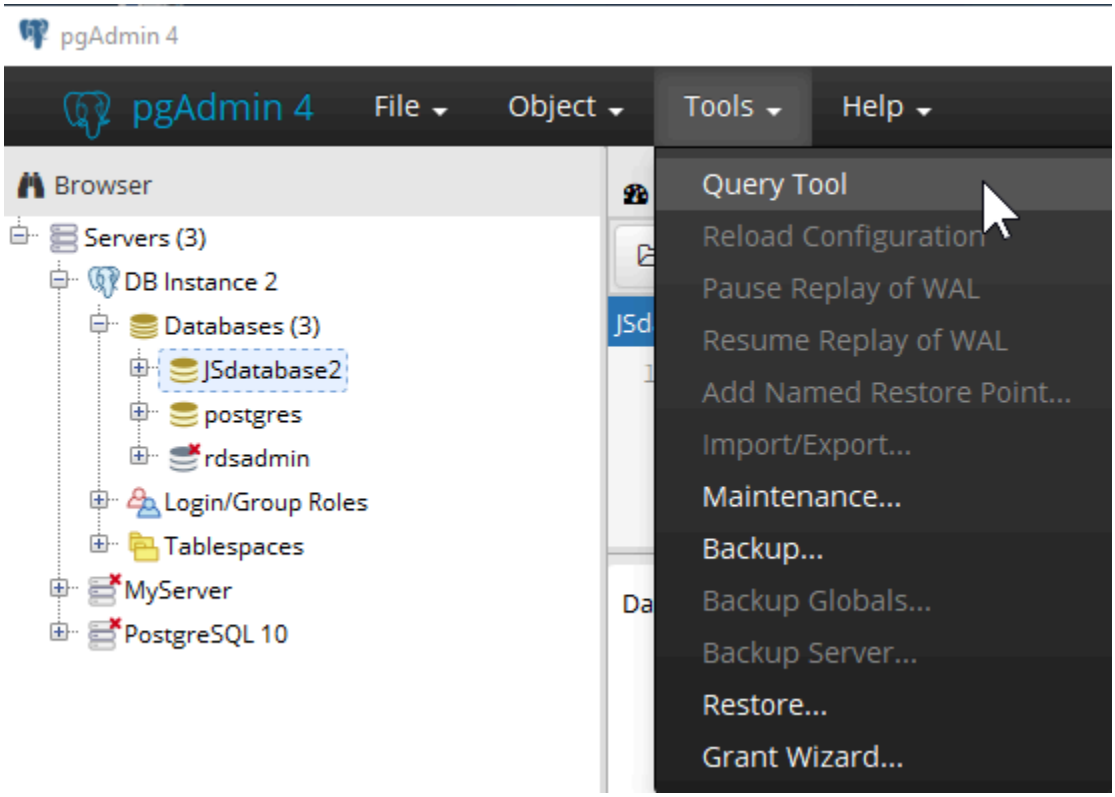
   - In Host, digitare l'endpoint, ad esempio `mypostgresql.c6c8dntfzzhgv0.us-east-2.rds.amazonaws.com`.
   - In Port (Porta) digitare la porta assegnata.
   - Per Username (Nome utente), digitare il nome utente immesso quando è stata creata l'istanza database (se è stato modificato il «nome utente master» dal valore predefinito,`postgres`).
   - In Password, digitare la password immessa quando è stata creata l'istanza database.



5. Scegliere Save (Salva).

   In caso di problemi con la connessione, consulta Risoluzione dei problemi relativi alle connessioni all'istanza RDS per PostgreSQL.

6. Per accedere a un database nel browser pgAdmin, espandere Servers (Server), l'istanza database e Databases (Database). Scegliere il nome del database dell'istanza database.



7. Per aprire un pannello dove immettere i comandi SQL, scegliere Tools (Strumenti), Query Tool (Strumento di query).

Utilizzo di psql per connettersi a un'istanza database RDS per PostgreSQL

È possibile utilizzare un'istanza locale dell'utility a riga di comando psql per connettersi a un'istanza database RDS per PostgreSQL. È necessario che PostgreSQL o il client psql sia installato sul computer client.

Puoi scaricare il client PostgreSQL dal sito web di PostgreSQL. Per installare psql, segui le istruzioni relative al tuo sistema operativo.

Per eseguire la connessione all'istanza database RDS for PostgreSQL utilizzando psql, devi fornire le informazioni host (DNS), le credenziali di accesso e il nomde del database.

Utilizza uno dei seguenti formati per eseguire la connessione a un'istanza database RDS for PostgreSQL. Quando si esegue la connessione, verrà richiesta una password. Per gli script o i processi batch, utilizzare l'opzione `--no-password`. Questa opzione è impostata per l'intera sessione.

> **Note**
>
> Un tentativo di connessione con `--no-password` non riesce quando il server richiede l'autenticazione con password e una password non è disponibile da altre origini. Per ulteriori informazioni, consulta la documentazione di psql.

Se è la prima volta che ti stai connettendo a questa istanza database o se non hai ancora creato un database per questa istanza RDS for PostgreSQL, puoi connetterti al database postgres utilizzando il «nome utente master» e la password.

Per Unix, utilizzare il seguente formato:

```
psql \
   --host=<DB instance endpoint> \
   --port=<port> \
   --username=<master username> \
   --password \
   --dbname=<database name>
```

Per Windows, utilizzare il seguente formato:

```
psql ^
   --host=<DB instance endpoint> ^
   --port=<port> ^
   --username=<master username> ^
   --password ^
   --dbname=<database name>
```

Il comando seguente esegue ad esempio la connessione a un database denominato `mypgdb` su un'istanza database PostgreSQL denominata `mypostgresql` tramite credenziali fittizie.

```
psql --host=mypostgresql.c6c8mwvfdgv0.us-west-2.rds.amazonaws.com --port=5432 --
username=awsuser --password --dbname=mypgdb
```

Connessione a RDS per PostgreSQL con il driver JDBC Amazon Web Services ()AWS

Il driver JDBC di Amazon Web Services (AWS) è progettato come wrapper JDBC avanzato. Questo wrapper è complementare e amplia le funzionalità di un driver JDBC esistente. Il driver è compatibile direttamente con il driver pgJDBC della community.

Per installare il driver AWS JDBC, aggiungi il file.jar del driver AWS JDBC (che si trova nell'applicazione) e mantieni i riferimenti al rispettivo driver della community. `CLASSPATH` Aggiorna il rispettivo prefisso dell'URL di connessione come segue:

- `jdbc:postgresql://` Da a `jdbc:aws-wrapper:postgresql://`

Per ulteriori informazioni sul driver AWS JDBC e istruzioni complete per il suo utilizzo, consulta l'archivio dei driver JDBC di Amazon Web Services (AWS). GitHub

Connessione a RDS per PostgreSQL con il driver Python di Amazon Web Services ()AWS

Il driver Python di Amazon Web Services (AWS) è progettato come wrapper Python avanzato. Questo wrapper è complementare ed estende le funzionalità del driver open source Psycopg. Il AWS Python Driver supporta le versioni Python 3.8 e successive. È possibile installare il `aws-advanced-python-wrapper` pacchetto utilizzando il `pip` comando, insieme ai pacchetti open source. `psycopg`

Per ulteriori informazioni sul driver AWS Python e istruzioni complete per il suo utilizzo, consulta il repository Amazon Web Services ()AWS Python Driver. GitHub

Risoluzione dei problemi relativi alle connessioni all'istanza RDS per PostgreSQL

Argomenti

- Errore – IRREVERSIBILE: nome database non esiste
- Errore – Impossibile connettersi al server: timeout della connessione
- Errori con regole di accesso ai gruppi di sicurezza

Errore – IRREVERSIBILE: *nome* database non esiste

Se ricevi un errore `FATAL: database name does not exist` durante il tentativo di connessione, prova a usare il nome del database predefinito postgres per l'opzione `--dbname`.

Errore – Impossibile connettersi al server: timeout della connessione

Se la connessione all'istanza database non riesce, l'errore più comunemente restituito è `Could not connect to server: Connection timed out.` Se ricevi questo errore, procedi come segue:

- Controllare che il nome host utilizzato corrisponda all'endpoint dell'istanza database e che il numero di porta utilizzato sia corretto.
- Assicurati che l'accessibilità pubblica dell'istanza database sia impostata su Sì per consentire le connessioni esterne. Per modificare l'impostazione Accesso pubblico, consulta Modifica di un'istanza database Amazon RDS.
- Assicurarsi che l'utente che si connette al database disponga dell'accesso CONNECT. La seguente query può essere usata per fornire l'accesso di connessione al database.

  ```
  GRANT CONNECT ON DATABASE database name TO username;
  ```

- Controllare che il gruppo di sicurezza assegnato all'istanza database disponga di regole che consentano l'accesso tramite firewall utilizzati dalla connessione. Se ad esempio l'istanza database è stata creata utilizzando la porta predefinita 5432, è necessario che la tua azienda disponga di regole firewall che blocchino le connessioni eseguite dai dispositivi dell'azienda tramite tale porta.

  Per correggere l'errore, dovrai modificare l'istanza database in modo da utilizzare un'altra porta. Assicurarsi inoltre che il gruppo di sicurezza associato all'istanza database consenta le connessioni alla nuova porta. Per modificare l'impostazione Porta database, consulta Modifica di un'istanza database Amazon RDS.
- Consulta anche Errori con regole di accesso ai gruppi di sicurezza.

Errori con regole di accesso ai gruppi di sicurezza

Il problema di connessione più comune è sicuramente correlato alle regole di accesso del gruppo di sicurezza assegnate all'istanza database. Se hai usato il gruppo di sicurezza predefinito quando hai creato l'istanza database, è probabile che il gruppo di sicurezza non disponga di regole di accesso che consentano di eseguire l'accesso all'istanza.

Per un corretto funzionamento della connessione, è necessario che il gruppo di sicurezza assegnato all'istanza database al momento della creazione consenta l'accesso all'istanza database. Ad esempio, se l'istanza database è stata creata in un VPC, deve disporre di un gruppo di sicurezza VPC che autorizzi le connessioni. Controlla se l'istanza database è stata creata tramite un gruppo di sicurezza che non autorizza le connessioni dal dispositivo o dall'istanza di Amazon EC2 su cui è eseguita l'applicazione.

Puoi aggiungere o modificare una regola in entrata nel gruppo di sicurezza: Per Source (Origine), se si sceglie My IP (Il mio IP), è possibile accedere all'istanza database dall'indirizzo IP rilevato nel browser. Per ulteriori informazioni, consulta Fornisci accesso alla istanza database nel VPC creando un gruppo di sicurezza.

In alternativa, se l'istanza database è stata creata al di fuori di un VPC, deve disporre di un gruppo di sicurezza del database che autorizzi tali connessioni.

Per ulteriori informazioni sui gruppi di sicurezza Amazon RDS, consulta Controllo dell'accesso con i gruppi di sicurezza.

Protezione delle connessioni a RDS for PostgreSQL con SSL/TLS

RDS for PostgreSQL supporta la crittografia Secure Socket Layer (SSL) per le istanze database PostgreSQL. Utilizzando SSL, puoi crittografare una connessione PostgreSQL tra le tue applicazioni e le tue istanze database PostgreSQL. Puoi inoltre imporre a tutte le connessioni per la tua istanza database PostgreSQL di utilizzare SSL. RDS for PostgreSQL supporta inoltre Transport Layer Security (TLS), il protocollo successore di SSL.

Per ulteriori informazioni su Amazon RDS e sulla protezione dei dati, inclusa la crittografia delle connessioni tramite SSL/TLS, consulta Protezione dei dati in Amazon RDS.

Argomenti

- Utilizzo del protocollo SSL con un'istanza database PostgreSQL
- Aggiornamento delle applicazioni per la connessione a istanze database PostgreSQL mediante nuovi certificati SSL/TLS.

Utilizzo del protocollo SSL con un'istanza database PostgreSQL

Amazon RDS supporta la crittografia SSL per le istanze database di PostgreSQL. Utilizzando SSL, puoi crittografare una connessione PostgreSQL tra le tue applicazioni e le tue istanze database PostgreSQL. Per impostazione di default, RDS for PostgreSQL utilizza e prevede che tutti i client si connettano utilizzando SSL/TLS, ma puoi anche renderlo obbligatorio. RDS per PostgreSQL supporta le versioni Transport Layer Security (TLS) 1.1, 1.2 e 1.3.

Per informazioni generali sul supporto SSL e sui database PostgreSQL, consulta l'argomento relativo al supporto SSL nella documentazione di PostgreSQL. Per informazioni sull'utilizzo della connessione SSL in JDBC, consulta l'argomento relativo alla configurazione del client nella documentazione di PostgreSQL.

Il supporto SSL è disponibile in tutte le AWS regioni per PostgreSQL. Amazon RDS crea un certificato SSL per l'istanza database PostgreSQL al momento della creazione dell'istanza. Se abiliti la verifica del certificato SSL, il certificato SSL include l'endpoint dell'istanza database come nome comune (CN) per il certificato SSL per la protezione contro attacchi di spoofing.

Argomenti

- Connessione a un'istanza database PostgreSQL tramite SSL
- Richiesta di una connessione SSL a un'istanza database PostgreSQL

- Determinazione dello stato di connessione SSL
- Suite di crittografie SSL in RDS per PostgreSQL

Connessione a un'istanza database PostgreSQL tramite SSL

Per effettuare la connessione a un'istanza database PostgreSQL tramite SSL

1. Scaricare il certificato.

   Per ulteriori informazioni sul download dei certificati, consultare .

2. Effettuare la connessione a un'istanza database PostgreSQL tramite SSL.

   Quando ti connetti tramite SSL, il tuo client può scegliere di verificare o meno la catena di certificati. Se i parametri di connessione specificano `sslmode=verify-ca` o `sslmode=verify-full`, il client richiede che i certificati CA RDS siano nell'archivio attendibilità o facciano riferimento all'URL della connessione. Questo requisito è verificare la catena di certificati che firma il certificato del database.

   Quando un client, come psql o JDBC, è configurato con il supporto SSL, per impostazione predefinita tenta innanzitutto di connettersi al database con SSL. Se il client non riesce a connettersi con SSL, torna a connettersi senza SSL. La modalità predefinita `sslmode` utilizzata per i client basati su libpq (come psql) è diversa da quella per JDBC. I client basati su libpq utilizzano `prefer` per impostazione predefinita, mentre i client JDBC utilizzano `verify-full` per impostazione predefinita.

   Utilizzare il parametro `sslrootcert` come riferimento per il certificato, ad esempio `sslrootcert=rds-ssl-ca-cert.pem`.

Di seguito è riportato un esempio di utilizzo di `psql` per connettersi a un'istanza database PostgreSQL tramite SSL con verifica del certificato.

```
$ psql "host=db-name.555555555555.ap-southeast-1.rds.amazonaws.com
    port=5432 dbname=testDB user=testuser sslrootcert=rds-ca-rsa2048-g1.pem
 sslmode=verify-full"
```

Richiesta di una connessione SSL a un'istanza database PostgreSQL

Puoi richiedere che le connessioni alla tua istanza database PostgreSQL utilizzino SSL adoperando il parametro `rds.force_ssl`. Il parametro predefinito `rds.force_ssl` è impostato su 1 (on) per RDS per PostgreSQL versione 15. Tutti gli altri database RDS per PostgreSQL versione principale 14 e precedenti hanno il valore predefinito del parametro `rds.force_ssl` impostato su 0 (off). Puoi impostare il parametro `rds.force_ssl` su 1 (on) per richiedere la crittografia SSL per le connessioni alla tua istanza database.

Per modificare il valore di questo parametro, devi creare un gruppo parametri del database personalizzato. Quindi modifica il valore di `rds.force_ssl` nel gruppo parametri del database personalizzato su 1 per attivare questa caratteristica. Se prepari il gruppo parametri del database personalizzato prima di creare l'istanza database RDS for PostgreSQL, puoi sceglierlo (invece del gruppo di parametri di default) durante il processo di creazione. Se esegui questa operazione quando l'istanza database RDS for PostgreSQL è in esecuzione, devi riavviare l'istanza in modo che utilizzi il gruppo di parametri personalizzati. Per ulteriori informazioni, consulta Utilizzo di gruppi di parametri.

Quando la caratteristica `rds.force_ssl` è attiva sull'istanza database, i tentativi di connessione che non utilizzano SSL vengono rifiutati con il seguente messaggio:

```
$ psql -h db-name.555555555555.ap-southeast-1.rds.amazonaws.com port=5432 dbname=testDB
 user=testuser
psql: error: FATAL: no pg_hba.conf entry for host "w.x.y.z", user "testuser", database
 "testDB", SSL off
```

Determinazione dello stato di connessione SSL

Lo stato crittografato della tua connessione è indicato nel banner di accesso quando ti connetti all'istanza database:

```
Password for user master:
psql (10.3)
SSL connection (cipher: DHE-RSA-AES256-SHA, bits: 256)
Type "help" for help.
postgres=>
```

Puoi anche caricare l'estensione `sslinfo` e quindi richiamare la funzione `ssl_is_used()` per determinare se viene utilizzata la crittografia SSL. La funzione restituisce `t` se la connessione utilizza la crittografia SSL, altrimenti restituisce `f`.

```
postgres=> CREATE EXTENSION sslinfo;
CREATE EXTENSION
postgres=> SELECT ssl_is_used();
ssl_is_used
---------
t
(1 row)
```

Per informazioni dettagliate, puoi utilizzare la seguente query per ottenere informazioni da `pg_settings`:

```
SELECT name as "Parameter name", setting as value, short_desc FROM pg_settings WHERE
 name LIKE '%ssl%';
             Parameter name             |                  value                  |
                  short_desc
----------------------------------------+-----------------------------------------
+-----------------------------------------------------
 ssl                                    | on                                      |
 Enables SSL connections.
 ssl_ca_file                            | /rdsdbdata/rds-metadata/ca-cert.pem     |
 Location of the SSL certificate authority file.
 ssl_cert_file                          | /rdsdbdata/rds-metadata/server-cert.pem |
 Location of the SSL server certificate file.
 ssl_ciphers                            | HIGH:!aNULL:!3DES                       |
 Sets the list of allowed SSL ciphers.
 ssl_crl_file                           |                                         |
 Location of the SSL certificate revocation list file.
 ssl_dh_params_file                     |                                         |
 Location of the SSL DH parameters file.
 ssl_ecdh_curve                         | prime256v1                              |
 Sets the curve to use for ECDH.
 ssl_key_file                           | /rdsdbdata/rds-metadata/server-key.pem  |
 Location of the SSL server private key file.
 ssl_library                            | OpenSSL                                 |
 Name of the SSL library.
 ssl_max_protocol_version               |                                         |
 Sets the maximum SSL/TLS protocol version to use.
 ssl_min_protocol_version               | TLSv1.2                                 |
 Sets the minimum SSL/TLS protocol version to use.
 ssl_passphrase_command                 |                                         |
 Command to obtain passphrases for SSL.
 ssl_passphrase_command_supports_reload | off                                     |
 Also use ssl_passphrase_command during server reload.
```

```
 ssl_prefer_server_ciphers              | on                                      |
 Give priority to server ciphersuite order.
(14 rows)
```

Puoi anche raccogliere tutte le informazioni sull'utilizzo SSL dell'istanza database RDS for PostgreSQL per processo, client e applicazione utilizzando la query seguente:

```
SELECT datname as "Database name", usename as "User name", ssl, client_addr,
 application_name, backend_type
   FROM pg_stat_ssl
   JOIN pg_stat_activity
   ON pg_stat_ssl.pid = pg_stat_activity.pid
   ORDER BY ssl;
 Database name | User name | ssl |  client_addr   |    application_name    |
 backend_type
---------------+-----------+-----+----------------+------------------------
+------------------------------
               |           | f   |                |                        | autovacuum
 launcher
               | rdsadmin  | f   |                |                        | logical
 replication launcher
               |           | f   |                |                        | background
 writer
               |           | f   |                |                        |
 checkpointer
               |           | f   |                |                        | walwriter
 rdsadmin      | rdsadmin  | t   | 127.0.0.1      |                        | client
 backend
 rdsadmin      | rdsadmin  | t   | 127.0.0.1      | PostgreSQL JDBC Driver | client
 backend
 postgres      | postgres  | t   | 204.246.162.36 | psql                   | client
 backend
(8 rows)
```

Per identificare la cifra utilizzata per la tua connessione SSL, puoi eseguire la seguente query:

```
postgres=> SELECT ssl_cipher();
ssl_cipher
--------------------
DHE-RSA-AES256-SHA
(1 row)
```

Per ulteriori informazioni sull'opzione `sslmode`, consulta Database connection control functions nella documentazione di PostgreSQL.

Suite di crittografie SSL in RDS per PostgreSQL

Il parametro di configurazione PostgreSQL ssl_ciphers specifica le categorie di suite di crittografie consentite per le connessioni SSL. Nella tabella seguente sono elencate le suite di crittografie predefinite utilizzate in RDS per PostgreSQL.

Versione del motore PostgreSQL	Suite di cifratura
16	`HIGH:!aNULL:!3DES`
15	`HIGH:!aNULL:!3DES`
14	`HIGH:!aNULL:!3DES`
13	`HIGH:!aNULL:!3DES`
12	`HIGH:!aNULL:!3DES`
Versioni secondarie 11.4 e successive	`HIGH:MEDIUM:+3DES:!aNULL:!RC4`
11.1, 11.2	`HIGH:MEDIUM:+3DES:!aNULL`
Versioni secondarie 10.9 e successive	`HIGH:MEDIUM:+3DES:!aNULL:!RC4`
Versioni secondarie 10.7 e precedenti	`HIGH:MEDIUM:+3DES:!aNULL`

Aggiornamento delle applicazioni per la connessione a istanze database PostgreSQL mediante nuovi certificati SSL/TLS.

I certificati utilizzati per Secure Socket Layer o Transport Layer Security (SSL/TLS) hanno in genere una durata prestabilita. Quando i provider di servizi aggiornano i certificati dell'autorità di certificazione (CA), i client devono aggiornare le loro applicazioni per utilizzare i nuovi certificati. Di seguito, puoi trovare informazioni su come determinare se le applicazioni client utilizzano SSL/TLS per connettersi all'istanza database Amazon RDS for PostgreSQL. Troverai inoltre informazioni su come controllare se tali applicazioni verificano il certificato del server al momento della connessione.

> **Note**
>
> Un'applicazione client configurata per verificare il certificato del server prima della connessione SSL/TLS deve disporre di un certificato CA valido nell'archivio trust del client. Aggiorna l'archivio trust del client quando necessario per i nuovi certificati.

Dopo aver aggiornato i certificati CA negli archivi di trust delle applicazioni client, puoi ruotare i certificati nelle istanze database. Consigliamo vivamente di testare queste procedure in un ambiente non di produzione prima di implementarle negli ambienti di produzione.

Per ulteriori informazioni sulla rotazione dei certificati, consulta Rotazione del certificato SSL/TLS. Per ulteriori informazioni sul download, consulta . Per informazioni sull'utilizzo di SSL/TLS con le istanze database PostgreSQL, consulta Utilizzo del protocollo SSL con un'istanza database PostgreSQL.

Argomenti

- Determinare se un'applicazione si connette alle istanze DB PostgreSQL utilizzando SSL
- Determinare se un client richiede la verifica del certificato per la connessione
- Aggiornare l'archivio di trust delle applicazioni
- Utilizzo delle connessioni SSL/TLS per diversi tipi di applicazioni

Determinare se un'applicazione si connette alle istanze DB PostgreSQL utilizzando SSL

Verifica la configurazione delle istanze database per il valore del parametro `rds.force_ssl`. Per impostazione predefinita, il parametro `rds.force_ssl` è impostato su `0` (disattivato) per le istanze DB che utilizzano versioni di PostgreSQL precedenti alla versione 15. Per impostazione predefinita, il parametro `rds.force_ssl` è impostato su 1 (attivato) per le istanze DB che utilizzano PostgreSQL versione 15 e versioni principali successive. Se il parametro `rds.force_ssl` è impostato su 1 (attivato), i client devono utilizzare SSL/TLS per le connessioni. Per ulteriori informazioni sui gruppi di parametri, consultare Utilizzo di gruppi di parametri.

Se stai utilizzando la versione PostgreSQL RDS 9.5 o successiva `erds.force_ssl` non è impostato su 1 (attivato), esegui la query di visualizzazione `pg_stat_ssl` per verificare le connessioni che usano SSL. Ad esempio, la query seguente restituisce solo le connessioni SSL e le informazioni sui client che utilizzano SSL.

```
SELECT datname, usename, ssl, client_addr
  FROM pg_stat_ssl INNER JOIN pg_stat_activity ON pg_stat_ssl.pid =
 pg_stat_activity.pid
  WHERE ssl is true and usename<>'rdsadmin';
```

Vengono visualizzate solo le righe che utilizzano connessioni SSL/TLS con informazioni sulla connessione. Di seguito è riportato un output di esempio.

```
 datname  | usename | ssl | client_addr
----------+---------+-----+-------------
 benchdb  | pgadmin | t   | 53.95.6.13
 postgres | pgadmin | t   | 53.95.6.13
(2 rows)
```

La query precedente visualizza solo le connessioni in uso al momento della query. L'assenza di risultati non indica che nessuna applicazione stia utilizzando connessioni SSL. Altre connessioni SSL potrebbero essere stabilite in un momento diverso.

Determinare se un client richiede la verifica del certificato per la connessione

Quando un client, come psql o JDBC, è configurato con il supporto SSL, per impostazione predefinita tenta innanzitutto di connettersi al database con SSL. Se il client non riesce a connettersi con SSL, torna a connettersi senza SSL. La modalità predefinita `sslmode` utilizzata per i client basati su libpq (come psql) è diversa da quella per JDBC. I client basati su libpq utilizzano `prefer` per impostazione predefinita, mentre i client JDBC utilizzano `verify-full` per impostazione predefinita. Il certificato sul server viene verificato solo quando `sslrootcert` viene fornito con `sslmode` set to `verify-ca` o`verify-full`. Se il certificato non è valido viene generato un errore.

`PGSSLROOTCERT`Da utilizzare per verificare il certificato con la variabile di `PGSSLMODE` ambiente, `PGSSLMODE` impostata su `verify-ca` o`verify-full`.

```
PGSSLMODE=verify-full PGSSLROOTCERT=/fullpath/ssl-cert.pem psql -h
 pgdbidentifier.cxxxxxxxx.us-east-2.rds.amazonaws.com -U masteruser -d postgres
```

Utilizza l'`sslrootcert`argomento per verificare il certificato `sslmode` nel formato della stringa di connessione, con `sslmode` impostato su `verify-ca` o `verify-full` per verificare il certificato.

```
psql "host=pgdbidentifier.cxxxxxxxx.us-east-2.rds.amazonaws.com sslmode=verify-full
 sslrootcert=/full/path/ssl-cert.pem user=masteruser dbname=postgres"
```

Ad esempio, nel caso precedente, se stai utilizzando un certificato root non valido, sul client vedrai il seguente errore.

```
psql: SSL error: certificate verify failed
```

Aggiornare l'archivio di trust delle applicazioni

Per informazioni sull'aggiornamento dell'archivio attendibilità per le applicazioni PostgreSQL, consulta l'argomento relativo alle connessioni TCP/IP sicure con SSL nella documentazione di PostgreSQL.

Per ulteriori informazioni sul download del certificato root, consulta .

Per gli script di esempio che importano i certificati, consulta Script di esempio per l'importazione di certificati nel tuo archivio di trust.

> **Note**
>
> Quando aggiorni l'archivio di trust puoi conservare i certificati meno recenti oltre ad aggiungere i nuovi certificati.

Utilizzo delle connessioni SSL/TLS per diversi tipi di applicazioni

Di seguito vengono fornite le informazioni sull'utilizzo delle connessioni SSL/TLS per diversi tipi di applicazioni.

- psql

  Il client viene invocato dalla riga di comando specificando le opzioni come stringa di connessione o variabili di ambiente. Le opzioni rilevanti per le connessioni SSL/TLS sono `sslmode` (variabile di ambiente `PGSSLMODE`), `sslrootcert` (variabile di ambiente `PGSSLROOTCERT`).

  Per l'elenco completo delle opzioni, consulta l'argomento relativo alle parole chiave dei parametri nella documentazione di PostgreSQL. Per l'elenco completo delle variabili di ambiente, consulta l'argomento relativo alle variabili di ambiente nella documentazione di PostgreSQL.

- pgAdmin

  Questo client basato su browser fornisce un'interfaccia più intuitiva per la connessione a un database PostgreSQL.

Per informazioni sulla configurazione delle connessioni, consulta la documentazione di pgAdmin.

- JDBC

  JDBC abilita le connessioni al database con le applicazioni Java.

  Per informazioni generali sulla connessione a un database PostgreSQL con JDBC, consulta Connecting to the database (Connessione al database) nella documentazione del driver PostgreSQL JDBC. Per informazioni sulla connessione con SSL/TLS, consulta Configuring the client (Configurazione del client) nella documentazione del driver PostgreSQL JDBC.

- Python

  Una popolare libreria Python per la connessione ai database PostgreSQL è `psycopg2`.

  Per informazioni sull'utilizzo di `psycopg2`, consulta la documentazione psycopg2. Per un breve tutorial su come connettersi a un database PostgreSQL, consulta il tutorial di Psycopg2. Le informazioni sulle opzioni accettate dal comando di connessione sono disponibili nell'argomento relativo al contenuto del modulo Psycopg2.

> **Important**
>
> Dopo aver stabilito che le connessioni al database utilizzano SSL/TLS e aver aggiornato l'archivio attendibile dell'applicazione, è possibile aggiornare il database per utilizzare i certificati 2048-g1. rds-ca-rsa Per istruzioni, consulta la fase 3 in Aggiornamento del certificato CA modificando l'istanza o il cluster di database.

Utilizzo di Autenticazione Kerberos con Amazon RDS for PostgreSQL

Puoi utilizzare Kerberos per autenticare gli utenti quando si connettono all'istanza database che esegue PostgreSQL. A tale scopo, configura l'istanza database in modo da utilizzare AWS Directory Service for Microsoft Active Directory per l'autenticazione Kerberos. AWS Directory Service for Microsoft Active Directory è anche chiamato AWS Managed Microsoft AD. È una funzionalità disponibile con AWS Directory Service. Per ulteriori informazioni, consultare Che cos'è AWS Directory Service?nella Guida di amministrazione di AWS Directory Service.

Per iniziare, crea una directory AWS Managed Microsoft AD in cui archiviare le credenziali utente. Per l'istanza database PostgreSQL, specifica quindi il dominio di Active Directory e altre informazioni. Quando gli utenti eseguono l'autenticazione con l'istanza database PostgreSQL, le richieste di autenticazione vengono inoltrate alla directory AWS Managed Microsoft AD.

Mantenere tutte le credenziali nella stessa directory consente di ridurre il tempo e l'impegno. È disponibile una posizione centralizzata per archiviare e gestire le credenziali per più istanze database. L'uso di una directory può inoltre migliorare il profilo di sicurezza complessivo.

Puoi inoltre accedere alle credenziali da Microsoft Active Directory on-premise. A tale scopo, crea una relazione di dominio trusting in modo che la directory AWS Managed Microsoft AD consideri attendibile Microsoft Active Directory on-premise. In questo modo, gli utenti possono accedere alle istanze PostgreSQL con la stessa esperienza SSO (Single Sign-On) Windows dei carichi di lavoro nella rete locale.

Un database può utilizzare l'autenticazione con password o l'autenticazione con password con l'autenticazione Kerberos o AWS Identity and Access Management (IAM). Per ulteriori informazioni sull'autenticazione IAM, consulta Autenticazione del database IAM per MariaDB, MySQL e PostgreSQL.

Argomenti

Disponibilità di regioni e versioni

Il supporto varia a seconda delle versioni specifiche di ciascun motore di database e a seconda delle Regioni AWS. Per ulteriori informazioni sulla disponibilità della versione e della regione di RDS per PostgreSQL con autenticazione Kerberos, consulta Regioni e motori DB supportati per l'autenticazione Kerberos in Amazon RDS.

Panoramica di Autenticazione Kerberos per istanze database di PostgreSQL

Per configurare l'autenticazione Kerberos per un'istanza database di PostgreSQL, segui queste fasi, descritte dettagliatamente più avanti:

1. Utilizza AWS Managed Microsoft AD per creare una directory AWS Managed Microsoft AD. Puoi utilizzare la AWS Management Console, AWS CLI o l'API AWS Directory Service per creare la directory. Assicurati di aprire le porte in uscita rilevanti nel gruppo di sicurezza della directory in modo che la directory possa comunicare con l'istanzadel .
2. Crea un ruolo che fornisca l'accesso Amazon RDS per effettuare chiamate alla directory AWS Managed Microsoft AD. Per far ciò, crea un ruolo AWS Identity and Access Management (IAM) che utilizza la policy IAM gestita `AmazonRDSDirectoryServiceAccess`.

   Affinché il ruolo IAM possa permettere l'accesso, l'endpoint AWS Security Token Service (AWS STS) deve essere attivato nella regione AWS corretta per l'account AWS. Gli endpoint AWS STS sono attivi per impostazione predefinita in tutte le Regioni AWS e possono essere utilizzati senza ulteriori interventi. Per ulteriori informazioni, consulta Attivazione e disattivazione di AWS STS in una regione AWS nella Guida per l'utente di IAM.
3. Crea e configura utenti nella directory AWS Managed Microsoft AD utilizzando gli strumenti di Microsoft Active Directory. Per ulteriori informazioni sulla creazione di utenti Microsoft Active Directory, consulta Gestione di utenti e gruppi in AWS Managed Microsoft AD nella Guida all'amministrazione di AWS Directory Service.
4. Se si prevede di salvare la directory e l'istanza database in account AWS o in cloud privati virtuali (VPC, Virtual Private Cloud) differenti, configurare il peering di VPC. Per ulteriori informazioni, consulta Che cos'è il peering di VPC? nella Amazon VPC Peering Guide.
5. Creare o modificare un'istanza database di PostgreSQL dalla console, da CLI o dall'API di RDS utilizzando uno dei seguenti metodi:
   - Creazione di un'istanza database Amazon RDS

- Modifica di un'istanza database Amazon RDS
- Ripristino da uno snapshot database
- Ripristino a un'ora specifica per un'istanza database

Puoi individuare l'istanza nello stesso Amazon Virtual Private Cloud (VPC) della directory o in un VPC o account AWS diverso. Quando crei o modifichi l'istanza database PostgreSQL, completa le seguenti operazioni:

- Specifica l'identificativo del dominio (identificativo `d-*`) generato al momento della creazione della directory.
- Specifica anche il nome del ruolo IAM creato.
- Assicurati che il gruppo di sicurezza dell'istanza database possa ricevere traffico in entrata dal gruppo di sicurezza della directory.

6. Utilizzare le credenziali dell'utente master RDS per connettersi all'istanza database di PostgreSQL. Creare l'utente in PostgreSQL per l'identificazione esterna. Gli utenti identificati esternamente possono accedere all'istanza database di PostgreSQL con l'autenticazione Kerberos.

Configurazione dell'autenticazione Kerberos per istanze database di PostgreSQL

Per configurare l'autenticazione Kerberos, completa la procedura seguente.

Argomenti

- Passaggio 1: creare una directory utilizzando AWS Managed Microsoft AD
- Passaggio 2: (Facoltativo) Creare una relazione di fiducia tra Active Directory locale e AWS Directory Service
- Fase 3: creare un ruolo IAM per RDS per accedere a AWS Directory Service
- Fase 4: creazione e configurazione di utenti
- Fase 5: abilitazione del traffico tra VPC tra la directory e l'istanza database
- Fase 6: creazione o modifica di un'istanza database PostgreSQL
- Fase 7: creazione di utenti PostgreSQL per i principali Kerberos
- Fase 8: configurazione di un client PostgreSQL

Passaggio 1: creare una directory utilizzando AWS Managed Microsoft AD

AWS Directory Service crea una Active Directory completamente gestita nel AWS cloud. Quando crei una AWS Managed Microsoft AD directory, AWS Directory Service crea due controller di dominio e server DNS per te. I server di directory vengono creati in sottoreti diverse in un VPC. Questa ridondanza assicura che la directory rimanga accessibile anche se si verifica un errore.

Quando si crea una AWS Managed Microsoft AD AWS directory, Directory Service esegue le seguenti attività per conto dell'utente:

- Configura una Active Directory all'interno del VPC.
- Crea un account amministratore della directory con il nome utente `Admin` e la password specificata. Puoi utilizzare questo account per gestire le directory.

  > **Important**
  >
  > Assicurati di salvare questa password. AWS Directory Service non memorizza questa password e non può essere recuperata o reimpostata.

- Crea un gruppo di sicurezza per i controller della directory. Il gruppo di sicurezza deve consentire la comunicazione con l'istanza database PostgreSQL.

All'avvio AWS Directory Service for Microsoft Active Directory, AWS crea un'unità organizzativa (OU) che contiene tutti gli oggetti della directory. Questa unità organizzativa, che ha lo stesso nome NetBIOS che hai immesso al momento della creazione della directory, si trova nella radice del dominio. La radice del dominio è di proprietà e gestita da AWS.

L'`Admin`account creato con la AWS Managed Microsoft AD directory dispone delle autorizzazioni per le attività amministrative più comuni dell'unità organizzativa:

- Creazione, aggiornamento o eliminazione di utenti
- Aggiungi risorse al dominio, come file server o server di stampa, e assegna le autorizzazioni per tali risorse a utenti dell'unità organizzativa
- Creazione di unità organizzative e container aggiuntivi
- Delega dell'autorità
- Ripristino degli oggetti eliminati dal cestino di Active Directory

- Esegui i moduli Active Directory e Domain Name Service (DNS) per Windows PowerShell sul servizio Web Active Directory

L'account `Admin` dispone anche dei diritti per eseguire queste attività in tutto il dominio:

- gestione delle configurazioni DNS (aggiunta, eliminazione o aggiornamento di record, zone e server d'inoltro);
- visualizzazione di log di eventi DNS;
- visualizzazione di log di eventi di sicurezza.

Per creare una directory con AWS Managed Microsoft AD

1. Nel riquadro di navigazione della console AWS Directory Service, scegliere Directory, quindi selezionare Configurazione della directory.
2. Seleziona AWS Managed Microsoft AD. AWS Managed Microsoft AD è la sola opzione supportata per l'uso con Amazon RDS.
3. Seleziona Successivo.
4. Nella pagina Enter directory information (Inserisci le informazioni sulla directory) inserisci le seguenti informazioni:

   Edizione

   Scegliere l'edizione più adatta alle proprie esigenze.

   Nome DNS directory

   Il nome completo della directory, ad esempio **`corp.example.com`**.

   Nome NetBIOS della directory

   Nome breve opzionale della directory, ad esempio `CORP`.

   Descrizione della directory

   Descrizione opzionale della directory.

   Password amministratore

   La password dell'amministratore della directory. Con il processo di creazione della directory viene generato un account amministratore con nome utente `Admin` e questa password.

La password dell'amministratore della directory non può includere il termine "admin". La password distingue tra maiuscole e minuscole e la lunghezza deve essere compresa tra 8 e 64 caratteri. Deve anche contenere un carattere di almeno tre delle seguenti quattro categorie:

- Lettere minuscole (a–z)
- Lettere maiuscole (A–Z)
- Numeri (0–9)
- Caratteri non alfanumerici (~!@#$%^&*_-+=`|\(){}[]:;"'<>,.?/)

Confirm password (Conferma password)

Digitare di nuovo la password dell'amministratore.

> **Important**
>
> Assicurati di salvare questa password. AWS Directory Service non memorizza questa password e non può essere recuperata o reimpostata.

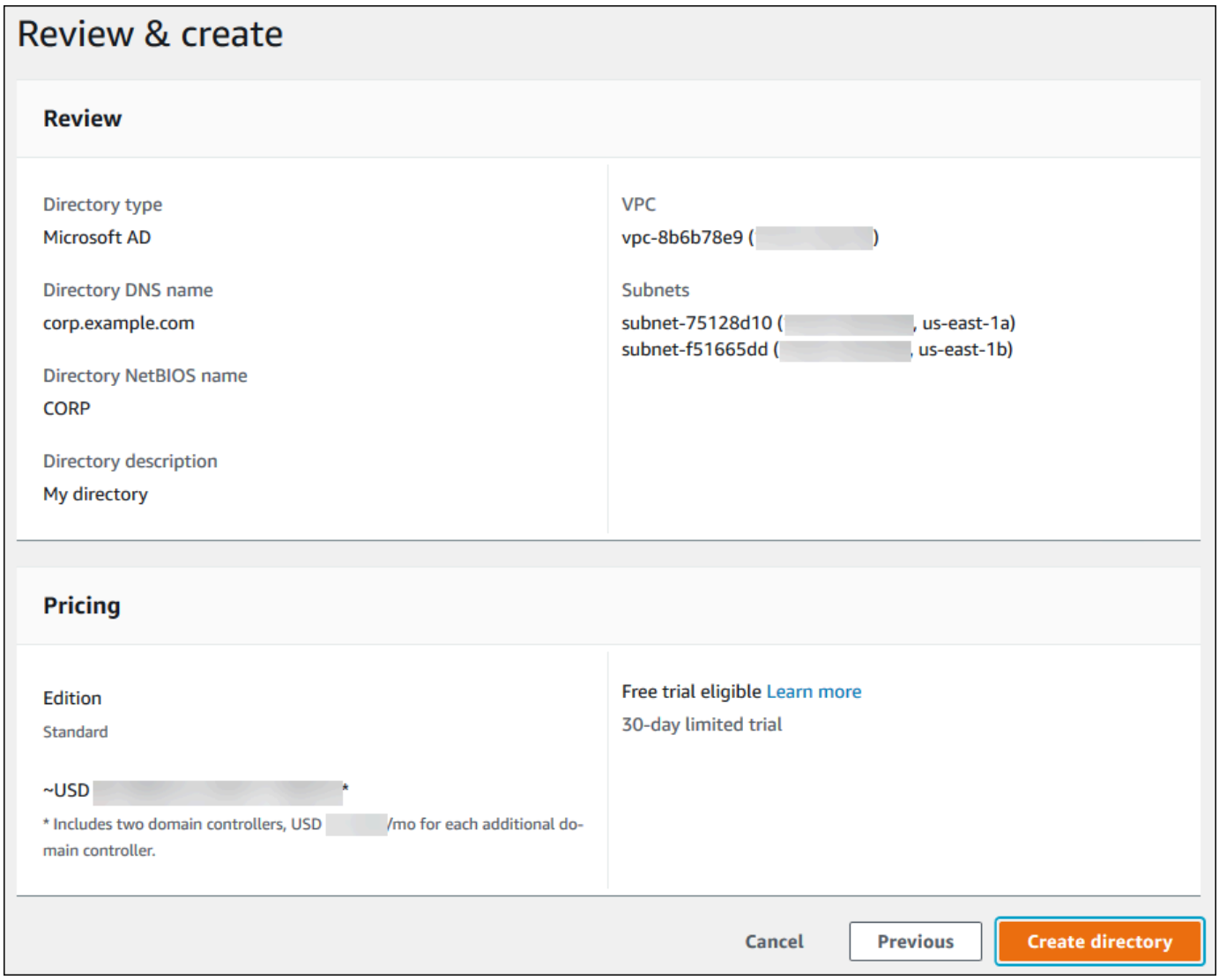
5. Seleziona Successivo.
6. Nella pagina Choose VPC and subnets (Scegli VPC e sottoreti) fornire le seguenti informazioni:

   VPC

   Scegliere il VPC per la directory. È possibile creare l'istanza database di PostgreSQL in questo VPC o in uno diverso.

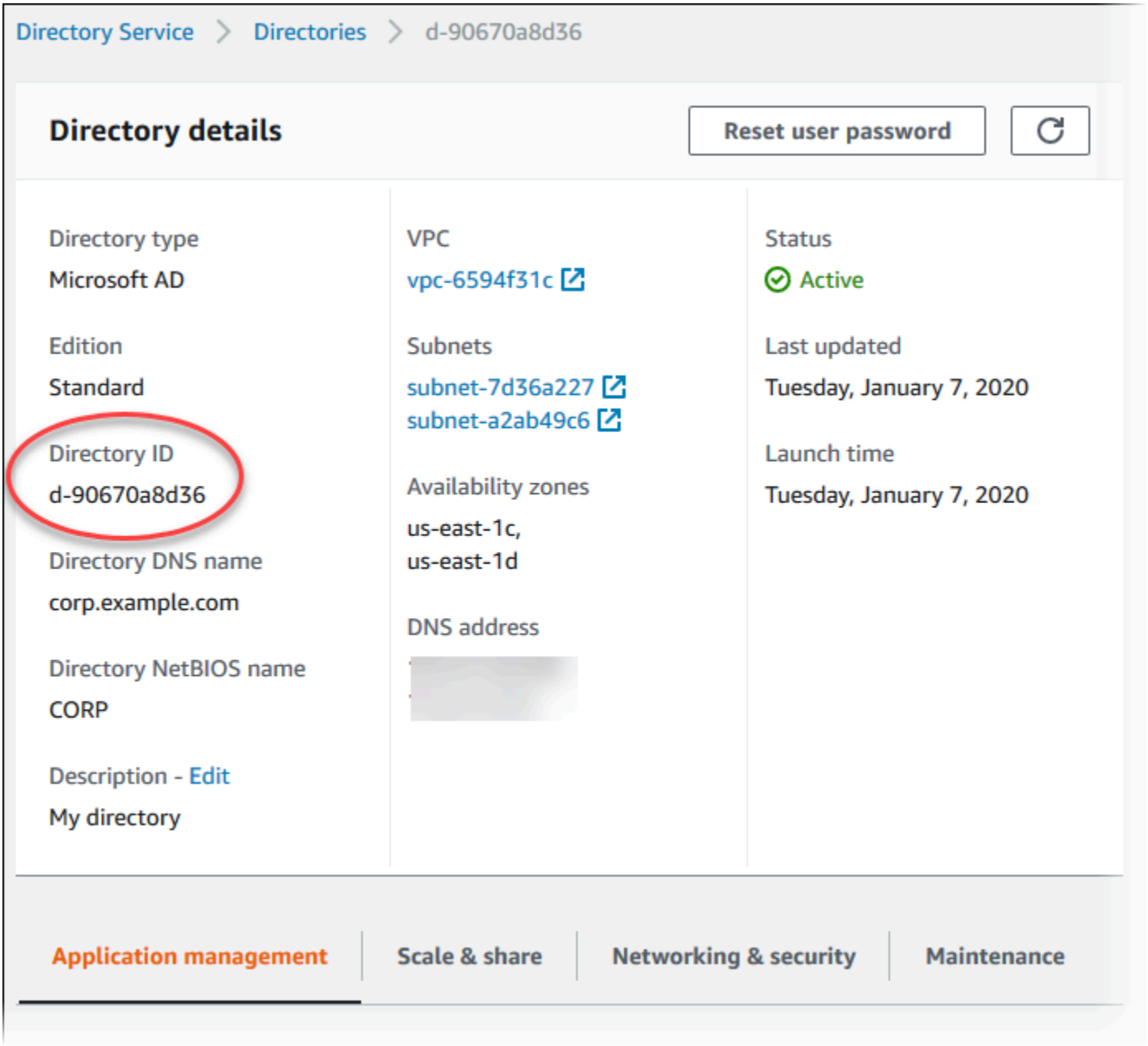
   Sottoreti

   Seleziona le sottoreti per i server di directory. Le due sottoreti devono trovarsi in diverse zone di disponibilità.
7. Seleziona Successivo.
8. Verificare le informazioni della directory Se sono necessarie modifiche, seleziona Previous (Precedente) e apporta le modifiche. Quando le informazioni sono corrette, scegli Create Directory (Crea directory).



Per creare la directory sono necessari alcuni minuti. Una volta creata correttamente la directory, il valore Status (Stato) viene modificato in Active (Attivo).

Per consultare le informazioni sulla directory, selezionare l'ID della directory nell'elenco di directory. Prendere nota del valore Directory ID (ID directory). Questo valore è necessario per creare o modificare l'istanza database PostgreSQL.



Passaggio 2: (Facoltativo) Creare una relazione di fiducia tra Active Directory locale e AWS Directory Service

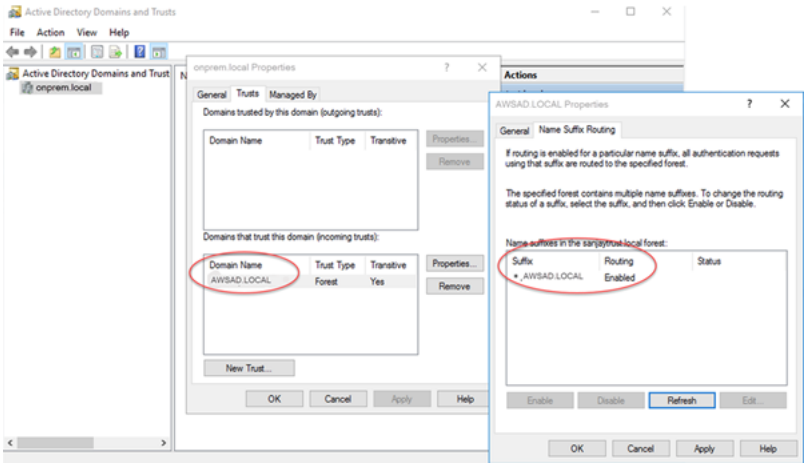
Se non prevedi di utilizzare Microsoft Active Directory locale, passa a Fase 3: creare un ruolo IAM per RDS per accedere a AWS Directory Service.

Per ottenere l'autenticazione Kerberos utilizzando l'Active Directory locale, è necessario creare una relazione di dominio affidabile utilizzando un trust di foresta tra Microsoft Active Directory locale e la AWS Managed Microsoft AD directory (creata in). Passaggio 1: creare una directory utilizzando AWS Managed Microsoft AD L'attendibilità può essere unidirezionale, in cui la AWS Managed

Microsoft AD directory considera attendibile Microsoft Active Directory locale. Il trust può anche essere bidirezionale, in cui entrambe le Active Directory si considerano reciprocamente attendibili. Per ulteriori informazioni sulla configurazione dei trust utilizzando AWS Directory Service, vedere Quando creare una relazione di trust nella Administration Guide.AWS Directory Service

> **Note**
>
> Se utilizzi un Microsoft Active Directory locale, i client Windows si connettono utilizzando il nome di dominio di AWS Directory Service nell'endpoint anziché rds.amazonaws.com. Per ulteriori informazioni, consulta Connessione a PostgreSQL con Autenticazione Kerberos.

Assicurati che il nome di dominio di Microsoft Active Directory locale includa un routing del suffisso DNS che corrisponde alla nuova relazione di trust creata. Il risultato è mostrato nella screenshot seguente.



Fase 3: creare un ruolo IAM per RDS per accedere a AWS Directory Service

RDS possa AWS Directory Service chiamarti, il tuo AWS account deve avere un ruolo IAM che utilizzi la policy IAM gestita. `AmazonRDSDirectoryServiceAccess` Questo ruolo permette ad Amazon RDS di effettuare chiamate ad AWS Directory Service.

Quando crei un'istanza DB utilizzando l'account utente della console AWS Management Console e l'account utente della console dispone dell'`iam:CreateRole`autorizzazione, la console crea automaticamente il ruolo IAM necessario. In questo caso, il nome del ruolo è `rds-directoryservice-kerberos-access-role`. In caso contrario, è necessario creare manualmente il ruolo IAM. Quando crei questo ruolo IAM`Directory Service`, scegli e allega la policy AWS gestita `AmazonRDSDirectoryServiceAccess` ad esso.

Per ulteriori informazioni sulla creazione di ruoli IAM per un servizio, consulta Creating a role to delegate permissions to an AWS service nella IAM User Guide.

> **Note**
>
> Il ruolo IAM utilizzato per l'autenticazione Windows per RDS per Microsoft SQL Server non può essere utilizzato per Amazon RDS for PostgreSQL.

Facoltativamente, puoi creare policy con le autorizzazioni richieste anziché utilizzare la policy `AmazonRDSDirectoryServiceAccess` gestita. In questo caso, il ruolo IAM deve avere la seguente policy di attendibilità IAM.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "directoryservice.rds.amazonaws.com",
          "rds.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Il ruolo deve anche disporre della seguente policy del ruolo IAM.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "ds:DescribeDirectories",
        "ds:AuthorizeApplication",
        "ds:UnauthorizeApplication",
        "ds:GetAuthorizedApplicationDetails"
```

```
      ],
    "Effect": "Allow",
    "Resource": "*"
    }
  ]
}
```

Fase 4: creazione e configurazione di utenti

Puoi creare gli utenti con lo strumento Utenti Active Directory e computer. Questo è uno degli strumenti di Active Directory Domain Services e Active Directory Lightweight Directory Services. Per ulteriori informazioni, consulta Add Users and Computers to the Active Directory domain (Aggiungere utenti e computer al dominio di Active Directory) nella documentazione di Microsoft. In questo caso, gli utenti sono individui o altre entità, ad esempio i computer, che fanno parte del dominio e le cui identità vengono conservate nella directory.

Per creare utenti in una AWS Directory Service directory, devi essere connesso a un'istanza Amazon EC2 basata su Windows che fa parte della directory. AWS Directory Service Allo stesso tempo, devi essere connesso come un utente che dispone di privilegi per creare utenti. Per ulteriori informazioni, consulta Creazione di un utente nella Guida di amministrazione di AWS Directory Service .

Fase 5: abilitazione del traffico tra VPC tra la directory e l'istanza database

Se prevedi di individuare la directory e l'istanza database nello stesso VPC, ignora questa fase e passa a Fase 6: creazione o modifica di un'istanza database PostgreSQL.

Se prevedi di individuare la directory e l'istanza database in VPC differenti, configura il traffico tra VPC utilizzando il peering di VPC o AWS Transit Gateway.

La procedura seguente abilita il traffico tra VPC utilizzando il peering di VPC. Segui le istruzioni in Che cos'è il peering di VPC? nella Amazon Virtual Private Cloud Peering Guide.

Per abilitare il traffico tra VPC utilizzando il peering di VPC

1. Configurare le regole di routing VPC appropriate per garantire che il traffico di rete possa scorrere in entrambe le direzioni.
2. Assicurati che il gruppo di sicurezza dell'istanza database possa ricevere traffico in entrata dal gruppo di sicurezza della directory.
3. Assicurati che non sia presente una regola della lista di controllo accessi (ACL) di rete per bloccare il traffico.

Se la directory è di proprietà di un altro AWS account, devi condividerla.

Per condividere la cartella tra AWS account

1. Inizia a condividere la directory con l' AWS account in cui verrà creata l'istanza DB seguendo le istruzioni riportate nel Tutorial: Sharing your AWS Managed Microsoft AD directory for seamless EC2 Domain-join nella Administration Guide.AWS Directory Service
2. Accedi alla AWS Directory Service console utilizzando l'account per l'istanza DB e assicurati che il dominio abbia lo stato prima di procedere. `SHARED`
3. Dopo aver effettuato l'accesso alla AWS Directory Service console utilizzando l'account per l'istanza DB, annota il valore Directory ID. Utilizzare questo ID directory per aggiungere l'istanza database al dominio.

Fase 6: creazione o modifica di un’istanza database PostgreSQL

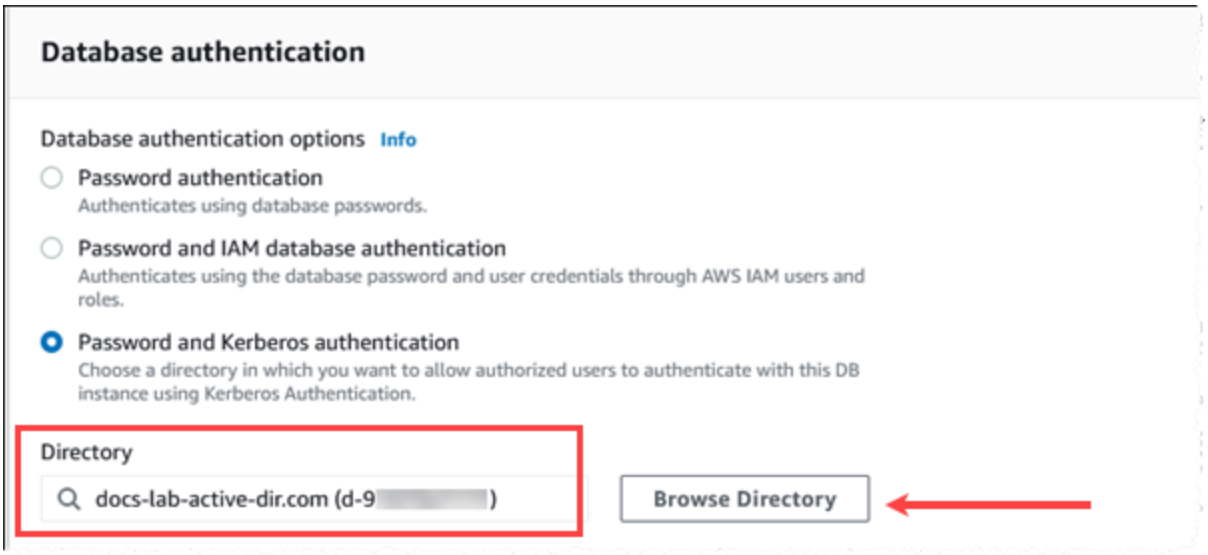
Crea o modifica un'istanza database di PostgreSQL da usare con la directory. Puoi utilizzare la console, CLI o l'API di RDS per associare un'istanza database a una directory. Questa operazione può essere eseguita in uno dei seguenti modi:

- Crea una nuova istanza DB PostgreSQL utilizzando la console, il comando create-db-instanceCLI o l'operazione API CreateDBInstance RDS. Per istruzioni, consulta Creazione di un'istanza database Amazon RDS.
- Modifica un'istanza DB PostgreSQL esistente utilizzando la console, il comando modify-db-instanceCLI o l'operazione API ModifyDBInstance RDS. Per istruzioni, consulta Modifica di un'istanza database Amazon RDS.
- Ripristina un'istanza DB PostgreSQL da uno snapshot DB utilizzando la console, il comando CLI restore-db-instance-from-db-snapshot o l'operazione API RestoreDB DBSnapshot RDS. InstanceFrom Per istruzioni, consulta Ripristino da uno snapshot database.
- Ripristina un'istanza DB PostgreSQL utilizzando la console, il comando restore-db-instance-to-point-in-time CLI o l'operazione dell'API RestoreDB RDS. point-in-time InstanceToPointInTime Per istruzioni, consulta Ripristino a un'ora specifica per un'istanza database.

L'autenticazione Kerberos è supportata solo per istanze di PostgreSQL DB in un VPC. L'istanza database può trovarsi nello stesso VPC della directory o in un VPC diverso. Il l’istanza database deve utilizzare un gruppo di sicurezza che accetta traffico in ingresso e in uscita all'interno del VPC della directory, in modo che l’istanza database possa comunicare con la directory.

Console

Quando utilizzi la console per creare, modificare o ripristinare un'istanza database, scegli Password e autenticazione Kerberos nella sezione Autenticazione database. Quindi scegli Sfoglia directory. Seleziona la directory o seleziona Crea una nuova directory per utilizzare il servizio directory.



AWS CLI

Quando si utilizza il AWS CLI, sono necessari i seguenti parametri affinché l'istanza del DB possa utilizzare la directory creata:

- Per il parametro `--domain`, utilizza l'identificatore di dominio (identificatore "d-*") generato durante la creazione della directory.
- Per il parametro `--domain-iam-role-name`, utilizza il ruolo creato che utilizza la policy IAM gestita `AmazonRDSDirectoryServiceAccess`.

Ad esempio, il comando CLI seguente modifica un'istanza database per utilizzare una directory.

```
aws rds modify-db-instance --db-instance-identifier mydbinstance --domain d-Directory-ID --domain-iam-role-name role-name
```

> **Important**
>
> Se modifichi un'istanza database per abilitare l'autenticazione Kerberos, riavvia il l'istanza database dopo aver apportato la modifica.

Fase 7: creazione di utenti PostgreSQL per i principali Kerberos

A questo punto, l'istanza database RDS per PostgreSQL viene aggiunta al dominio AWS Managed Microsoft AD . Gli utenti che hai creato nella directory in Fase 4: creazione e configurazione di utenti devono essere impostati come utenti del database PostgreSQL e devono essere concessi loro i privilegi per accedere al database. Puoi farlo accedendo come utente del database con privilegi `rds_superuser`. Ad esempio, se hai accettato i valori predefiniti quando hai creato l'istanza database RDS per PostgreSQL, utilizzi `postgres`, come illustrato nei passaggi seguenti.

Per creare utenti del database PostgreSQL per i principali Kerberos

1. Usa `psql` per la connessione all'endpoint dell'istanza database RDS per PostgreSQL utilizzando `psql`. L'esempio seguente utilizza l'account `postgres` predefinito per il ruolo `rds_superuser`.

   ```
   psql --host=cluster-instance-1.111122223333.aws-region.rds.amazonaws.com --
   port=5432 --username=postgres --password
   ```

2. Crea un nome utente del database per ogni principale Kerberos (nome utente di Active Directory) a cui desideri fornire l'accesso al database. Utilizza il nome utente canonico (identità) come definito nell'istanza Active Directory, ovvero un `alias` minuscolo (nome utente in Active Directory) e il nome maiuscolo del dominio Active Directory per quel nome utente. Il nome utente di Active Directory è un utente autenticato esternamente, quindi usa le virgolette intorno al nome come mostrato di seguito.

   ```
   postgres=> CREATE USER "username@CORP.EXAMPLE.COM" WITH LOGIN;
   CREATE ROLE
   ```

3. Autorizza il ruolo `rds_ad` per l'utente del database.

   ```
   postgres=> GRANT rds_ad TO "username@CORP.EXAMPLE.COM";
   GRANT ROLE
   ```

Dopo aver completato la creazione di tutti gli utenti PostgreSQL per le identità utente Active Directory, gli utenti possono accedere all'istanza database RDS per PostgreSQL utilizzando le proprie credenziali Kerberos.

È necessario che gli utenti del database che si autenticano tramite Kerberos lo facciano da computer client membri del dominio Active Directory.

Gli utenti del database a cui è stato concesso il ruolo `rds_ad` non possono avere anche il ruolo `rds_iam`. Questo vale anche per le appartenenze nidificate. Per ulteriori informazioni, consulta Autenticazione del database IAM per MariaDB, MySQL e PostgreSQL.

Fase 8: configurazione di un client PostgreSQL

Per configurare un client PostgreSQL, procedi come indicato di seguito:

- Crea un file krb5.conf (o equivalente) che punti al dominio.
- Verifica che il traffico possa fluire tra l'host del client e AWS Directory Service. Utilizza un'utilità di rete come Netcat per le operazioni seguenti:
  - Verifica il traffico su DNS per la porta 53.
  - Verifica il traffico su TCP/UDP per la porta 53 e per Kerberos, che include le porte 88 e 464 per AWS Directory Service.
- Verifica che il traffico scorra senza problemi tra l'host client e l'istanza database sulla porta del database. Ad esempio, utilizza psql per connetterti e accedere al database.

Di seguito è riportato un esempio di contenuto krb5.conf per. AWS Managed Microsoft AD

```
[libdefaults]
 default_realm = EXAMPLE.COM
[realms]
 EXAMPLE.COM = {
  kdc = example.com
  admin_server = example.com
 }
[domain_realm]
 .example.com = EXAMPLE.COM
 example.com = EXAMPLE.COM
```

Di seguito è riportato un esempio di contenuto krb5.conf per una Microsoft Active Directory locale.

```
[libdefaults]
 default_realm = EXAMPLE.COM
[realms]
 EXAMPLE.COM = {
  kdc = example.com
  admin_server = example.com
 }
 ONPREM.COM = {
```

```
  kdc = onprem.com
  admin_server = onprem.com
 }
[domain_realm]
 .example.com = EXAMPLE.COM
 example.com = EXAMPLE.COM
 .onprem.com = ONPREM.COM
 onprem.com = ONPREM.COM
 .rds.amazonaws.com = EXAMPLE.COM
 .amazonaws.com.cn = EXAMPLE.COM
 .amazon.com = EXAMPLE.COM
```

Gestione di un'istanza database in un dominio

Puoi utilizzare la console, la CLI o l'API di RDS per gestire l'istanza database e la sua relazione con Microsoft Active Directory. Ad esempio, puoi associare una Active Directory per abilitare l'autenticazione Kerberos. Puoi anche annullare l'associazione ad Active Directory per disabilitare l'autenticazione Kerberos. Puoi anche spostare un'istanza database affinché venga autenticata esternamente da una Microsoft Active Directory a un'altra.

Ad esempio, utilizzando la CLI, puoi effettuare quanto segue:

- Per provare ad abilitare di nuovo l'autenticazione Kerberos per un'appartenenza non riuscita, utilizza il comando CLI modify-db-instance . Specifica l'ID della directory dell'appartenenza corrente per l'opzione `--domain`.
- Per disabilitare l'autenticazione Kerberos su un'istanza database, utilizza il comando CLI modify-db-instance . Specifica `none` per l'opzione `--domain`.
- Per spostare un'istanza database da un dominio all'altro, utilizza il comando CLI modify-db-instance . Specifica l'identificatore del nuovo dominio per l'opzione `--domain`.

Appartenenza al dominio

Dopo avere creato o modificato l'istanza database, diventa un membro del dominio. Puoi visualizzare lo stato dell'appartenenza al dominio nella console o eseguendo il comando CLI describe-db-instances. Lo stato dell'istanza di database può essere uno dei seguenti:

- `kerberos-enabled`: l'autenticazione Kerberos è abilitata nell'istanza database.
- `enabling-kerberos`: AWS si trova nella fase di abilitazione dell'autenticazione Kerberos su questa istanza database.

- `pending-enable-kerberos`: l'abilitazione dell'autenticazione Kerberos è in corso su questa istanza database.
- `pending-maintenance-enable-kerberos`: AWS proverà ad abilitare l'autenticazione Kerberos sull'istanza database durante la prossima finestra di manutenzione pianificata.
- `pending-disable-kerberos`: la disabilitazione dell'autenticazione Kerberos è in corso su questa istanza database.
- `pending-maintenance-disable-kerberos`: AWS proverà a disabilitare l'autenticazione Kerberos sull'istanza database durante la prossima finestra di manutenzione pianificata.
- `enable-kerberos-failed`: un problema di configurazione ha impedito ad AWS di abilitare l'autenticazione Kerberos sull'istanza database. Correggi il problema di configurazione prima di inviare nuovamente il comando per modificare l'istanza database.
- `disabling-kerberos`: AWS si trova nella fase di disabilitazione dell'autenticazione Kerberos su questa istanza database.

Una richiesta per abilitare l'autenticazione Kerberos potrebbe non andare a buon fine a causa di un problema di connettività di rete o un ruolo IAM non corretto. In alcuni casi, il tentativo di abilitare l'autenticazione Kerberos potrebbe non riuscire quando crei o modifichi un'istanza database. In questo caso, verifica di utilizzare il ruolo IAM corretto, quindi modifica l'istanza database per l'aggiunta al dominio.

> **Note**
>
> Solo l'autenticazione Kerberos con RDS per PostgreSQL invia il traffico ai server DNS del dominio. Tutte le altre richieste DNS vengono gestite come accesso di rete in uscita sulle istanze database che eseguono PostgreSQL. Per ulteriori informazioni sull'accesso di rete in uscita con RDS per PostgreSQL, consulta Utilizzo di un Server DNS personalizzato per Outbound Network Access..

Connessione a PostgreSQL con Autenticazione Kerberos

Puoi connetterti a PostgreSQL con l'autenticazione Kerberos tramite l'interfaccia pgAdmin o un'interfaccia a riga di comando come psql. Per ulteriori informazioni sulla connessione, consulta Connessione a un'istanza database che esegua il motore di database di PostgreSQL . Per informazioni su come ottenere l'endpoint, il numero di porta e altri dettagli necessari per la connessione, consulta Fase 3: connessione a un'istanza database PostgreSQL.

pgAdmin

Per utilizzare pgAdmin per connetterti a PostgreSQL con l'autenticazione Kerberos, completa la procedura seguente:

1. Avviare l'applicazione pgAdmin sul computer client.
2. Nella scheda Dashboard (Pannello di controllo) selezionare Add New Server (Aggiungi nuovo server).
3. Nella finestra di dialogo Crea - Server, immettere un nome nella scheda Generale per identificare il server in pgAdmin.
4. Nella scheda Connection (Connessione), immetti le informazioni seguenti relative al database RDS per PostgreSQL.
   - In Host, immetti l'endpoint per l' l'istanza database RDS per PostgreSQL. Un endpoint è simile al seguente:

     ```
     RDS-DB-instance.111122223333.aws-region.rds.amazonaws.com
     ```

     Per connetterti a Microsoft Active Directory on-premise da un client Windows, usa il nome di dominio di Active Directory gestito da AWS anziché `rds.amazonaws.com` nell'endpoint host. Si supponga, ad esempio, che il nome di dominio per la Active Directory gestita da AWS sia `corp.example.com`. In Host, l'endpoint viene quindi specificato come segue:

     ```
     RDS-DB-instance.111122223333.aws-region.corp.example.com
     ```

   - Per Porta, immettere la porta assegnata.
   - In Database di manutenzione immettere il nome del database iniziale a cui si connetterà il client.
   - In Nome utente, immettere il nome utente immesso per l'autenticazione Kerberos in Fase 7: creazione di utenti PostgreSQL per i principali Kerberos .
5. Seleziona Salva.

Psql

Per utilizzare psql per connetterti a PostgreSQL con l'autenticazione Kerberos, completare la procedura seguente:

1. Al prompt dei comandi, eseguire questo comando.

```
kinit username
```

Sostituire *username* con il nome utente. Al prompt, immettere la password per l'utente memorizzata in Microsoft Active Directory.

2. Se l'istanza database PostgreSQL utilizza un VPC accessibile pubblicamente, inserire un indirizzo IP per l'endpoint dell'istanza database nel file `/etc/hosts` nelclient EC2. Ad esempio, i comandi seguenti ottengono l'indirizzo IP e lo inseriscono nel file `/etc/hosts`.

```
% dig +short PostgreSQL-endpoint.AWS-Region.rds.amazonaws.com
;; Truncated, retrying in TCP mode.
ec2-34-210-197-118.AWS-Region.compute.amazonaws.com.
34.210.197.118

% echo " 34.210.197.118  PostgreSQL-endpoint.AWS-Region.rds.amazonaws.com" >> /etc/
hosts
```

Se utilizzi una Microsoft Active Directory locale da un client Windows, dovrai connetterti utilizzando un endpoint speciale. Anziché utilizzare il dominio Amazon `rds.amazonaws.com` nell'endpoint host, utilizzare il nome di dominio della Active Directory gestita da AWS.

Si supponga, ad esempio, che il nome di dominio per la Active Directory gestita da AWS sia `corp.example.com`. Quindi utilizzare il formato *PostgreSQL-endpoint.AWS-Region.corp.example.com* per l'endpoint e inserirlo nel file `/etc/hosts`.

```
% echo " 34.210.197.118  PostgreSQL-endpoint.AWS-Region.corp.example.com" >> /etc/
hosts
```

3. Utilizzare il comando psql seguente per accedere a un'istanza database PostgreSQL integrata con Active Directory.

```
psql -U username@CORP.EXAMPLE.COM -p 5432 -h PostgreSQL-endpoint.AWS-
Region.rds.amazonaws.com postgres
```

Per accedere al cluster di database PostgreSQL da un client Windows utilizzando una Active Directory locale, utilizzare il comando psql seguente con il nome di dominio del passaggio precedente (`corp.example.com`):

```
psql -U username@CORP.EXAMPLE.COM -p 5432 -h PostgreSQL-endpoint.AWS-
Region.corp.example.com postgres
```

Utilizzo di un Server DNS personalizzato per Outbound Network Access.

RDS for PostgreSQL supporta l'accesso di rete in uscita sulle istanze database e consente la risoluzione Domain Name Service (DNS) da un server DNS personalizzato di proprietà del cliente. È possibile risolvere solo nomi di dominio completamente qualificati dall'istanza database RDS for PostgreSQL tramite il server DNS personalizzato.

Argomenti

Attivazione della risoluzione DNS personalizzata

Per attivare la risoluzione DNS nel VPC del cliente, associa innanzitutto un gruppo parametri del database personalizzato all'istanza RDS for PostgreSQL. Quindi attiva il parametro `rds.custom_dns_resolution` impostandolo su 1 e riavvia l'istanza database affinché le modifiche diventino effettive.

Disattivazione della risoluzione DNS personalizzata

Per disattivare la risoluzione DNS nel VPC del cliente, disattiva innanzitutto il parametro `rds.custom_dns_resolution` del gruppo parametri del database personalizzato impostandolo su 0. Quindi riavvia l'istanza database affinché le modifiche diventino effettive.

Impostazione di un server DNS personalizzato

Dopo aver impostato il server dei nomi DNS personalizzato, ci vogliono circa 30 minuti per propagare le modifiche all'istanza database. Dopo che le modifiche vengono propagate all'istanza database, tutto il traffico di rete in uscita che richiede una ricerca DNS esegue una query al server DNS tramite la porta 53.

> **Note**
>
> Se non si imposta un server DNS personalizzato e `rds.custom_dns_resolution` è impostato su 1, gli host vengono risolti utilizzando una zona privata di Amazon Route 53. Per ulteriori informazioni, consulta Utilizzo delle zone ospitate private.

Come configurare un server DNS personalizzato per l'istanza database RDS for PostgreSQL

1. Dal set di opzioni Protocollo di configurazione per host dinamico (DHCP) collegate al VPC, imposta l'opzione `domain-name-servers` all'indirizzo IP del server dei nomi DNS. Per ulteriori informazioni, consulta la pagina relativa ai Set di opzioni DHCP.

   > **Note**
   >
   > L'opzione `domain-name-servers` accetta fino a quattro valori, ma l'istanza database Amazon RDS usa solo il primo valore.

2. Assicurati che il server DNS possa risolvere tutte le query di ricerca, compresi i nomi DNS pubblici, i nomi DNS privati Amazon EC2 e i nomi DNS specifici per i clienti. Se il traffico di rete in uscita contiene ricerche DNS che il server DNS non può gestire, il server DNS deve avere fornitori DNS upstream appropriati configurati.
3. Configura il server DNS per produrre risposte UDP (User Datagram Protocol) di 512 byte o meno.
4. Configura il server DNS per produrre risposte TCP (Transmission Control Protocol) di 1024 byte o meno.
5. Configura il server DNS per consentire il traffico in entrata dalle istanze database Amazon RDS tramite la porta 53. Se il server DNS si trova in un Amazon VPC, il VPC deve avere un gruppo di sicurezza che contiene regole in entrata che permettono traffico UDP e TCP tramite la porta 53. Se il server DNS non si trova in un Amazon VPC, deve avere impostazioni firewall appropriate per permettere traffico in entrata UDP e TCP tramite la porta 53.

   Per ulteriori informazioni, consulta Gruppi di sicurezza per il VPC e Aggiunta e rimozione di regole.
6. Configura il VPC dell'istanza database Amazon RDS per permettere traffico in uscita tramite la porta 53. Il VPC deve avere un gruppo di sicurezza che contiene regole in uscita che permettono traffico UDP e TCP tramite la porta 53.

Per ulteriori informazioni, consulta Gruppi di sicurezza per il VPC e Aggiunta ed eliminazione delle regole nella Guida per l'utente di Amazon VPC.

7. Accertati che il percorso di routing tra l'istanza database Amazon RDS e il server DNS sia configurato correttamente per consentire traffico DNS.

   Inoltre, se l'istanza database Amazon RDS e il server DNS non si trovano nello stesso VPC, una connessione peer deve essere configurata tra loro. Per ulteriori informazioni, consulta Che cos'è il peering di VPC? nella Guida di Amazon VPC Peering.

Aggiornamento del motore del database PostgreSQL per Amazon RDS

Esistono due tipi di aggiornamenti che puoi gestire per il tuo database PostgreSQL:

- Aggiornamenti del sistema operativo: a volte Amazon RDS potrebbe dover aggiornare il sistema operativo sottostante del database per applicare correzioni di sicurezza o modifiche del sistema operativo. Puoi decidere quando Amazon RDS applicare gli aggiornamenti del sistema operativo utilizzando la console RDS, AWS Command Line Interface (AWS CLI) o l'API RDS. Per ulteriori informazioni sugli aggiornamenti del sistema operativo, consulta Applicazione di aggiornamenti a un'istanza database.
- Aggiornamenti del motore di database: quando Amazon RDS supporta una nuova versione di un motore di database, puoi aggiornare i database alla nuova versione.

Un database in questo contesto è un'istanza database RDS per PostgreSQL o un cluster database multi-AZ.

Esistono due tipi di aggiornamenti del motore per i database PostgreSQL: aggiornamenti delle versioni principali e aggiornamenti delle versioni secondarie.

Aggiornamenti di una versione principale

Gli aggiornamenti di versione principali possono contenere modifiche al database non compatibili con le versioni precedenti delle applicazioni esistenti. Ne risulta che è necessario eseguire manualmente gli aggiornamenti della versione principale per i propri database. Puoi avviare manualmente un aggiornamento della versione principale modificando l'istanza database o il cluster database multi-AZ. Prima di eseguire un aggiornamento della versione principale, si consiglia di seguire i passaggi descritti in. Scelta di un aggiornamento di versione principale per PostgreSQL

Se aggiorni un'istanza database con repliche di lettura nella regione, Amazon RDS aggiorna le repliche assieme all'istanza database primaria.

Amazon RDS non aggiorna le repliche di lettura del cluster database multi-AZ. Se si esegue un aggiornamento della versione principale di un cluster DB Multi-AZ, lo stato di replica delle relative repliche di lettura diventa terminato. Devi eliminare e ricreare manualmente le repliche di lettura al completamento dell'aggiornamento.

> **Tip**
>
> È possibile ridurre al minimo i tempi di inattività necessari per l'aggiornamento di una versione principale utilizzando una distribuzione blu/verde. Per ulteriori informazioni, consulta Utilizzo delle implementazioni blu/verde per gli aggiornamenti del database.

Aggiornamenti della versione secondaria

Al contrario, gli aggiornamento secondari a una versione includono solo modifiche compatibili con le versioni precedenti delle applicazioni esistenti. Puoi avviare manualmente un aggiornamento della versione secondaria modificando il cluster di database. In alternativa, è possibile abilitare l'opzione di aggiornamento automatico della versione secondaria durante la creazione o la modifica di un database. Ciò significa che Amazon RDS aggiorna automaticamente il database dopo aver testato e approvato la nuova versione. Se il database PostgreSQL utilizza repliche di lettura, è necessario aggiornare tutte le repliche di lettura prima di aggiornare l'istanza o il cluster di origine.

Se il database è una distribuzione di istanze DB Multi-AZ, Amazon RDS aggiorna contemporaneamente l'istanza primaria e tutte le istanze di standby. Pertanto, il database potrebbe non essere disponibile fino al completamento dell'aggiornamento. Se il database è una distribuzione di cluster DB Multi-AZ, Amazon RDS aggiorna le istanze Reader DB una alla volta. Quindi, una delle istanze Reader DB diventa la nuova istanza DB Writer. Amazon RDS aggiorna quindi la vecchia istanza writer (che ora è un'istanza reader).

> **Note**
>
> Il tempo di inattività per un aggiornamento di versione minore di un'implementazione di un'istanza DB Multi-AZ può durare diversi minuti. I cluster DB Multi-AZ in genere riducono i tempi di inattività degli aggiornamenti di versioni minori a circa 35 secondi. Se utilizzati con RDS Proxy, è possibile ridurre ulteriormente i tempi di inattività a un secondo o meno. Per ulteriori informazioni, consulta Utilizzo del Proxy RDS. In alternativa, è possibile utilizzare un proxy di database open source come ProxySQL o il driver PgBouncer AWSJDBC per MySQL.

Per ulteriori informazioni, consulta Aggiornamenti a versioni secondarie automatiche per PostgreSQL. Per ulteriori informazioni sull'esecuzione manuale di un aggiornamento alla versione secondaria, consulta Aggiornamento manuale della versione del motore.

Per ulteriori informazioni sulle versioni dei motori di database e sulla politica per la deprecazione delle versioni dei motori di database, consulta Versioni del motore di database nelle domande frequenti su Amazon RDS.

Argomenti

Panoramica dell'aggiornamento PostgreSQL

Per aggiornare i database in modo sicuro, Amazon RDS utilizza la utilità `pg_upgrade` descritta nella documentazione di PostgreSQL.

Quando usi il AWS Management Console per aggiornare un database, mostra gli obiettivi di aggiornamento validi per il database. È inoltre possibile utilizzare il AWS CLI comando seguente per identificare gli obiettivi di aggiornamento validi per un database:

Per LinuxmacOS, oUnix:

```
aws rds describe-db-engine-versions \
  --engine postgres \
  --engine-version version-number \
  --query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" --
output text
```

Per Windows:

```
aws rds describe-db-engine-versions ^
```

```
  --engine postgres ^
  --engine-version version-number ^
  --query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" --
output text
```

Ad esempio, per identificare gli obiettivi di aggiornamento validi per un database PostgreSQL versione 12.13, esegui il comando seguente: AWS CLI

PerLinux, o: macOS Unix

```
aws rds describe-db-engine-versions \
  --engine postgres \
  --engine-version 12.13 \
  --query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" --
output text
```

Per Windows:

```
aws rds describe-db-engine-versions ^
  --engine postgres ^
  --engine-version 12.13 ^
  --query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" --
output text
```

Amazon RDS esegue due snapshot DB durante il processo di aggiornamento se il tempo di conservazione del backup è maggiore di 0. Il primo snapshot database è relativo al database prima delle modifiche associate all'aggiornamento. Se l'aggiornamento dei database ha esito negativo, puoi ripristinare questo snapshot per creare un database che esegue la versione precedente. Il secondo snapshot DB viene acquisito al termine dell'aggiornamento.

> **Note**
>
> Amazon RDS acquisisce gli snapshot database durante il processo di aggiornamento solo se hai impostato il periodo di conservazione dei backup per il database impostando un valore maggiore di 0. Per modificare il periodo di conservazione dei backup per un'istanza database, consulta the section called “Modifica di un'istanza database”. Non è possibile configurare un periodo di conservazione dei backup personalizzato per un cluster database multi-AZ.

Quando aggiorni la versione principale di un'istanza database, tutte le repliche di lettura nella regione vengono aggiornate automaticamente. Dopo l'avvio del flusso di lavoro di aggiornamento, le istanze di repliche di lettura attendono che `pg_upgrade` venga completato correttamente sull'istanza database primaria. Quindi l'aggiornamento dell'istanza database primaria attende il completamento degli aggiornamenti di replicche di lettura. Finché l'aggiornamento non è completato, si verifica un'interruzione. Quando esegui l'aggiornamento della versione principale di un cluster database multi-AZ, lo stato delle relative repliche di lettura cambia in Terminato.

Al completamento di un aggiornamento, non è possibile ripristinare la versione precedente del motore di database. Se desideri ripristinare la versione precedente, ripristina lo snapshot database acquisito prima dell'aggiornamento per creare un nuovo database.

Numeri di versione di PostgreSQL

La sequenza di numerazione delle versioni per il motore del database PostgreSQL è la seguente:

- Per PostgreSQL versioni 10 e successive, il formato del numero di versione del motore è principale.secondario. Il numero di versione principale è la parte intera del numero di versione. Il numero di versione secondaria è la parte frazionaria del numero di versione.

  Un aggiornamento della versione principale incrementa la parte intera del numero di versione, ad esempio l'aggiornamento da 10.secondario a 11.secondario.
- Per le versioni di PostgreSQL precedenti alla 10, il formato del numero di versione del motore è principale.principale.secondaria. Il numero di versione principale del motore è il numero intero che la prima parte frazionaria del numero di versione. Ad esempio, 9.6 è una versione principale. Il numero di versione secondaria è la terza parte del numero di versione. Ad esempio, per la versione 9.6.12, 12 è il numero della versione secondaria.

  Un aggiornamento della versione principale incrementa la parte principale del numero di versione. Ad esempio, un aggiornamento da 9.6.12 a 11.14 è un aggiornamento della versione principale, in cui 9.6 e 11 sono i numeri delle versioni principali.

Per informazioni sulla numerazione delle versioni di RDS Extended Support, vedere. Denominazione delle versioni di Amazon RDS Extended Support

Numero di versione RDS

I numeri di versione RDS utilizzano lo schema di denominazione *major*.*minor*.*patch*. Una versione della patch di RDS include importanti correzioni di bug aggiunte a una versione secondaria

dopo il rilascio. Per informazioni sulla numerazione delle versioni di RDS Extended Support, vedere. Denominazione delle versioni di Amazon RDS Extended Support

Per identificare il numero di versione Amazon RDS del tuo database, è prima necessario creare l'estensione `rds_tools` utilizzando il seguente comando:

```
CREATE EXTENSION rds_tools;
```

A partire dal rilascio di PostgreSQL versione 15.2-R2, è possibile trovare il numero di versione di RDS del database RDS per PostgreSQL con la seguente query SQL:

```
postgres=> SELECT rds_tools.rds_version();
```

Ad esempio, l'esecuzione di una query su un database RDS per PostgreSQL 15.2 restituisce quanto segue:

```
rds_version
----------------
 15.2.R2
(1 row)
```

Scelta di un aggiornamento di versione principale per PostgreSQL

È possibile che gli aggiornamenti a una versione principale contengano modifiche non compatibili con le versioni precedenti del database. La nuova funzionalità può causare l'interruzione del funzionamento corretto delle applicazioni esistenti. Per questo motivo, Amazon RDS non applica aggiornamenti automatici alla versione principale. Per eseguire un aggiornamento della versione principale, modifica il database manualmente. Accertati di testare in modo approfondito qualsiasi aggiornamento per verificare che le tue applicazioni funzionino correttamente prima di applicare l'aggiornamento ai database di produzione. Quando si esegue un aggiornamento della versione principale PostgreSQL, si consiglia di seguire i passaggi descritti in Come eseguire l'aggiornamento a una versione principale.

Quando aggiorni un'istanza database PostgreSQL single-AZ o una implementazione multi-AZ di un'istanza database alla versione principale successiva, anche le repliche di lettura associate al database vengono aggiornate alla versione principale successiva. In alcuni casi, è possibile passare a una versione principale successiva durante l'aggiornamento. Se l'aggiornamento ignora una versione principale, anche le repliche di lettura vengono aggiornate alla versione principale di destinazione. Gli aggiornamenti alla versione 11 che ignorano altre versioni principali

presentano alcune limitazioni. È possibile trovare i dettagli nei passaggi descritti in Come eseguire l'aggiornamento a una versione principale.

Un aggiornamento al motore PostgreSQL non aggiorna la maggior parte delle estensioni PostgreSQL. Queste devono essere aggiornate separatamente. Per ulteriori informazioni, consulta Aggiornamento estensioni PostgreSQL.

Puoi scoprire quali versioni principali sono disponibili per il tuo database RDS per PostgreSQL eseguendo la seguente query: AWS CLI

```
aws rds describe-db-engine-versions --engine postgres  --engine-version your-version
 --query "DBEngineVersions[*].ValidUpgradeTarget[*].{EngineVersion:EngineVersion}" --
output text
```

La tabella riportata di seguito riepiloga i risultati di questa query per tutte le versioni disponibili. Un asterisco (*) sul numero di versione indica che la versione è obsoleta. Se la versione corrente è obsoleta, ti consigliamo di eseguire l'aggiornamento alla versione secondaria più recente o a una delle altre destinazioni di aggiornamento disponibili per quella versione. Per ulteriori informazioni sull'obsolescenza di RDS per PostgreSQL versione 9.6, consulta Obsolescenza di PostgreSQL versione 9.6. Per ulteriori informazioni sull'obsolescenza di RDS per PostgreSQL versione 10, consulta Definizione come obsoleto di PostgreSQL versione 10.

Vers sorg corre (* obso	De oni di ag en alla ver pri e più rec	Destinazioni di aggiornamento disponibili
16.2		16
16,1		16 16

Vers sorg corre (* obso	De oni di ag en alla ver pri e più rec	Destinazioni di aggiornamento disponibili									
15,7	16										
15,6	16	16	15								
15,5	16	16	16	15	15						
15,4	16	16	16	15	15	15					
15,3	16	16	16	15	15	15	15				
15,2	16	16	16	15	15	15	15	15			
14,1	16	15									
14,1	16	15	15	14							
14,1	16	15	15	15	14	14					
14,9	15	15	15	15	14	14	14				
14,8	15	15	15	15	15	14	14	14	14		
14,7 *	15	15	15	15	15	14	14	14	14		
14,6	15	15	15	15	15	14	14	14	14	14	
14,5	15	15	15	15	15	14	14	14	14	14	14

Vers sorg corre (* obso	De oni di ag en alla ver pri e più rec	Destinazioni di aggiornamento disponibili																		
14,4	15	15	15	15	15	14	14	14	14	14	14	14								
14,3	15	15	15	15	15	14	14	14	14	14	14	14	14							
14.2	15	15	15	15	15	14	14	14	14	14	14	14	14	14						
14.1	15	15	15	15	15	14	14	14	14	14	14	14	14	14	14					
13,1	16	15	14																	
13,1	16	15	14	14	13															
13,1	16	15	14	14	14	13	13													
13,1	15	14	14	14	14	13	13	13												
13,1	15	14	14	14	14	14	13	13	13	13										
13,1	15	14	14	14	14	14	13	13	13	13										
13,9	14	14	14	14	14	14	13	13	13	13	13									
13,8	14	14	14	14	14	14	14	13	13	13	13	13	13							
13,7	14	14	14	14	14	14	14	14	14	13	13	13	13	13	13					
13.6	14	14	14	14	14	14	14	14	14	14	13	13	13	13	13	13	13			

Vers sorg corre (* obso	De oni di ag en alla ver pri e più rec	Destinazioni di aggiornamento disponibili																				
13.5*	14	14	14	14	14	14	14	14	14	14	14	13	13	13	13	13	13	13	13			
13.4*	14	14	14	14	14	14	14	14	14	14	14	13	13	13	13	13	13	13	13	13		
13.3*	14	14	14	14	14	14	14	14	14	14	14	13	13	13	13	13	13	13	13	13	13	
13.2* 13.1*	14	14	14	14	14	14	14	14	14	14	14	13	13	13	13	13	13	13	13	13	13	
12,1	16	15	14	13																		
12,1	16	15	14	13	13	12																
12,1	16	15	14	13	13	13	12	12														
12,1	15	14	13	13	13	13	12	12	12													
12,1	15	14	13	13	13	13	13	12	12	12	12											
12,1	15	14	13	13	13	13	13	12	12	12	12											
12,1	14	13	13	13	13	13	13	12	12	12	12	12										
12,1	14	13	13	13	13	13	13	13	12	12	12	12	12	12								
12,1	14	14	13	13	13	13	13	13	13	13	12	12	12	12	12	12	12					
12.1	14	13	13	13	13	13	13	13	13	13	12	12	12	12	12	12	12	12				

Vers sorg corre (* obso	De oni di ag en alla ver pri e più rec	Destinazioni di aggiornamento disponibili																					
12.9	14	13	13	13	13	13	13	13	13	13	13	12	12	12	12	12	12	12	12	12			
12.8	13	13	13	13	13	13	13	13	13	13	13	12	12	12	12	12	12	12	12	12	12		
12.7	13	13	13	13	13	13	13	13	13	13	13	13	12	12	12	12	12	12	12	12	12	12	12,8
12.6 12.5 12.4 12.3 12.2	13	13	13	13	13	13	12	12	12	12	12	12	12	12	12	12	12	12					
11,2	16	15	14	13	12	11 RD .20																	
11.2	15	14	13	12	12	11																	
11,2	15	14	13	12	12	12	11	11															
11,1	15	14	13	12	12	12	12	11	11	11													
11,1	14	13	12	12	12	12	12	11	11	11	11												
11,1	14	13	12	12	12	12	12	12	11	11	11	11	11										
11,1	14	14	13	12	12	12	12	12	12	12	11	11	11	11	11	11							

Vers sorg corre (* obso	De oni di ag en alla ver pri e più rec	Destinazioni di aggiornamento disponibili																					
11,1	14	13	12	12	12	12	12	12	12	12	11	11	11	11	11	11	11						
11,1	14	13	12	12	12	12	12	12	12	12	12	11	11	11	11	11	11	11	11				
11,1	13	12	12	12	12	12	12	12	12	12	12	11	11	11	11	11	11	11	11	11			
11,1	13	12	12	12	12	12	12	12	12	12	12	12	11	11	11	11	11	11	11	11	11	11	
10.2	14	13	12	11	11	11	11	11															
10.2	14	13	12	11	11	11	11	11	11	10													
10.2	14	14	13	12	11	11	11	11	11	11	11	10	10										
10.2	14	13	12	11	11	11	11	11	11	11	11	10	10	10									
10.1	14	13	12	11	11	11	11	11	11	11	11	11	10	10	10	10							
10.1	13	12	11	11	11	11	11	11	11	11	11	11	10	10	10	10	10						
10.1	13	12	11	11	11	11	11	11	11	11	11	11	11	10	10	10	10	10	10				
9.6.2	14	13	12	11	10	10																	
9.6.2	13	12	11	10	10	10	9,6																
9.6.2	13	12	11	10	10	10	10	9,6	9,6														

Vers sorg corre (* obso	De oni di ag en alla ver pri e più rec	Destinazioni di aggiornamento disponibili							
9.6.1 9.6.1 9.6.1 9.6.1 9.6.1 9.6.1 9.6.1 6.10* 9.6.9 9.6.8 9.6.6 9.6.5 9.6.3 9.6.2 9.6.1	9,6	14	13	12	11	10	10	9,6	9,6

Come eseguire l'aggiornamento a una versione principale

Consigliamo la seguente procedura per eseguire l'aggiornamento della versione principale su un database Amazon RDS per PostgreSQL:

1. Disponibile gruppo di parametri compatibile con la versione – Se si sta utilizzando un gruppo di parametri personalizzato, sono disponibili due opzioni. È possibile specificare un gruppo di

parametri predefinito per la nuova versione del motore database. Oppure è possibile creare un gruppo di parametri personalizzato per la nuova versione del motore database. Per ulteriori informazioni, consulta the section called “Utilizzo di gruppi di parametri” e the section called “Utilizzo di gruppi di parametri di cluster di database”.

2. Verifica della presenza di classi di database non supportate: verifica che la classe di istanza del database sia compatibile con la versione PostgreSQL a cui si sta eseguendo l'aggiornamento. Per ulteriori informazioni, consulta Motori DB supportati per classi di istanza database.
3. Controllare l'utilizzo non supportato:
   - Transazioni preparate – Eseguire il commit o il rollback di tutte le transazioni preparate prima di provare a eseguire un aggiornamento.

     È possibile utilizzare la seguente query per verificare che sull'istanza non siano presenti transazioni preparate aperte.

     ```
     SELECT count(*) FROM pg_catalog.pg_prepared_xacts;
     ```

   - Tipi di dati Reg* – Rimuovere tutti gli utilizzi dei tipi di dati reg* prima di tentare un aggiornamento. Ad eccezione di `regtype` e `regclass`, non è possibile aggiornare i tipi di dati reg*. L'utilità `pg_upgrade` non può preservare questo tipo di dati che sono utilizzati da per eseguire l'aggiornamento.

     Per verificare che non siano presenti utilizzi di tipi di dati reg* non supportati, utilizzare la query seguente per ogni database.

     ```
     SELECT count(*) FROM pg_catalog.pg_class c, pg_catalog.pg_namespace n,
      pg_catalog.pg_attribute a
       WHERE c.oid = a.attrelid
           AND NOT a.attisdropped
           AND a.atttypid IN ('pg_catalog.regproc'::pg_catalog.regtype,
                              'pg_catalog.regprocedure'::pg_catalog.regtype,
                              'pg_catalog.regoper'::pg_catalog.regtype,
                              'pg_catalog.regoperator'::pg_catalog.regtype,
                              'pg_catalog.regconfig'::pg_catalog.regtype,
                              'pg_catalog.regdictionary'::pg_catalog.regtype)
           AND c.relnamespace = n.oid
           AND n.nspname NOT IN ('pg_catalog', 'information_schema');
     ```

4. Gestione degli slot di replica logica: non è possibile eseguire un aggiornamento se il database dispone di slot di replica logica. Gli slot di replica logica vengono generalmente utilizzati per la migrazione AWS DMS e la replica di tabelle dal database a data lake, strumenti di BI e altre destinazioni. Prima di eseguire l'aggiornamento, assicurati di conoscere lo scopo di qualsiasi slot di replica logica in uso e verifica che sia corretto eliminarli. Se gli slot di replica logica sono ancora in uso, non devono essere eliminati e non è possibile procedere con l'aggiornamento.

   Se gli slot di replica logica non sono necessari, è possibile eliminarli utilizzando il seguente SQL:

   ```
   SELECT * FROM pg_replication_slots;
   SELECT pg_drop_replication_slot(slot_name);
   ```

   È inoltre necessario rimuovere gli slot nelle configurazioni di replica logica che utilizzano l'estensione `pglogical` per un corretto aggiornamento della versione principale. Per informazioni su come identificare e rimuovere gli slot creati utilizzando l'estensione `pglogical`, consulta Gestione degli slot di replica logica per RDS per PostgreSQL.
5. Gestione delle repliche di lettura: un aggiornamento di un'istanza database single-AZ o di una implementazione multi-AZ di un'istanza database aggiorna anche le repliche di lettura nella regione assieme all'istanza database primaria. Amazon RDS non aggiorna le repliche di lettura del cluster database multi-AZ.

   Non è possibile aggiornare le repliche di lettura separatamente. Se fosse consentito, si potrebbero verificare situazioni in cui i database primari e di replica hanno versioni principali PostgreSQL diverse. Tuttavia, gli aggiornamenti delle repliche di lettura potrebbero aumentare i tempi di inattività sull'istanza database primaria. Per impedire un aggiornamento della replica di lettura, promuovi la replica a un'istanza autonoma o eliminala prima di avviare il processo di aggiornamento.

   Il processo di aggiornamento ricrea il gruppo di parametri della replica di lettura in base al gruppo di parametri corrente della replica de lettura. Puoi applicare un gruppo di parametri personalizzato a una replica di lettura solo dopo il completamento dell'aggiornamento modificando la replica di lettura. Per ulteriori informazioni sulle repliche di lettura, consulta Utilizzo delle repliche di lettura per Amazon RDS per PostgreSQL.
6. Eseguire un backup – Si consiglia di eseguire un backup prima di eseguire un aggiornamento della versione principale in modo da avere un punto di ripristino noto per il database. Se il periodo di conservazione dei backup è maggiore di 0, il processo di aggiornamento crea snapshot database del database prima e dopo un aggiornamento. Per cambiare il periodo di conservazione dei

backup, consulta Modifica di un'istanza database Amazon RDS e the section called “Modifica di un cluster di database Multi-AZ”.

Per eseguire un backup manuale, consulta the section called “Creazione di uno snapshot DB per un'istanza DB Single-AZ” e the section called “Creazione di uno snapshot di un cluster di database Multi-AZ”.

7. Aggiorna determinate estensioni prima dell'aggiornamento della versione principale: se intendi ignorare una versione principale durante l'aggiornamento, è necessario aggiornare alcune estensioni prima di eseguire l’aggiornamento della versione principale. Ad esempio, l'aggiornamento dalle versioni 9.5.x o 9.6.x alla versione 11.x salta una versione principale. Le estensioni da aggiornare includono PostGIS e le relative estensioni per l'elaborazione dei dati spaziali.

   - `address_standardizer`
   - `address_standardizer_data_us`
   - `postgis_raster`
   - `postgis_tiger_geocoder`
   - `postgis_topology`

   Esegui il comando seguente per ogni estensione in uso:

   ```
   ALTER EXTENSION PostgreSQL-extension UPDATE TO 'new-version';
   ```

   Per ulteriori informazioni, consulta Aggiornamento estensioni PostgreSQL. Per ulteriori informazioni sull'aggiornamento di PostGIS, consulta Passaggio 6: Aggiornamento dell'estensione PostGIS.

8. Rimuovere alcune estensioni prima dell'aggiornamento alla versione principale – Un aggiornamento che ignora una versione principale per passare direttamente alla versione 11.x non supporta l'aggiornamento dell'estensione `pgRouting`. L'aggiornamento dalle versioni 9.4.x, 9.5.x o 9.6.x alle versioni 11.x ignora una versione principale. È possibile rimuovere senza conseguenze l'estensione `pgRouting` e reinstallarla con una versione compatibile dopo l'aggiornamento. Per le versioni dell'estensione aggiornabili, consultare Versioni con estensione PostgreSQL supportate.

   Le estensioni `tsearch2` e `chkpass` non sono più supportate per PostgreSQL 11 o versioni successive. Se si esegue l'aggiornamento alla versione 11.x, rimuovere le estensioni `tsearch2` e `chkpass` prima dell'aggiornamento.

9. Eliminare tipi di dati sconosciuti – Eliminare i tipi di dati `unknown` a seconda della versione di destinazione.

   PostgreSQL versione 10 ha smesso di supportare il tipo di dati `unknown`. Se un database versione 9.6 utilizza il tipo di dati `unknown`, un aggiornamento a una versione 10 mostra un messaggio di errore del tipo seguente:

   ```
   Database instance is in a state that cannot be upgraded: PreUpgrade checks failed:
   The instance could not be upgraded because the 'unknown' data type is used in user
    tables.
   Please remove all usages of the 'unknown' data type and try again."
   ```

   Per trovare il tipo di dati `unknown` nel database in modo da poter rimuovere la colonna danneggiata o modificarla in un tipo di dati supportato, utilizza il seguente codice SQL:

   ```
   SELECT DISTINCT data_type FROM information_schema.columns WHERE data_type ILIKE
    'unknown';
   ```

10. Eseguire un test di aggiornamento – Si consiglia fortemente di testare l'aggiornamento alla versione principale su un duplicato del database di produzione prima di provare l'aggiornamento sul database effettivo. È possibile monitorare i piani di esecuzione sul database di test duplicato per eventuali regressioni del piano di esecuzione e valutarne le prestazioni. Per creare un'istanza di test duplicata, è possibile ripristinare il database da uno snapshot recente o point-in-time ripristinare il database all'ultima data di ripristino.

    Per ulteriori informazioni, consulta the section called “Ripristino da uno snapshot” o the section called “oint-in-time Ripristino P”. Per i cluster multi-AZ, consulta the section called “Ripristino da uno snapshot a un cluster di database Multi-AZ” o the section called “Ripristino di un cluster di database Multi-AZ a un determinato momento”.

    Per i dettagli sull'esecuzione dell'aggiornamento, consulta the section called “Aggiornamento manuale della versione del motore”.

    Durante l'aggiornamento di un database dalla versione 9.6 alla versione 10, tieni presente che in PostgreSQL 10 le query parallele sono abilitate per impostazione predefinita.
    Puoi testare l'impatto del parallelismo prima dell'aggiornamento modificando il parametro `max_parallel_workers_per_gather` sul tuo database di test impostandolo su 2.

> **Note**
>
> Il valore di default per il parametro `max_parallel_workers_per_gather` nel gruppo parametri del database `default.postgresql10` è 2.

Per ulteriori informazioni, consulta Parallel Query (Query parallela) nella documentazione di PostgreSQL. Per disabilitare il parallelismo sulla versione 10, imposta il parametro `max_parallel_workers_per_gather` su 0.

Durante l'aggiornamento della versione principale, i database `public` e `template1`, nonché lo schema `public` in ciascun database vengono rinominati temporaneamente. Questi oggetti vengono riportati nei log con il loro nome originale e con l'aggiunta di una stringa casuale. La stringa viene aggiunta in modo che, durante l'aggiornamento alla versione principale, vengano preservate le impostazioni personalizzate come `locale` e `owner`. Al termine dell'aggiornamento gli oggetti vengono rinominati con i loro nomi originali.

> **Note**
>
> Durante il processo di aggiornamento della versione principale, non è possibile eseguire un point-in-time ripristino dell'istanza DB o del cluster DB Multi-AZ. Dopo aver eseguito l'aggiornamento, Amazon RDS effettua un backup automatico del database. È possibile eseguire un point-in-time ripristino ai tempi precedenti all'inizio dell'aggiornamento e dopo il completamento del backup automatico del database.

11. Se un aggiornamento restituisce errori durante la procedura di controllo preliminare, risolvere i problemi – Durante l'aggiornamento alla versione principale, Amazon RDS for PostgreSQL esegue una procedura di controllo preliminare per identificare eventuali problemi che potrebbero impedire l'aggiornamento. La procedura di controllo preliminare verifica tutte le condizioni potenzialmente incompatibili in tutti i database dell'istanza.

    Se il controllo preliminare rileva un problema, crea un evento di log che indica che il controllo preliminare dell'aggiornamento non è riuscito. I dettagli del processo di controllo preliminare si trovano in un log dell'aggiornamento denominato `pg_upgrade_precheck.log` per tutti i database. Amazon RDS aggiunge un timestamp al nome file. Per ulteriori informazioni sulla visualizzazione dei log, consultare Monitoraggio dei file di log di Amazon RDS.

Se un aggiornamento della replica di lettura non riesce al controllo preliminare, la replica della replica di lettura non riuscita viene interrotta e la replica di lettura viene messa nello stato terminato. Elimina la replica di lettura e ricrea una nuova replica di lettura in base all'istanza primaria aggiornata.

Risolvere tutti i problemi rilevati nel log di controllo preliminare, quindi riprovare l'aggiornamento alla versione principale. Nell'esempio seguente viene mostrato un esempio di log di controllo preliminare.

```
-------------------------------------------------------------------------
Upgrade could not be run on Wed Apr 4 18:30:52 2018
-------------------------------------------------------------------------
The instance could not be upgraded from 9.6.11 to 10.6 for the following reasons.
Please take appropriate action on databases that have usage incompatible with the
 requested major engine version upgrade and try the upgrade again.

* There are uncommitted prepared transactions. Please commit or rollback all prepared
 transactions.* One or more role names start with 'pg_'. Rename all role names that
 start with 'pg_'.

* The following issues in the database 'my"million$"db' need to be corrected before
 upgrading:** The ["line","reg*"] data types are used in user tables. Remove all
 usage of these data types.
** The database name contains characters that are not supported by RDS for
 PostgreSQL. Rename the database.
** The database has extensions installed that are not supported on the target
 database version. Drop the following extensions from your database: ["tsearch2"].

* The following issues in the database 'mydb' need to be corrected before
 upgrading:** The database has views or materialized views that depend on
 'pg_stat_activity'. Drop the views.
```

12. Se un aggiornamento della replica di lettura non riesce durante l'aggiornamento del database, risolvi il problema: lo stato di una replica di lettura non riuscita viene impostato su `incompatible-restore` e la replica viene terminata sul database. Elimina la replica di lettura e ricrea una nuova replica di lettura in base all'istanza primaria aggiornata.

> **Note**
>
> Amazon RDS non aggiorna le repliche di lettura per i cluster database multi-AZ. Se si esegue un aggiornamento di una versione principale su un cluster DB Multi-AZ, lo stato di replica delle relative repliche di lettura diventa terminato.

L'aggiornamento della replica di lettura potrebbe non riuscire per i seguenti motivi:

- Non è stato in grado di recuperare il ritardo con l'istanza database primaria anche dopo un tempo di attesa.
- Il terminale o lo stato del ciclo di vita è incompatibile, come ad esempio spazio di storage esaurito, ripristino incompatibile e così via.
- Quando l'aggiornamento dell'istanza database principale è stato avviato, nella replica di lettura era in esecuzione un aggiornamento di versione secondaria separata.
- La replica di lettura ha utilizzato parametri incompatibili.
- La replica di lettura non è in grado di comunicare con l'istanza database primaria per sincronizzare la cartella dati.

13 Aggiornamento del database di produzione: se l'esecuzione dell'aggiornamento della versione principale ha esito positivo, dovresti essere in grado di eseguire l'aggiornamento del database di produzione senza problemi. Per ulteriori informazioni, consulta Aggiornamento manuale della versione del motore.

14 Eseguire l'operazione `ANALYZE` per aggiornare la tabella `pg_statistic`. È necessario farlo per ogni database su tutti i database PostgreSQL. Le statistiche di ottimizzazione non vengono trasferite durante un aggiornamento della versione principale, quindi è necessario rigenerare tutte le statistiche per evitare problemi di prestazioni. Esegui il comando senza parametri per generare statistiche per tutte le tabelle regolari del database corrente, come segue:

```
ANALYZE VERBOSE;
```

Il flag `VERBOSE` è facoltativo, ma usandolo viene mostrato lo stato di avanzamento. Per ulteriori informazioni, consulta ANALYZE nella documentazione di PostgreSQL.

> **Note**
>
> Esegui ANALYZE sul tuo sistema dopo l'aggiornamento per evitare problemi di prestazioni.

Al termine dell'aggiornamento alla versione principale, è consigliabile:

- Un aggiornamento PostgreSQL non aggiorna alcuna estensione PostgreSQL. Per aggiornare le estensioni, consulta Aggiornamento estensioni PostgreSQL.
- Facoltativamente, utilizza Amazon RDS per visualizzare i due log generati dalla utilità `pg_upgrade`. Questi sono `pg_upgrade_internal.log` e `pg_upgrade_server.log`. Amazon RDS accoda un timestamp al nome file per questi log. Puoi visualizzare questi log come qualsiasi altro log. Per ulteriori informazioni, consulta Monitoraggio dei file di log di Amazon RDS.

  Puoi anche caricare i log di aggiornamento su Amazon CloudWatch Logs. Per ulteriori informazioni, consulta Pubblicazione dei log PostgreSQL su Amazon Logs CloudWatch .
- Per verificare che tutto funzioni come previsto, testa l'applicazione sul database aggiornato con un carico di lavoro analogo. Dopo la verifica dell'aggiornamento è possibile eliminare l'istanza di test.

Aggiornamenti a versioni secondarie automatiche per PostgreSQL

Se abiliti l'opzione Aggiornamento automatico versione secondaria quando crei o modifichi un'istanza database o un cluster database multi-AZ, il database viene aggiornato automaticamente.

Per ogni versione principale di RDS per PostgreSQL, una versione minore viene designata da RDS come versione di aggiornamento automatico. Una volta che una versione secondaria è stata testata e approvata da Amazon RDS, l'aggiornamento della versione secondaria avviene automaticamente nel corso della finestra di manutenzione. RDS non imposta mai automaticamente le nuove release secondarie come versione di aggiornamento automatico. Prima che RDS indichi una versione di aggiornamento automatico più recente, vengono considerati diversi livelli di valutazione, quali:

- Problemi di sicurezza noti
- Bug nella versione della community di PostgreSQL
- Stabilità generale del parco istanze da quando la versione secondaria è stata rilasciata

È possibile utilizzare il seguente AWS CLI comando per determinare la versione di destinazione dell'aggiornamento secondario automatico corrente per una versione secondaria di PostgreSQL specificata in una specifica. Regione AWS

PerLinux, o: macOS Unix

```
aws rds describe-db-engine-versions \
--engine postgres \
--engine-version minor-version \
--region region \
--query "DBEngineVersions[*].ValidUpgradeTarget[*].
{AutoUpgrade:AutoUpgrade,EngineVersion:EngineVersion}" \
--output text
```

Per Windows:

```
aws rds describe-db-engine-versions ^
--engine postgres ^
--engine-version minor-version ^
--region region ^
--query "DBEngineVersions[*].ValidUpgradeTarget[*].
{AutoUpgrade:AutoUpgrade,EngineVersion:EngineVersion}" ^
--output text
```

Ad esempio, il AWS CLI comando seguente determina l'obiettivo di aggiornamento secondario automatico per la versione secondaria di PostgreSQL 12.13 negli Stati Uniti orientali (Ohio) (us-east-2). Regione AWS

Per, oUnix: Linux macOS

```
aws rds describe-db-engine-versions \
--engine postgres \
--engine-version 12.13 \
--region us-east-2 \
--query "DBEngineVersions[*].ValidUpgradeTarget[*].
{AutoUpgrade:AutoUpgrade,EngineVersion:EngineVersion}" \
--output table
```

Per Windows:

```
aws rds describe-db-engine-versions ^
```

```
--engine postgres ^
--engine-version 12.13 ^
--region us-east-2 ^
--query "DBEngineVersions[*].ValidUpgradeTarget[*].
{AutoUpgrade:AutoUpgrade,EngineVersion:EngineVersion}" ^
--output table
```

L'output è simile a quello riportato di seguito.

```
---------------------------------
|    DescribeDBEngineVersions    |
+--------------+-----------------+
|  AutoUpgrade |  EngineVersion  |
+--------------+-----------------+
|  True        |  12.14          |
|  False       |  12.15          |
|  False       |  13.9           |
|  False       |  13.10          |
|  False       |  13.11          |
|  False       |  14.6           |
+--------------+-----------------+
```

In questo esempio, il valore `AutoUpgrade` è `True` per PostgreSQL versione 12.14. Quindi, la destinazione dell'aggiornamento secondario automatico è PostgreSQL versione 12.14, che è evidenziata nell'output.

Un database PostgreSQL viene aggiornato automaticamente durante la finestra di manutenzione se vengono soddisfatti i seguenti criteri:

- Nel database è abilitata l'opzione Aggiornamento automatico versione secondaria.
- Il database esegue una versione del motore di database secondaria rispetto a una versione secondaria automatica dell'aggiornamento corrente.

Per ulteriori informazioni, consulta Aggiornamento automatico della versione secondaria del motore.

> **Note**
>
> Un aggiornamento PostgreSQL non aggiorna alcuna estensione PostgreSQL. Per aggiornare le estensioni, consulta Aggiornamento estensioni PostgreSQL.

Aggiornamento estensioni PostgreSQL

Un aggiornamento al motore PostgreSQL non aggiorna alcuna estensione PostgreSQL. Per aggiornare un'estensione dopo un aggiornamento a una versione, utilizza il comando `ALTER EXTENSION UPDATE`.

> **Note**
>
> Per informazioni sull'aggiornamento dell'estensione PostGIS, consulta Gestione dei dati spaziali con estensione PostGIS (Passaggio 6: Aggiornamento dell'estensione PostGIS). Per aggiornare l'estensione `pg_repack`, rimuovi l'estensione e quindi crea la nuova versione nel database aggiornato. Per ulteriori informazioni, consulta pg_repack installation nella documentazione `pg_repack`.

Per aggiornare un'estensione, utilizza il comando seguente.

```
ALTER EXTENSION extension_name UPDATE TO 'new_version';
```

Per l'elenco delle versioni supportate delle estensioni PostgreSQL, consulta Versioni con estensione PostgreSQL supportate.

Per elencare le estensioni attualmente installate, usa il catalogo PostgreSQL pg_extension nel seguente comando.

```
SELECT * FROM pg_extension;
```

Per visualizzare l'elenco delle versioni delle estensioni specifiche disponibili per l'installazione, utilizza la vista PostgreSQL pg_available_extension_versions nel seguente comando.

```
SELECT * FROM pg_available_extension_versions;
```

Aggiornamento di una versione del motore di snapshot database PostgreSQL

Con Amazon RDS puoi creare uno snapshot DB del volume di storage dell'istanza database PostgreSQL. Quando crei uno snapshot DB, lo snapshot si basa sulla versione del motore utilizzata dall'istanza Amazon RDS. Oltre ad aggiornare la versione del motore DB dell'istanza database, puoi anche aggiornare la versione del motore per gli snapshot DB.

Dopo aver ripristinato uno snapshot DB aggiornato a una nuova versione del motore, verificare che l'aggiornamento abbia avuto esito positivo. Per maggiori informazioni sull'aggiornamento di una versione principale, consultare Aggiornamento del motore del database PostgreSQL per Amazon RDS. Per informazioni su come ripristinare uno snapshot DB, consulta Ripristino da uno snapshot database.

Puoi aggiornare gli snapshot database manuali crittografati e non crittografati.

Per l'elenco delle versioni di motore disponibili per aggiornare uno snapshot database, consultare Aggiornamento del motore di database PostgreSQL per Amazon RDS.

> **Note**
>
> Non è possibile aggiornare gli snapshot DB automatizzati creati durante il processo di backup automatico.

Console

Per aggiornare uno snapshot DB

1. Accedi alla AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, selezionare Snapshots (Snapshot).
3. Scegli la snapshot da usare per l'aggiornamento.
4. Per Actions (Operazioni), scegliere Upgrade snapshot (Aggiorna snapshot). Viene visualizzata la pagina Upgrade snapshot (Aggiorna snapshot).
5. Scegli New engine version (Nuova versione del motore) per eseguire l'aggiornamento.
6. Scegliere Save changes (Salva modifiche) per aggiornare lo snapshot.

Durante il processo di aggiornamento, tutte le operazioni dello snapshot sono disabilitate per lo snapshot database. Inoltre, lo stato dello snapshot DB cambia da available (disponibile) a upgrading (in aggiornamento), quindi diventa active (attivo) al completamento. Se lo snapshot DB non può essere aggiornato a causa di problemi di corruzione, lo stato diventa unavailable (non disponibile). Non è possibile recuperare lo snapshot quando è in questo stato.

> **Note**
>
> Se l'aggiornamento dello snapshot fallisce, lo snapshot viene riportato allo stato originario con la versione iniziale.

AWS CLI

Per aggiornare uno snapshot DB a una nuova versione del motore di database, utilizzare il AWS CLI modify-db-snapshotcomando.

Parametri

- `--db-snapshot-identifier` – Identificatore dello snapshot DB da aggiornare. L'identificatore deve essere un Amazon Resource Name (ARN) univoco. Per ulteriori informazioni, consulta Utilizzo di Amazon Resource Name (ARN) in Amazon RDS.
- `--engine-version` – Versione del motore a cui aggiornare lo snapshot DB.

Example

Per LinuxmacOS, oUnix:

```
aws rds modify-db-snapshot \
    --db-snapshot-identifier my_db_snapshot \
    --engine-version new_version
```

Per Windows:

```
aws rds modify-db-snapshot ^
    --db-snapshot-identifier my_db_snapshot ^
    --engine-version new_version
```

API RDS

Per aggiornare uno snapshot DB a una nuova versione del motore di database, chiama l'operazione API Amazon RDS ModifyDBSnapshot.

- `DBSnapshotIdentifier` – Identificatore dello snapshot DB da aggiornare. L'identificatore deve essere un Amazon Resource Name (ARN) univoco. Per ulteriori informazioni, consulta Utilizzo di Amazon Resource Name (ARN) in Amazon RDS.
- `EngineVersion` – Versione del motore a cui aggiornare lo snapshot DB.

Utilizzo delle repliche di lettura per Amazon RDS per PostgreSQL

Puoi scalare le letture per le tue istanze DB Amazon RDS for PostgreSQL aggiungendo repliche di lettura alle istanze. Come altri motori di database Amazon RDS, RDS per PostgreSQL utilizza meccanismi di replica nativi di PostgreSQL per mantenere le repliche di lettura aggiornate in base alle modifiche sul DB di origine. Per informazioni generali sulle repliche di lettura e Amazon RDS, consulta Uso delle repliche di lettura dell'istanza database.

Questa sezione contiene informazioni specifiche sull'utilizzo delle repliche di lettura con RDS per PostgreSQL.

Decodifica logica su una replica di lettura

RDS per PostgreSQL supporta la replica logica dagli standby con PostgreSQL 16.1. Ciò consente di creare una decodifica logica da uno standby di sola lettura che riduce il carico sull'istanza DB principale. È possibile ottenere una maggiore disponibilità per le applicazioni che devono sincronizzare i dati su più sistemi. Questa funzionalità migliora le prestazioni del data warehouse e dell'analisi dei dati.

Inoltre, gli slot di replica su un determinato standby mantengono la promozione di tale standby a principale. Ciò significa che, in caso di failover di un'istanza DB primaria o di promozione di un'istanza di standby come nuova istanza principale, gli slot di replica persisteranno e i precedenti abbonati in standby non ne risentiranno.

Per creare una decodifica logica su una replica di lettura

1. Attiva la replica logica: per creare la decodifica logica in standby, è necessario attivare la replica logica sull'istanza DB di origine e sulla relativa replica fisica. Per ulteriori informazioni, consulta Configurazione delle repliche di lettura con PostgreSQL.

   - Per attivare la replica logica per un'istanza DB RDS for PostgreSQL appena creata, crea un nuovo gruppo di parametri DB personalizzato e imposta il parametro statico su. `rds.logical_replication` 1 Quindi, associa questo gruppo di parametri DB all'istanza DB di origine e alla relativa replica fisica di lettura. Per ulteriori informazioni, consulta Associazione di un gruppo di parametri database a un'istanza database.
   - Per attivare la replica logica per un'istanza DB RDS for PostgreSQL esistente: modifica il gruppo di parametri database personalizzati dell'istanza DB di origine e la relativa replica

di lettura fisica su cui impostare il parametro statico. `rds.logical_replication` 1 Per ulteriori informazioni, consulta Modifica di parametri in un gruppo di parametri del database.

> **Note**
>
> È necessario riavviare l'istanza DB per applicare queste modifiche ai parametri.

È possibile utilizzare la seguente query per verificare i valori per `wal_level` e `rds.logical_replication` sull'istanza DB di origine e la relativa replica fisica di lettura.

```
Postgres=>SELECT name,setting FROM pg_settings WHERE name IN
 ('wal_level','rds.logical_replication');

 name                    | setting
-------------------------+---------
 rds.logical_replication | on
 wal_level               | logical
(2 rows)
```

2. Crea una tabella nel database di origine: connettiti al database nell'istanza DB di origine. Per ulteriori informazioni, consulta Connessione a un'istanza database che esegua il motore di database di PostgreSQL.

   Utilizzate le seguenti query per creare una tabella nel database di origine e inserire valori:

   ```
   Postgres=>CREATE TABLE LR_test (a int PRIMARY KEY);
   CREATE TABLE
   ```

   ```
   Postgres=>INSERT INTO LR_test VALUES (generate_series(1,10000));
   INSERT 0 10000
   ```

3. Crea una pubblicazione per la tabella di origine: utilizza la seguente query per creare una pubblicazione per la tabella sull'istanza DB di origine.

   ```
   Postgres=>CREATE PUBLICATION testpub FOR TABLE LR_test;
   CREATE PUBLICATION
   ```

Utilizzate una query SELECT per verificare i dettagli della pubblicazione creata sia sull'istanza DB di origine che sull'istanza fisica di replica di lettura.

```
Postgres=>SELECT * from pg_publication;

oid    | pubname | pubowner | puballtables | pubinsert | pubupdate | pubdelete |
 pubtruncate | pubviaroot
-------+---------+----------+--------------+-----------+-----------+-----------
+-------------+------------
 16429 | testpub |    16413 | f            | t         | t         | t         | t
          | f
(1 row)
```

4. Crea una sottoscrizione da un'istanza di replica logica: crea un'altra istanza DB RDS per PostgreSQL come istanza di replica logica. Assicurati che il VPC sia configurato correttamente per garantire che questa istanza di replica logica possa accedere all'istanza di replica fisica di lettura. Per ulteriori informazioni, consulta VPC di Amazon VPC e Amazon RDS. Se l'istanza DB di origine è inattiva, potrebbero verificarsi problemi di connettività e l'istanza primaria non invia i dati in standby.

```
Postgres=>CREATE SUBSCRIPTION testsub CONNECTION 'host=Physical replica host name
 port=port
                dbname=source_db_name user=user password=password
 PUBLICATION testpub;
NOTICE:  created replication slot "testsub" on publisher
CREATE SUBSCRIPTION
```

```
Postgres=>CREATE TABLE LR_test (a int PRIMARY KEY);
CREATE TABLE
```

Utilizza una query SELECT per verificare i dettagli dell'abbonamento sull'istanza di replica logica.

```
Postgres=>SELECT oid,subname,subenabled,subslotname,subpublications FROM
 pg_subscription;

oid    | subname | subenabled | subslotname | subpublications
-------+---------+------------+-------------+-----------------
 16429 | testsub | t          | testsub     | {testpub}
(1 row)
postgres=> select count(*) from LR_test;
```

```
 count
-------
 10000
(1 row)
```

5. Controlla lo stato dello slot di replica logica: puoi vedere solo lo slot di replica fisica sull'istanza DB di origine.

```
Postgres=>select slot_name, slot_type, confirmed_flush_lsn from
 pg_replication_slots;

slot_name                                    | slot_type | confirmed_flush_lsn
---------------------------------------------+-----------+---------------------
 rds_us_west_2_db_dhqfsmo5wbbjqrn3m6b6ivdhu4 | physical  |
(1 row)
```

Tuttavia, sull'istanza di replica di lettura, è possibile vedere lo slot di replica logica e il `confirmed_flush_lsn` valore cambia man mano che l'applicazione utilizza attivamente le modifiche logiche.

```
Postgres=>select slot_name, slot_type, confirmed_flush_lsn from
 pg_replication_slots;

slot_name | slot_type | confirmed_flush_lsn
----------+-----------+---------------------
 testsub   | logical   | 0/500002F0
(1 row)
```

```
Postgres=>select slot_name, slot_type, confirmed_flush_lsn from
 pg_replication_slots;

slot_name | slot_type | confirmed_flush_lsn
----------+-----------+---------------------
 testsub   | logical   | 0/5413F5C0
(1 row)
```

Limitazioni per le repliche di lettura con PostgreSQL

Di seguito sono elencate le limitazioni per le repliche di lettura PostgreSQL:

> **Note**
>
> Una replica di lettura per un'istanza DB RDS for PostgreSQL Multi-AZ e Single-AZ che esegue PostgreSQL versione 12 e precedenti, si riavvia automaticamente per applicare la rotazione della password durante la finestra di manutenzione da 60 a 90 giorni.

- Le repliche di lettura PostgreSQL sono di sola lettura. Sebbene una replica di lettura non sia un'istanza database scrivibile, è possibile promuoverla a un'istanza database RDS per PostgreSQL autonoma. Tuttavia, il processo non è reversibile.
- Non è possibile creare una replica di lettura da un'altra replica di lettura se l'istanza database RDS per PostgreSQL esegue una versione di PostgreSQL precedente alla 14.1. RDS per PostgreSQL supporta solo le repliche di lettura a cascata su RDS per PostgreSQL versione 14.1 e versioni successive. Per ulteriori informazioni, consulta Utilizzo di repliche di lettura a cascata con RDS per PostgreSQL.
- Se promuovi una replica di lettura PostgreSQL, questa diventa un'istanza database scrivibile. Smette di ricevere i file WAL (write-ahead log) da un'istanza database di origine e non è più un'istanza di sola lettura. È possibile creare nuove repliche di lettura dall'istanza database promossa come avviene per qualsiasi istanza database RDS per PostgreSQL. Per ulteriori informazioni, consulta Promozione di una replica di lettura a istanza database standalone.
- Se promuovi una replica di lettura PostgreSQL dall'interno di una catena di replica (una serie di repliche di lettura a cascata), tutte le repliche di lettura downstream esistenti continuano a ricevere automaticamente i file WAL dall'istanza promossa. Per ulteriori informazioni, consulta Utilizzo di repliche di lettura a cascata con RDS per PostgreSQL.
- Se sull'istanza database di origine non sono in esecuzione transazioni utente, la replica di lettura PostgreSQL associata è caratterizzata da un ritardo di replica fino a cinque minuti. Il ritardo di replica viene calcolato come `currentTime - lastCommitedTransactionTimestamp`, il che significa che quando non viene elaborata alcuna transazione, il valore del ritardo di replica aumenta per un periodo di tempo fino a quando il segmento WAL (write-ahead log) cambia. Per impostazione predefinita, RDS per PostgreSQL cambia il segmento WAL ogni 5 minuti, il che comporta un record di transazione e una riduzione del ritardo segnalato.
- Non è possibile attivare i backup automatici per le repliche di lettura PostgreSQL per le versioni precedenti alla 14.1 di RDS per PostgreSQL. I backup automatici per le repliche di lettura sono supportati solo per RDS per PostgreSQL 14.1 e versioni successive. Per RDS per PostgreSQL 13 e versioni precedenti, crea uno snapshot da una replica di lettura se si desidera creare un backup da tale snapshot.

- Il ripristino P (PITR) oint-in-time non è supportato per le repliche di lettura. È possibile utilizzare ripristino point-in-time (PITR) solo con un'istanza primaria (istanza di scrittura), non con una replica di lettura. Per ulteriori informazioni, consulta Ripristino a un'ora specifica per un'istanza database.

Configurazione delle repliche di lettura con PostgreSQL

RDS per PostgreSQL utilizza la replica di streaming nativa PostgreSQL per creare una copia di sola lettura di un'istanza database di origine. Questa istanza database di replica di lettura è una replica fisica creata in modo asincrono dell'istanza database di origine. Viene creata da una speciale connessione che trasmette i dati Write Ahead Log (WAL) dall'istanza database di origine alla replica di lettura. Per ulteriori informazioni, consulta la sezione relativa alla replica in streaming nella documentazione di PostgreSQL.

PostgreSQL trasmette in streaming in modo asincrono le modifiche al database a questa connessione sicura man mano che vengono effettuate sull'istanza database di origine. È possibile crittografare le comunicazioni dalle applicazioni client all'istanza database di origine o a qualsiasi replica di lettura impostando il parametro `ssl` su `1`. Per ulteriori informazioni, consulta Utilizzo del protocollo SSL con un'istanza database PostgreSQL.

PostgreSQL utilizza un ruolo di replica per eseguire la replica in streaming. Il ruolo presenta dei privilegi ma non può essere utilizzato per modificare i dati. PostgreSQL utilizza un unico processo per la gestione della replica.

È possibile creare una replica di lettura PostgreSQL senza alcun impatto sulle operazioni o sugli utenti dell'istanza database di origine. Amazon RDS imposta i parametri e le autorizzazioni necessari per l'istanza database di origine e la replica di lettura senza ripercussioni sul servizio. Viene acquisito uno snapshot dell'istanza database di origine e tale snapshot viene utilizzato per creare la replica di lettura. Se si elimina la replica di lettura in un secondo momento, non si verificherà alcuna interruzione.

È possibile creare fino a 15 repliche di lettura da un'istanza database di origine nella stessa regione. A partire da RDS per PostgreSQL 14.1, è anche possibile creare fino a tre livelli di replica di lettura in una catena (cascata) da un'istanza database di origine. Per ulteriori informazioni, consulta Utilizzo di repliche di lettura a cascata con RDS per PostgreSQL. In ogni caso, l'istanza database di origine deve disporre di backup automatici configurati. A questo scopo, imposta il periodo di conservazione del backup sull'istanza database su un valore diverso da zero. Per ulteriori informazioni, consulta Creazione di una replica di lettura.

Puoi creare repliche di lettura per la tua istanza DB RDS per PostgreSQL nella stessa istanza DB di origine. Regione AWS Questo processo è noto come replica nella regione. Puoi anche creare repliche di lettura in un'istanza DB diversa dall'istanza DB di origine Regioni AWS . Questo processo è noto come replica tra regioni. Per informazioni sull'impostazione delle repliche di lettura tra regioni, consulta Creazione di una replica di lettura in un altro Regione AWS. I vari meccanismi che supportano il processo di replica "in regione" e "tra regioni" variano leggermente a seconda della versione di RDS per PostgreSQL, come spiegato in Funzionamento della replica in streaming per diverse versioni di RDS per PostgreSQL.

Per un efficace funzionamento della replica, ciascuna replica di lettura dovrebbe avere la stessa quantità di risorse di calcolo e storage dell'istanza database di origine. Se si dimensiona l'istanza database di origine, verifica di dimensionare anche le repliche di lettura.

Amazon RDS sostituisce qualsiasi parametro incompatibile in una replica di lettura che impedisca l'avvio della replica di lettura. Ad esempio, supponiamo che il valore del parametro `max_connections` sull'istanza database di origine sia superiore a quello sulla replica di lettura. In questo caso, Amazon RDS aggiorna il parametro sulla replica di lettura in modo che il valore coincida con quello sull'istanza database di origine.

Le repliche di lettura di RDS per PostgreSQL hanno accesso a database esterni disponibili tramite wrapper di dati esteri (FDW) sull'istanza database di origine. Ad esempio, supponiamo che l'istanza database RDS per PostgreSQL stia utilizzando il wrapper `mysql_fdw` per accedere ai dati da RDS per MySQL. In questo caso, anche le repliche di lettura possono accedere a tali dati. Altri FDW supportati includono `oracle_fdw`, `postgres_fdw` e `tds_fdw`. Per ulteriori informazioni, consulta Utilizzo dei wrapper di dati esterni supportati per Amazon RDS for PostgreSQL.

Utilizzo di repliche di lettura di RDS per PostgreSQL con configurazioni Multi-AZ

È possibile creare una replica di lettura da un'istanza database Single-AZ o Multi-AZ. Puoi usare implementazioni Multi-AZ per migliorare la durabilità e la disponibilità di dati critici con una replica in standby. Una replica in standby è una replica di lettura dedicata che può assumere il carico di lavoro se si verifica il failover del database di origine. Non è possibile utilizzare una replica in standby per gestire il traffico di lettura. Puoi tuttavia creare repliche di lettura da istanze database Multi-AZ con traffico elevato per l'offload di query di sola lettura. Per ulteriori informazioni sulle implementazioni Multi-AZ, consultare Implementazioni dell'istanza database Multi-AZ.

Se viene eseguito il failover dell'istanza database di origine di un'implementazione Multi-AZ sull'istanza in standby, tutte le repliche di lettura associate passeranno a usare l'istanza in standby

(ora primaria) come origine della replica. Potrebbe essere necessario riavviare le repliche di lettura, a seconda della versione RDS per PostgreSQL, come segue:

- PostgreSQL 13 e versioni successive: il riavvio non è obbligatorio. Le repliche di lettura vengono automaticamente sincronizzate con il nuovo database primario. Tuttavia, in alcuni casi l'applicazione client potrebbe memorizzare nella cache i dettagli DNS (Domain Name Service) per le repliche di lettura. In tal caso, imposta il valore time-to-live (TTL) su un valore inferiore a 30 secondi. In questo modo si impedisce alla replica di lettura di mantenere un indirizzo IP obsoleto e pertanto si impedisce la sincronizzazione con il nuovo database primario. Per ulteriori informazioni su questa e altre best practice, consulta Linee guida operative di base per Amazon RDS.
- PostgreSQL 12 e tutte le versioni precedenti: le repliche di lettura si riavviano automaticamente dopo un failover della replica in standby perché la replica in standby (ora principale) ha un indirizzo IP e un nome di istanza diversi. Il riavvio sincronizza la replica di lettura con il nuovo database primario.

Per ulteriori informazioni sul failover, consulta Processo di failover per Amazon RDS. Per ulteriori informazioni su come le repliche di lettura funzionano in una implementazione Multi-AZ, consultare Uso delle repliche di lettura dell'istanza database.

Per garantire il supporto del failover per una replica di lettura, puoi creare una replica di lettura come un'istanza database Multi-AZ. Amazon RDS crea una replica in standby della replica in un'altra zona di disponibilità (AZ). La creazione della replica di lettura come un'istanza database Multi-AZ non dipende dal fatto che il database di origine sia un'istanza database Multi-AZ.

Utilizzo di repliche di lettura a cascata con RDS per PostgreSQL

A partire dalla versione 14.1, RDS per PostgreSQL supporta le repliche di lettura a cascata. Con le repliche di lettura a cascata, puoi dimensionare le letture senza aggiungere sovraccarico all'istanza database RDS per PostgreSQL di origine. Gli aggiornamenti del registro WAL non vengono inviati dall'istanza database di origine a ciascuna replica di lettura. Invece, ogni replica di lettura in una serie a cascata invia gli aggiornamenti del registro WAL alla successiva replica di lettura della serie. Questo riduce il carico sull'istanza database di origine.

Con le repliche di lettura a cascata, l'istanza database RDS per PostgreSQL invia i dati WAL alla prima replica di lettura della catena. La replica di lettura invia quindi i dati WAL alla seconda replica della catena e così via. Il risultato finale è che tutte le repliche di lettura nella catena includono le modifiche dall'istanza database RDS per PostgreSQL, ma senza sovraccaricare esclusivamente l'istanza database di origine.

È possibile creare una serie di fino a tre repliche di lettura in una catena da un'istanza database RDS per PostgreSQL di origine. Ad esempio, supponiamo di avere un'istanza database RDS per PostgreSQL 14.1, `rpg-db-main`. Puoi eseguire le operazioni indicate di seguito:

- A partire da `rpg-db-main`, crea la prima replica di lettura nella catena, `read-replica-1`.
- Da `read-replica-1`, crea quindi la successiva replica di lettura nella catena, `read-replica-2`.
- Da `read-replica-2`, crea infine la terza replica di lettura nella catena, `read-replica-3`.

Non è possibile creare un'altra replica di lettura oltre la terza replica di lettura a cascata nella serie per `rpg-db-main`. Una serie completa di istanze da un'istanza database RDS per PostgreSQL di origine fino alla fine di una serie di repliche di lettura a cascata può essere composta al massimo da quattro istanze database.

Affinché le repliche di lettura a cascata funzionino, attiva i backup automatici su RDS per PostgreSQL. Crea prima la replica di lettura e quindi attiva i backup automatici sull'istanza database RDS per PostgreSQL. Il processo è lo stesso valido per gli altri motori di database Amazon RDS. Per ulteriori informazioni, consulta Creazione di una replica di lettura.

Come per qualsiasi replica di lettura, puoi promuovere una replica di lettura appartenente a una cascata. La promozione di una replica di lettura all'interno di una catena di repliche di lettura rimuove la replica dalla catena. Ad esempio, supponiamo che tu voglia spostare parte del carico di lavoro fuori dall'istanza database `rpg-db-main` in una nuova istanza usata solo dal reparto contabile. Facendo riferimento alla catena di tre repliche di lettura dell'esempio, decidi di promuovere `read-replica-2`. La catena verrà modificata come segue:

- La promozione `read-replica-2` rimuove l'istanza dalla catena di replica.
  - Ora è un'istanza database completa di lettura/scrittura.
  - Continua a replicare su `read-replica-3`, proprio come prima della promozione.
- L'istanza `rpg-db-main` continua a venire replicata su `read-replica-1`.

Per ulteriori informazioni sulla promozione delle repliche di lettura, consulta Promozione di una replica di lettura a istanza database standalone.

> **Note**
>
> Per le repliche di lettura a cascata, RDS per PostgreSQL supporta 15 repliche di lettura per ogni istanza database di origine al primo livello di replica e 5 repliche di lettura per ogni istanza database di origine al secondo e al terzo livello di replica.

Funzionamento della replica in streaming per diverse versioni di RDS per PostgreSQL

Come illustrato in Configurazione delle repliche di lettura con PostgreSQL, RDS per PostgreSQL utilizza il protocollo di replica in streaming nativo di PostgreSQL per inviare dati WAL dall'istanza database di origine. Invia i dati WAL di origine per leggere le repliche sia per le repliche di lettura nella regione che tra regioni. Con la versione 9.4, PostgreSQL ha introdotto gli slot di replica fisica come meccanismo di supporto per il processo di replica.

Uno slot di replica fisica impedisce a un'istanza database di origine di rimuovere i dati WAL prima che vengano consumati da tutte le repliche di lettura. Ogni replica di lettura ha un proprio slot fisico sull'istanza database di origine. Lo slot tiene traccia dei dati WAL più vecchi (per numero di sequenza logica, LSN) che potrebbero essere necessari per la replica. Dopo che tutti gli slot e le connessioni di database sono progrediti oltre un determinato WAL (LSN), il numero di sequenza logica (LSN) diventa un candidato per la rimozione al checkpoint successivo.

Amazon RDS utilizza Amazon S3 per archiviare i dati WAL. Per le repliche di lettura nella regione, è possibile utilizzare questi dati archiviati per recuperare la replica di lettura quando necessario. Un esempio di quando è possibile farlo è se la connessione tra database di origine e replica di lettura viene interrotta per qualsiasi motivo.

Nella tabella seguente è possibile trovare un riepilogo delle differenze tra le versioni di PostgreSQL e i meccanismi di supporto per la replica nella regione e tra regioni utilizzata da RDS per PostgreSQL.

Nella regione	Tra regioni
PostgreSQL 14.1 and higher versions	
• Slot di replica • Archivio Amazon S3	• Slot di replica

Nella regione	Tra regioni
PostgreSQL 13 and lower versions	
• Archivio Amazon S3	• Slot di replica

Per ulteriori informazioni, consulta Monitoraggio e ottimizzazione del processo di replica.

Informazioni sui parametri di controllo della replica PostgreSQL

I seguenti parametri influenzano il processo di replica e determinano il modo in cui le repliche di lettura restano aggiornate con l'istanza database di origine:

max_wal_senders

Il parametro `max_wal_senders` specifica il numero massimo di connessioni che l'istanza database di origine può supportare contemporaneamente sul protocollo di replica in streaming. Il valore predefinito per RDS per PostgreSQL 13 e versioni successive è 20. Questo parametro deve essere impostato su un valore leggermente più alto del numero effettivo di repliche di lettura. Se questo parametro è impostato su un valore troppo basso per il numero di repliche di lettura, la replica viene interrotta.

Per ulteriori informazioni, consulta la sezione relativa al parametro max_wal_senders nella documentazione di PostgreSQL.

wal_keep_segments

Il parametro `wal_keep_segments` specifica il numero di file WAL (write-ahead log) conservati dall'istanza database di origine nella directory `pg_wal`. L'impostazione predefinita è 32.

Se il parametro `wal_keep_segments` non è impostato su un valore abbastanza grande per l'implementazione, una replica di lettura può avere un ritardo tale che la replica di streaming si arresta. In questo caso, Amazon RDS genera un errore di replica e inizia il ripristino sulla replica di lettura. A tale scopo, riproduce i dati WAL archiviati dell'istanza database di origine da Amazon S3. Il processo di ripristino continua finché la replica di lettura non avrà recuperato abbastanza per continuare la replica di streaming. È possibile vedere questo processo in esecuzione come viene acquisito dal registro PostgreSQL in Esempio: come ripristinare una replica di lettura dalle interruzioni della replica.

> **Note**
>
> In PostgreSQL versione 13, il parametro `wal_keep_segments` è denominato `wal_keep_size`. Ha lo stesso scopo di `wal_keep_segments`, ma il suo valore predefinito è espresso in megabyte (MB) (2048 MB) anziché in numero di file. Per ulteriori informazioni, consulta la sezione relativa ai parametri wal_keep_segments e wal_keep_size nella documentazione di PostgreSQL.

max_slot_wal_keep_size

Il parametro `max_slot_wal_keep_size` controlla la quantità di dati WAL che l'istanza database RDS per PostgreSQL conserva nella directory `pg_wal` per servire gli slot. Questo parametro viene utilizzato per le configurazioni che utilizzano gli slot di replica. Il valore predefinito per questo parametro è `-1`, ovvero non esiste alcun limite per la quantità di dati WAL conservati nell'istanza database di origine. Per informazioni sul monitoraggio degli slot di replica, consulta Monitoraggio degli slot di replica per l'istanza database RDS per PostgreSQL.

Per ulteriori informazioni su questo parametro, consulta la sezione max_slot_wal_keep_size nella documentazione di PostgreSQL.

Se si interrompe il flusso WAL che fornisce i dati a una replica di lettura, PostgreSQL passa alla modalità di ripristino. Ripristina la replica di lettura utilizzando i dati WAL archiviati da Amazon S3 o utilizzando i dati WAL associati allo slot di replica. Al termine di questo processo, PostgreSQL tenta di ristabilire la replica in streaming.

Esempio: come ripristinare una replica di lettura dalle interruzioni della replica

Nell'esempio seguente sono disponibili i dettagli del registro che dimostrano il processo di ripristino per una replica di lettura. L'esempio proviene da un'istanza DB RDS per PostgreSQL che esegue PostgreSQL versione 12.9 nello Regione AWS stesso database di origine, quindi non vengono utilizzati slot di replica. Il processo di ripristino è lo stesso valido per le altre istanze database RDS per PostgreSQL che eseguono la versione precedente alla 14.1 di PostgreSQL con repliche di lettura nella regione.

Quando la replica di lettura perde il contatto con l'istanza database di origine, Amazon RDS registra il problema nel registro sotto forma di messaggio `FATAL: could not receive data from WAL stream`, assieme a `ERROR: requested WAL segment ... has already been removed`.

Come evidenziato nella riga in grassetto, Amazon RDS recupera la replica riproducendo un file WAL archiviato.

```
2014-11-07 19:01:10 UTC::@:[23180]:DEBUG:  switched WAL source from archive to stream
 after failure
2014-11-07 19:01:10 UTC::@:[11575]:LOG: started streaming WAL from primary at 1A/
D3000000 on timeline 1
2014-11-07 19:01:10 UTC::@:[11575]:FATAL: could not receive data from WAL stream:
ERROR:  requested WAL segment 000000010000001A000000D3 has already been removed
2014-11-07 19:01:10 UTC::@:[23180]:DEBUG: could not restore file "00000002.history"
 from archive: return code 0
2014-11-07 19:01:15 UTC::@:[23180]:DEBUG: switched WAL source from stream to archive
 after failure recovering 000000010000001A000000D3
2014-11-07 19:01:16 UTC::@:[23180]:LOG:  restored log file "000000010000001A000000D3"
 from archive
```

Quando Amazon RDS riproduce abbastanza file WAL archiviati sulla replica da recuperare, viene ripreso lo streaming nella replica di lettura. Quando riprende lo streaming, Amazon RDS scrive una voce nel file di registro, simile alla seguente.

```
2014-11-07 19:41:36 UTC::@:[24714]:LOG:started streaming WAL from primary at 1B/
B6000000 on timeline 1
```

Impostazioni dei parametri di controllo della memoria condivisa

I parametri impostati determinano la dimensione della memoria condivisa per tenere traccia degli ID delle transazioni, dei blocchi e delle transazioni preparate. La struttura della memoria condivisa di un'istanza in standby deve essere uguale o superiore a quella di un'istanza primaria. Ciò garantisce che la prima non esaurisca la memoria condivisa durante il ripristino. Se i valori dei parametri sulla replica sono inferiori ai valori dei parametri sulla replica primaria, Amazon RDS regolerà automaticamente i parametri della replica e riavvierà il motore.

I parametri interessati sono:

- max_connections
- max_worker_processes
- max_wal_senders
- max_prepared_transactions
- max_locks_per_transaction

Per evitare che RDS riavvii le repliche a causa della memoria insufficiente, si consiglia di applicare le modifiche ai parametri come riavvio in sequenza a ciascuna replica. È necessario applicare le seguenti regole quando si impostano i parametri:

- Aumento dei valori dei parametri:
  - È sempre necessario aumentare prima i valori dei parametri di tutte le repliche di lettura ed eseguire un riavvio in sequenza di tutte le repliche. Quindi, applica le modifiche ai parametri sull'istanza primaria ed esegui un riavvio.
- Riduzione dei valori dei parametri:
  - È innanzitutto necessario ridurre i valori dei parametri dell'istanza primaria ed eseguire un riavvio. Quindi, applica le modifiche ai parametri a tutte le repliche di lettura associate ed esegui un riavvio in sequenza.

Monitoraggio e ottimizzazione del processo di replica

Si consiglia vivamente di monitorare regolarmente l'istanza database RDS per PostgreSQL e le repliche di lettura. È necessario assicurarsi che le repliche di lettura siano aggiornate in base alle modifiche dell'istanza database di origine. Amazon RDS recupera in modo trasparente le repliche di lettura quando si verificano interruzioni del processo di replica. Tuttavia, è meglio evitare il ripristino. Il ripristino tramite slot di replica è più rapido rispetto all'utilizzo dell'archivio Amazon S3, ma qualsiasi processo di ripristino può influire sulle prestazioni di lettura.

Per determinare la qualità dell'aggiornamento delle repliche di lettura in base all'istanza database di origine, è possibile effettuare le seguenti operazioni:

- Verifica valore di **`ReplicaLag`** tra istanza database di origine e repliche. Il valore del ritardo di replica si riferisce al tempo di ritardo di una replica di lettura, in secondi, rispetto all'istanza database di origine. Questo parametro indica il risultato della query seguente.

  ```
  SELECT extract(epoch from now() - pg_last_xact_replay_timestamp()) AS "ReplicaLag";
  ```

  Il ritardo di replica è un'indicazione della velocità con cui una replica di lettura rimane al passo con l'istanza database di origine. È il valore della latenza tra l'istanza database di origine e un'istanza di lettura specifica. Un valore elevato del ritardo di replica può indicare una mancata corrispondenza tra le classi di istanza database o i tipi di archiviazione (o entrambi) utilizzati dall'istanza database di origine e le relative repliche di lettura. La classe di istanza database, i tipi di archiviazione per l'istanza database di origine e tutte le repliche di lettura devono essere uguali.

Il ritardo della replica può anche essere il risultato di problemi di connessione non stabile. Puoi monitorare il ritardo di replica in Amazon CloudWatch visualizzando la metrica Amazon `ReplicaLag` RDS. Per ulteriori informazioni su `ReplicaLag` e altri parametri per Amazon RDS, consulta CloudWatch Parametri Amazon per Amazon RDS.

- Controlla il registro PostgreSQL per informazioni che puoi usare per regolare le impostazioni. In ogni checkpoint, il registro PostgreSQL acquisisce il numero di file di registro delle transazioni riciclati, come illustrato nell'esempio seguente.

```
2014-11-07 19:59:35 UTC::@:[26820]:LOG:  checkpoint complete: wrote 376 buffers
 (0.2%);
0 transaction log file(s) added, 0 removed, 1 recycled; write=35.681 s, sync=0.013 s,
 total=35.703 s;
sync files=10, longest=0.013 s, average=0.001 s
```

  Puoi utilizzare queste informazioni per capire quanti file di transazione verranno riciclati in un determinato periodo di tempo. Puoi modificare l'impostazione del parametro `wal_keep_segments`, se necessario. Supponiamo, ad esempio, che il registro di PostgreSQL in `checkpoint complete` mostri `35 recycled` per un intervallo di 5 minuti. In questo caso, il valore predefinito 32 del parametro `wal_keep_segments` non è sufficiente per tenere il passo con l'attività di streaming e pertanto è consigliabile aumentare il valore di questo parametro.

- Usa Amazon CloudWatch per monitorare i parametri in grado di prevedere i problemi di replica. Invece di analizzare direttamente il log di PostgreSQL, puoi utilizzare CloudWatch Amazon per controllare i parametri raccolti. Ad esempio, è possibile monitorare il valore del parametro `TransactionLogsGeneration` per vedere quanti dati WAL vengono generati dall'istanza database di origine. In alcuni casi, il carico di lavoro sull'istanza database potrebbe generare una grande quantità di dati WAL. In questo caso, potrebbe essere necessario modificare la classe di istanza database per l'istanza database di origine e delle repliche di lettura. L'utilizzo di una classe di istanza con prestazioni di rete elevate (10 Gb/s) può ridurre il ritardo delle repliche.

Monitoraggio degli slot di replica per l'istanza database RDS per PostgreSQL

Tutte le versioni di RDS per PostgreSQL utilizzano gli slot di replica per le repliche di lettura tra regioni. RDS per PostgreSQL 14.1 e versioni successive utilizzano gli slot di replica per le repliche di lettura a livello di regione. Le repliche di lettura a livello di regione utilizzano anche Amazon S3 per archiviare i dati WAL. In altre parole, se l'istanza database e le repliche di lettura eseguono PostgreSQL 14.1 o versioni successive, gli slot di replica e gli archivi Amazon S3 sono entrambi

disponibili per il ripristino della replica in lettura. Il ripristino di una replica di lettura utilizzando lo slot di replica è più veloce del ripristino dall'archivio Amazon S3. Pertanto, ti consigliamo di monitorare gli slot di replica e i relativi parametri.

È possibile visualizzare gli slot di replica sulle istanze database RDS per PostgreSQL eseguendo una query sulla vista `pg_replication_slots`, come segue.

```
postgres=> SELECT * FROM pg_replication_slots;
slot_name                  | plugin | slot_type | datoid | database | temporary |
 active | active_pid | xmin | catalog_xmin | restart_lsn | confirmed_flush_lsn |
 wal_status | safe_wal_size | two_phase
---------------------------+--------+-----------+--------+----------+-----------
+--------+------------+------+--------------+-------------+---------------------
+------------+---------------+-----------
rds_us_west_1_db_555555555 |        | physical  |        |          | f         | t
  |      13194 |      |              | 23/D8000060 |                     | reserved   |
               | f
(1 row)
```

Il parametro `wal_status` con valore `reserved` significa che la quantità di dati WAL conservati dallo slot rientra nei limiti del parametro `max_wal_size`. In altre parole, lo slot di replica è dimensionato correttamente. I valori di stato possibili sono i seguenti:

- `extended`: lo slot supera l'impostazione del parametro `max_wal_size`, ma i dati WAL vengono conservati.
- `unreserved`: lo slot non include più tutti i dati WAL richiesti. Alcuni di essi verranno rimossi al prossimo checkpoint.
- `lost`: alcuni dati WAL obbligatori sono stati rimossi. Lo slot non è più utilizzabile.

`lost`Gli stati `unreserved` e di `wal_status` vengono visualizzati solo quando non è negativo. `max_slot_wal_keep_size`

La vista `pg_replication_slots` mostra lo stato corrente degli slot di replica. Per valutare le prestazioni dei tuoi slot di replica, puoi utilizzare Amazon CloudWatch e monitorare i seguenti parametri:

- **`OldestReplicationSlotLag`**: elenca la replica con il ritardo più elevato, cioè è più distante rispetto al database primario. Questo ritardo può essere dovuto alla replica di lettura ma anche alla connessione.

- **`TransactionLogsDiskUsage`** - Mostra la quantità di archiviazione utilizzata per i dati WAL. Quando una replica di lettura è in ritardo significativo, il valore di questo parametro può aumentare notevolmente.

Per ulteriori informazioni sull'utilizzo di Amazon CloudWatch e dei relativi parametri per RDS per PostgreSQL, consulta. Monitoraggio dei parametri di Amazon RDS con Amazon CloudWatch Per ulteriori informazioni sul monitoraggio della replica in streaming sulle istanze database RDS per PostgreSQL, consulta la sezione relativa alle best practice per la replica Amazon RDS PostgreSQL sul blog dei database AWS .

Risoluzione dei problemi relativi alla replica di lettura RDS per PostgreSQL

Di seguito, puoi trovare idee per la risoluzione di alcuni problemi comuni relativi alla replica di lettura di RDS per PostgreSQL.

Termina la query che causa il ritardo della replica di lettura

Le transazioni attive o inattive in stato di transazione che sono in esecuzione da molto tempo nel database potrebbero interferire con il processo di replica WAL, aumentando così il ritardo di replica. Pertanto, assicurati di monitorare il runtime di queste transazioni con la vista PostgreSQL`pg_stat_activity`.

Esegui una query sull'istanza principale simile alla seguente per trovare l'ID di processo (PID) della query in esecuzione da molto tempo:

```
SELECT datname, pid,usename, client_addr, backend_start,
xact_start, current_timestamp - xact_start AS xact_runtime, state,
backend_xmin FROM pg_stat_activity WHERE state='active';
```

```
SELECT now() - state_change as idle_in_transaction_duration, now() - xact_start as
 xact_duration,*
FROM  pg_stat_activity
WHERE state  = 'idle in transaction'
AND   xact_start is not null
ORDER BY 1 DESC;
```

Dopo aver identificato il PID della query, puoi scegliere di terminare la query.

Esegui una query sull'istanza principale simile alla seguente per terminare la query in esecuzione da molto tempo:

```
SELECT pg_terminate_backend(PID);
```

Prestazioni delle query migliorate per RDS per PostgreSQL con Letture ottimizzate per Amazon RDS

Puoi velocizzare l'elaborazione delle query per RDS per PostgreSQL con Letture ottimizzate per Amazon RDS. Un'istanza database o un cluster database multi-AZ RDS per PostgreSQL che utilizza Letture ottimizzate per Amazon RDS può ottenere un'elaborazione delle query fino al 50% più veloce rispetto a un'istanza database o a un cluster database che non lo utilizza.

Argomenti

Panoramica di Letture ottimizzate per Amazon RDS in PostgreSQL

Letture ottimizzate è disponibile per impostazione predefinita su RDS per PostgreSQL versioni 15.2 e successive, 14.7 e successive e 13.10 e successive.

Quando si utilizza un'istanza database o un cluster database multi-AZ RDS per PostgreSQL con la funzionalità Letture ottimizzate per Amazon RDS attivata, si ottengono ottiene prestazioni di query fino al 50% più rapide tramite l'archiviazione locale a livello di blocchi SSD basata su NVMe (Non-Volatile Memory Express). È possibile velocizzare l'elaborazione delle query posizionando le tabelle temporanee generate da PostgreSQL nello spazio di archiviazione locale, il che riduce il traffico verso Elastic Block Storage (EBS) in rete.

In PostgreSQL, gli oggetti temporanei vengono assegnati a uno spazio dei nomi temporaneo che viene eliminato automaticamente alla fine della sessione. Lo spazio dei nomi temporaneo durante l'eliminazione rimuove tutti gli oggetti dipendenti dalla sessione, inclusi gli oggetti qualificati dallo schema, come tabelle, funzioni, operatori o persino estensioni.

In RDS per PostgreSQL, il parametro `temp_tablespaces` è configurato per questa area di lavoro temporanea in cui sono archiviati gli oggetti temporanei.

Le seguenti query restituiscono il nome dello spazio dei nomi e la sua posizione.

```
postgres=> show temp_tablespaces;
temp_tablespaces
---------------------
rds_temp_tablespace
(1 row)
```

`rds_temp_tablespace` è una tablespace configurata da RDS che punta allo spazio di archiviazione locale NVMe. È sempre possibile tornare allo spazio di archiviazione di Amazon EBS modificando questo parametro nel `Parameter group` utilizzando la AWS Management Console per puntare a qualsiasi tablespace diverso da `rds_temp_tablespace`. Per ulteriori informazioni, consulta sull'argomento relativo alla modifica dei parametri in un gruppo di parametri del database. È inoltre possibile utilizzare il comando SET per modificare il valore del parametro `temp_tablespaces` impostandolo su `pg_default` a livello di sessione utilizzando il comando SET. La modifica del parametro reindirizza l'area di lavoro temporanea su Amazon EBS. Il passaggio ad Amazon EBS è utile quando lo spazio di archiviazione locale per l'istanza o un cluster RDS non è sufficiente per eseguire una specifica operazione SQL.

```
postgres=> SET temp_tablespaces TO 'pg_default';
SET
```

```
postgres=> show temp_tablespaces;

 temp_tablespaces
------------------
 pg_default
```

Casi d'uso per RDS Optimized Reads

Di seguito sono riportati alcuni casi d'uso che possono trarre vantaggio dalla funzionalità Letture ottimizzate per Amazon RDS:

- Query analitiche con espressioni di tabella comuni (CTE), tabelle derivate e operazioni di raggruppamento.
- Repliche di lettura che gestiscono le query non ottimizzate per un'applicazione.
- Query di reporting on demand o dinamiche con operazioni complesse come GROUP BY e ORDER BY che non sempre possono utilizzare indici appropriati.

- Altri carichi di lavoro che utilizzano tabelle temporanee interne.
- `CREATE INDEX`o `REINDEX` operazioni di ordinamento.

Best practice per RDS Optimized Reads

Usa le seguenti best practice per RDS Optimized Reads:

- Aggiungi la logica dei tentativi per le query di sola lettura, nel caso in cui non riescano perché l'archivio dell'istanza è completo durante l'esecuzione.
- Monitora lo spazio di archiviazione disponibile sull'instance store con la CloudWatch metrica`FreeLocalStorage`. Se l'archivio dell'istanza sta raggiungendo il limite a causa del carico di lavoro dell'istanza database o sul cluster database multi-AZ, modificare l'istanza database in modo da utilizzare una classe di istanza database più grande.

Utilizzo di RDS Optimized Reads

L'istanza database utilizza automaticamente la funzionalità Letture ottimizzate per Amazon RDS quando si effettua il provisioning di un'istanza database RDS per PostgreSQL con una delle seguenti classi di istanza database basate su NVMe in un'implementazione di istanza database Single-AZ, un'implementazione di istanza database Multi-AZ o un'implementazione di cluster database Multi-AZ.

Per ulteriori informazioni sull'implementazione Multi-AZ, consulta Configurazione e gestione di un'implementazione Multi-AZ.

Per attivare RDS Optimized Reads, procedi in uno dei seguenti modi:

- Crea un'istanza database o un cluster database Multi-AZ RDS per PostgreSQL utilizzando una di queste classi di istanza database NVMe. Per ulteriori informazioni, consulta Creazione di un'istanza database Amazon RDS.
- Modifica un'istanza database o un cluster database Multi-AZ RDS per PostgreSQL esistente per utilizzare una di queste classi di istanza database NVMe. Per ulteriori informazioni, consulta Modifica di un'istanza database Amazon RDS.

La funzionalità Letture ottimizzate per Amazon RDS è disponibile in tutte le Regioni AWS in cui sono supportate una o più delle classi di istanza database con spazio di archiviazione SSD NVMe locale. Per ulteriori informazioni, consulta Classi di istanze database.

Per tornare a un'istanza senza la funzionalità Letture ottimizzate per Amazon RDS abilitata, modificare la classe dell'istanza database dell'istanza o del cluster RDS con una classe di istanza simile che supporta solo lo spazio di archiviazione EBS per i carichi di lavoro del database. Ad esempio, se la classe di istanza database corrente è db.r6gd.4xlarge, scegli db.r6g.4xlarge per tornare indietro. Per ulteriori informazioni, consulta Modifica di un'istanza DB di Amazon RDS.

Monitoraggio delle istanze database che utilizzano RDS Optimized Reads

È possibile monitorare le istanze DB che utilizzano RDS Optimized Reads utilizzando le seguenti metriche: CloudWatch

- `FreeLocalStorage`
- `ReadIOPSLocalStorage`
- `ReadLatencyLocalStorage`
- `ReadThroughputLocalStorage`
- `WriteIOPSLocalStorage`
- `WriteLatencyLocalStorage`
- `WriteThroughputLocalStorage`

Queste metriche forniscono dati sullo spazio di archiviazione dell'archivio dell'istanza, sulle operazioni IOPS e sulla velocità di trasmissione effettiva disponibili. Per ulteriori informazioni su questi parametri, consulta Parametri a CloudWatch livello di istanza Amazon per Amazon RDS.

Per monitorare l'utilizzo corrente dello spazio di archiviazione locale, accedi al tuo database utilizzando la seguente query:

```
SELECT
    spcname AS "Name",
    pg_catalog.pg_size_pretty(pg_catalog.pg_tablespace_size(oid)) AS "size"
FROM
    pg_catalog.pg_tablespace
WHERE
    spcname IN ('rds_temp_tablespace');
```

Per ulteriori informazioni sui file temporanei e sul loro utilizzo, consulta Gestione dei file temporanei con PostgreSQL.

Limitazioni per Letture ottimizzate per Amazon RDS in PostgreSQL

Alla funzionalità Letture ottimizzate per Amazon RDS si applica la seguente limitazione:

- Le transazioni possono non riuscire quando l'archivio dell'istanza è pieno.

Importazione di dati in PostgreSQL su Amazon RDS

Si supponga di avere una distribuzione PostgreSQL esistente da spostare in Amazon RDS. La complessità dell'attività dipende dalle dimensioni del database e dal tipo di oggetti di database da trasferire. Pensa, ad esempio, a un database con diversi gigabyte di set di dati, trigger e stored procedure. Trasferire un database di questo tipo è più complicato che trasferire un semplice database con pochi megabyte di dati di test e senza trigger o stored procedure.

Ti consigliamo di utilizzare gli strumenti di migrazione dei database PostgreSQL nativi nei seguenti casi:

- Hai una migrazione omogenea, dove effettui la migrazione da un database con lo stesso motore del database del database di destinazione.
- Desideri migrare un intero database.
- Gli strumenti nativi ti consentono di migrare il tuo sistema con tempi di inattività ridotti.

Nella maggior parte degli altri casi, eseguire una migrazione del database utilizzando AWS Database Migration Service (AWSDMS) è l'approccio migliore. AWS DMS è in grado di migrare i database senza tempi di inattività e, per molti motori di database, procede con la replica continua fino a quando non è tutto pronto per passare al database di destinazione. Puoi effettuare la migrazione allo stesso motore del database o a un motore del database diverso tramite AWS DMS. Se si esegue la migrazione a un motore del database diverso rispetto al database di origine, si può utilizzare AWS Schema Conversion Tool (AWS SCT). AWS SCT può essere utilizzato per eseguire la migrazione degli oggetti dello schema di cui non viene eseguita la migrazione con AWS DMS. Per ulteriori informazioni su AWS DMS, consulta Che cos'è AWS Database Migration Service?

Modifica il gruppo di parametri database per includere le seguenti impostazioni solo per la tua importazione. Per individuare le impostazioni più efficienti per le dimensioni della tua istanza database devi testare le impostazioni dei parametri. Dovrai inoltre ripristinare i valori di produzione per questi parametri al termine dell'impostazione.

Modifica i parametri dell'istanza database come segue:

- Disabilita i backup delle istanze database (imposta backup_retention su 0).
- Disabilita Multi-AZ.

Modifica il gruppo di parametri del database in modo da includere le seguenti impostazioni. Utilizza queste impostazioni soltanto quando importi i dati. Per individuare le impostazioni più efficienti per le dimensioni della tua istanza database devi testare le impostazioni dei parametri. Dovrai inoltre ripristinare i valori di produzione per questi parametri al termine dell'impostazione.

Parametro	Valori consigliati durante l'importazione	Descrizione
`maintenance_work_mem`	524288, 1048576, 2097152 o 4194304 (in KB). Queste impostazioni sono paragonabili a 512 MB, 1 GB, 2 GB e 4 GB.	Il valore di questa impostazione dipende dalle dimensioni dell'host. Questo parametro viene utilizzato durante le istruzioni CREATE INDEX e ogni comando parallelo può utilizzare questa quantità di memoria. Calcola il valore ottimale, per evitare di impostare un valore troppo alto ed esaurire la memoria.
`max_wal_size`	256 (per la versione 9.6), 4096 (per le versioni 10 e successive)	Dimensione massima per far crescere il WAL durante i checkpoint automatici. Aumentando questo parametro puoi aumentare il tempo necessario per il ripristino di caso di arresto anomalo. Questo parametro sostituisce `checkpoint_segments` per PostgreSQL 9.6 e versioni successive. Per PostgreSQL versione 9.6, questo valore è espresso in unità da 16 MB. Per le versioni successive, il valore è espresso in unità da 1 MB. Ad esempio, nella versione 9.6, 128 significa 128 blocchi ognuno con una dimensione di 16 MB. Nella versione 12.4, 2048 significa 2048 blocchi che hanno una dimensione di 1 MB.
`checkpoint_timeout`	1800	Il valore per questa impostazione consente una rotazione WAL meno frequente.
`synchronous_commit`	Disattivata	Disabilita questa impostazione per velocizzare la scrittura. La disattivazione del parametro può

Parametro	Valori consigliati durante l'importazione	Descrizione
		aumentare il rischio di perdita di dati in caso di arresto anomalo del server (non disattivare FSYNC)
`wal_buffers`	8192	Questo valore è impostato in unità di 8 KB. Anche questo aiuta a velocizzare la generazio ne dei WAL.
`autovacuum`	0	Disabilita il parametro di eliminazione automatic a PostgreSQL durante il caricamento dei dati in modo che non utilizzi risorse.

Utilizza i comandi `pg_dump -Fc` (compresso) o `pg_restore -j` (parallelo) con queste impostazioni.

> **Note**
>
> Il comando PostgreSQL `pg_dumpall` richiede autorizzazioni super_user che non vengono concesse quando crei un'istanza database, quindi non puoi utilizzarlo per importare i dati.

Argomenti

- Importazione di un database PostgreSQL da un'istanza Amazon EC2
- Utilizzo del comando \copy per importare i dati in una tabella su un'istanza database PostgreSQL
- Importazione di dati da Amazon S3 in un'istanza database RDS per PostgreSQL
- Trasporto dei database PostgreSQL tra istanze database

Importazione di un database PostgreSQL da un'istanza Amazon EC2

Se i tuoi dati si trovano in un server PostgreSQL su un'istanza Amazon EC2 e desideri trasferirli in un'istanza database PostgreSQL, puoi utilizzare la seguente procedura. Nell'elenco seguente è indicata la procedura da eseguire. Ciascun passaggio della procedura è descritto in modo dettagliato nelle sezioni seguenti.

1. Creare un file utilizzando pg_dump che contiene i dati da caricare
2. Creare l'istanza database di destinazione
3. Utilizzare psql per creare il database sull'istanza database e caricare i dati
4. Creare uno snapshot DB dell'istanza database

Passo 1: creare un file utilizzando pg_dump che contiene i dati da caricare

L'utilità pg_dump utilizza il comando COPY per creare uno schema e il dump dei dati di un database PostgreSQL. Lo script del dump generato da pg_dump carica i dati in un database con lo stesso nome e ricrea le tabelle, gli indici e le chiavi esterne. Puoi utilizzare il comando pg_restore e il parametro -d per ripristinare i dati in un database con un nome diverso.

Prima di creare il dump dei dati, devi eseguire le query delle tabelle da sottoporre a dump per ottenere un conteggio delle righe e confermare il numero sull'istanza database di destinazione.

Il seguente commando crea un file dump denominato mydb2dump.sql per un database denominato mydb2.

```
prompt>pg_dump dbname=mydb2 -f mydb2dump.sql
```

Passo 2: creare l'istanza database di destinazione.

Crea l'istanza database PostgreSQL di destinazione usando la console Amazon RDS, AWS CLI o l'API. Crea l'istanza con l'impostazione di retention dei backup impostata su 0 e disabilita l'opzione Multi-AZ. Ciò consente di velocizzare l'importazione dei dati. Devi creare un database sull'istanza prima di poter effettuare il dump dei dati. Il database può avere lo stesso nome del database che conteneva i dati del dump. In alternativa, puoi creare un database con un nome diverso. In questo caso, puoi utilizzare il comando pg_restore e il parametro -d per ripristinare i dati nel database con il nuovo nome.

Ad esempio, puoi utilizzare i seguenti comandi per il dump, il ripristino e la ridenominazione di un database.

```
pg_dump -Fc -v -h [endpoint of instance] -U [master username] [database]
 > [database].dump
createdb [new database name]
pg_restore -v -h [endpoint of instance] -U [master username] -d [new database
 name] [database].dump
```

Passo 3: utilizzare psql per creare il database sull'istanza database e caricare i dati

Puoi utilizzare la stessa connessione utilizzata per eseguire il comando pg_dump per connetterti all'istanza database di destinazione e ricreare il database. Servendoti di psql, puoi utilizzare il nome utente master e la password master per creare il database sull'istanza database.

Il seguente esempio utilizza psql e un file dump denominato mydb2dump.sql per creare un database denominato mydb2 su un'istanza database PostgreSQL denominata mypginstance:

Per LinuxmacOS, oUnix:

```
psql \
   -f mydb2dump.sql \
   --host mypginstance.555555555555.aws-region.rds.amazonaws.com \
   --port 8199 \
   --username myawsuser \
   --password password \
   --dbname mydb2
```

Per Windows:

```
psql ^
   -f mydb2dump.sql ^
   --host mypginstance.555555555555.aws-region.rds.amazonaws.com ^
   --port 8199 ^
   --username myawsuser ^
   --password password ^
   --dbname mydb2
```

> **Note**
>
> Specifica una password diversa dal prompt mostrato qui come best practice per la sicurezza.

Passo 4: creare uno snapshot DB dell'istanza database.

Dopo aver verificato che i dati siano stati caricati nella tua istanza database, ti consigliamo di creare uno snapshot DB dell'istanza database PostgreSQL di destinazione. Le snapshot DB sono backup completi della tua istanza database che puoi utilizzare per ripristinare la tua istanza database in uno stato noto. Una snapshot DB ottenuta immediatamente dopo il caricamento consente di non dover

caricare nuovamente i dati in caso di problemi. Può essere utilizzata per inizializzare nuove istanze database. Per ulteriori informazioni sulla creazione di uno snapshot DB, consulta Creazione di uno snapshot DB per un'istanza DB Single-AZ.

Utilizzo del comando \copy per importare i dati in una tabella su un'istanza database PostgreSQL

Il comando `\copy` di PostgreSQL è un meta-comando disponibile dallo strumento client interattivo `psql`. Puoi utilizzare `\copy` per importare i dati in una tabella sull'istanza database RDS for PostgreSQL. Per utilizzare il comando `\copy`, è necessario innanzitutto creare la struttura della tabella sull'istanza database di destinazione in modo che `\copy` abbia una destinazione per i dati copiati.

Puoi utilizzare `\copy` per caricare i dati da un file di valori separati da virgole (CSV), ad esempio uno che è stato esportato e salvato nella workstation client.

Per importare i dati CSV nell'istanza database RDS for PostgreSQL di destinazione, connettiti prima all'istanza database di destinazione utilizzando `psql`.

```
psql --host=db-instance.111122223333.aws-region.rds.amazonaws.com --port=5432 --
username=postgres --password --dbname=target-db
```

Quindi esegui il comando `\copy` con i seguenti parametri per identificare la destinazione per i dati e il relativo formato.

- `target_table` - Il nome della tabella che dovrebbe ricevere i dati copiati dal file CSV.
- `column_list` - Specifiche delle colonne per la tabella.
- `'filename'` - Il percorso completo del file CSV sulla workstation locale.

```
 \copy target_table from '/path/to/local/filename.csv' WITH DELIMITER ',' CSV;
```

Se il file CSV contiene informazioni sull'intestazione di colonna, puoi utilizzare questa versione del comando e dei parametri.

```
\copy target_table (column-1, column-2, column-3, ...)
    from '/path/to/local/filename.csv' WITH DELIMITER ',' CSV HEADER;
```

Se il comando `\copy` ha esito negativo, PostgreSQL emette messaggi di errore.

Creazione di una nuova istanza DB nell'ambiente Database Preview utilizzando il `psql` comando con il `\copy` meta-comando, come mostrato negli esempi seguenti. Questo esempio utilizza source-table come nome della tabella di origine, source-table.csv come file .csv e target-db come database di destinazione:

PerLinux, omacOS: Unix

```
$psql target-db \
    -U <admin user> \
    -p <port> \
    -h <DB instance name> \
    -c "\copy source-table from 'source-table.csv' with DELIMITER ','"
```

Per Windows:

```
$psql target-db ^
    -U <admin user> ^
    -p <port> ^
    -h <DB instance name> ^
    -c "\copy source-table from 'source-table.csv' with DELIMITER ','"
```

Per informazioni dettagliate sul comando `\copy`, consulta la pagina psql nella documentazione di PostgreSQL, nella sezione Meta-comandi.

Importazione di dati da Amazon S3 in un'istanza database RDS per PostgreSQL

Puoi importare i dati che sono stati archiviati utilizzando Servizio di archiviazione semplice Amazon in una tabella su un'istanza database RDS per PostgreSQL. A questo scopo, installa innanzitutto l'estensione RDS per PostgreSQL `aws_s3`. Questa estensione fornisce le funzioni utilizzate per importare i dati da un bucket Amazon S3. Un bucket è un container Amazon S3 per oggetti e file. I dati possono trovarsi in un file con valori separati da virgole (CSV), un file di testo o un file compresso (gzip). Di seguito, sono fornite informazioni su come installare l'estensione e come importare dati da Amazon S3 in una tabella.

Per eseguire l'importazione da Amazon S3 a RDS for PostgreSQL, il database deve eseguire PostgreSQL versione 10.7 o successive.

Se Amazon S3 non contiene dati, occorre innanzitutto creare un bucket e archiviare i dati. Per ulteriori informazioni, consulta i seguenti argomenti nella Guida per l'utente di Servizio di archiviazione semplice Amazon.

- Creazione di un bucket
- Aggiunta di un oggetto a un bucket.

È supportata l'importazione multiaccount da Amazon S3. Per ulteriori informazioni, consulta Concessione di autorizzazioni multiaccount nella Guida per l'utente di Amazon Simple Storage Service.

Puoi utilizzare la chiave gestita dal cliente per la crittografia durante l'importazione dei dati da S3. Per ulteriori informazioni, consulta Chiavi KMS archiviate in AWS KMS nella Guida per l'utente di Amazon Simple Storage Service.

> **Note**
>
> L'importazione di dati da Amazon S3 non è supportata per Aurora Serverless v1. È supportata per Aurora Serverless v2.

Argomenti

- Installazione dell'estensione aws_s3
- Panoramica dell'importazione di dati dai dati di Amazon S3
- Configurazione dell'accesso a un bucket Amazon S3
- Importazione di dati da Amazon S3 nell'istanza database RDS per PostgreSQL
- Informazioni di riferimento sulle funzioni

Installazione dell'estensione aws_s3

Prima di poter utilizzare Amazon S3 con l'istanza database RDS per PostgreSQL, è necessario installare l'estensione `aws_s3`. Questa estensione fornisce funzioni per l'importazione dei dati da Amazon S3. Inoltre, fornisce funzioni per l'esportazione di dati da un'istanza database RDS per PostgreSQL in un bucket Amazon S3. Per ulteriori informazioni, consulta Esportazione di dati da un'istanza di database del RDS per PostgreSQL a Amazon S3. L'estensione `aws_s3` dipende da

alcune delle funzioni helper nell'estensione `aws_commons`, che vengono installate automaticamente quando necessario.

Per installare l'estensione **`aws_s3`**

1. Usa psql (o pgAdmin) per connetterti all'istanza database RDS per PostgreSQL come un utente che dispone di privilegi `rds_superuser`. Se hai mantenuto il nome predefinito durante il processo di configurazione, esegui la connessione come `postgres`.

```
psql --host=111122223333.aws-region.rds.amazonaws.com --port=5432 --
username=postgres --password
```

2. Per installare l'estensione, esegui il comando seguente.

```
postgres=> CREATE EXTENSION aws_s3 CASCADE;
NOTICE: installing required extension "aws_commons"
CREATE EXTENSION
```

3. Per verificare che l'estensione sia installata, puoi usare il metacomando psql `\dx`.

```
postgres=> \dx
       List of installed extensions
    Name     | Version |   Schema   |                Description
-------------+---------+------------+-------------------------------------------
 aws_commons | 1.2     | public     | Common data types across AWS services
 aws_s3      | 1.1     | public     | AWS S3 extension for importing data from S3
 plpgsql     | 1.0     | pg_catalog | PL/pgSQL procedural language
(3 rows)
```

Le funzioni per importare dati da Amazon S3 ed esportare dati in Amazon S3 sono ora disponibili per l'uso.

Panoramica dell'importazione di dati dai dati di Amazon S3

Per importare i dati S3 in Amazon RDS

Raccogli innanzitutto i dettagli che devi fornire alla funzione. Questi includono il nome della tabella su la tua istanza PostgreSQL RDSe il nome del bucket, il percorso del file, il tipo di file eRegione AWSdove sono memorizzati i dati Amazon S3. Per ulteriori informazioni, consulta Visualizzazione di un oggetto nella Guida per l'utente di Servizio di archiviazione semplice Amazon.

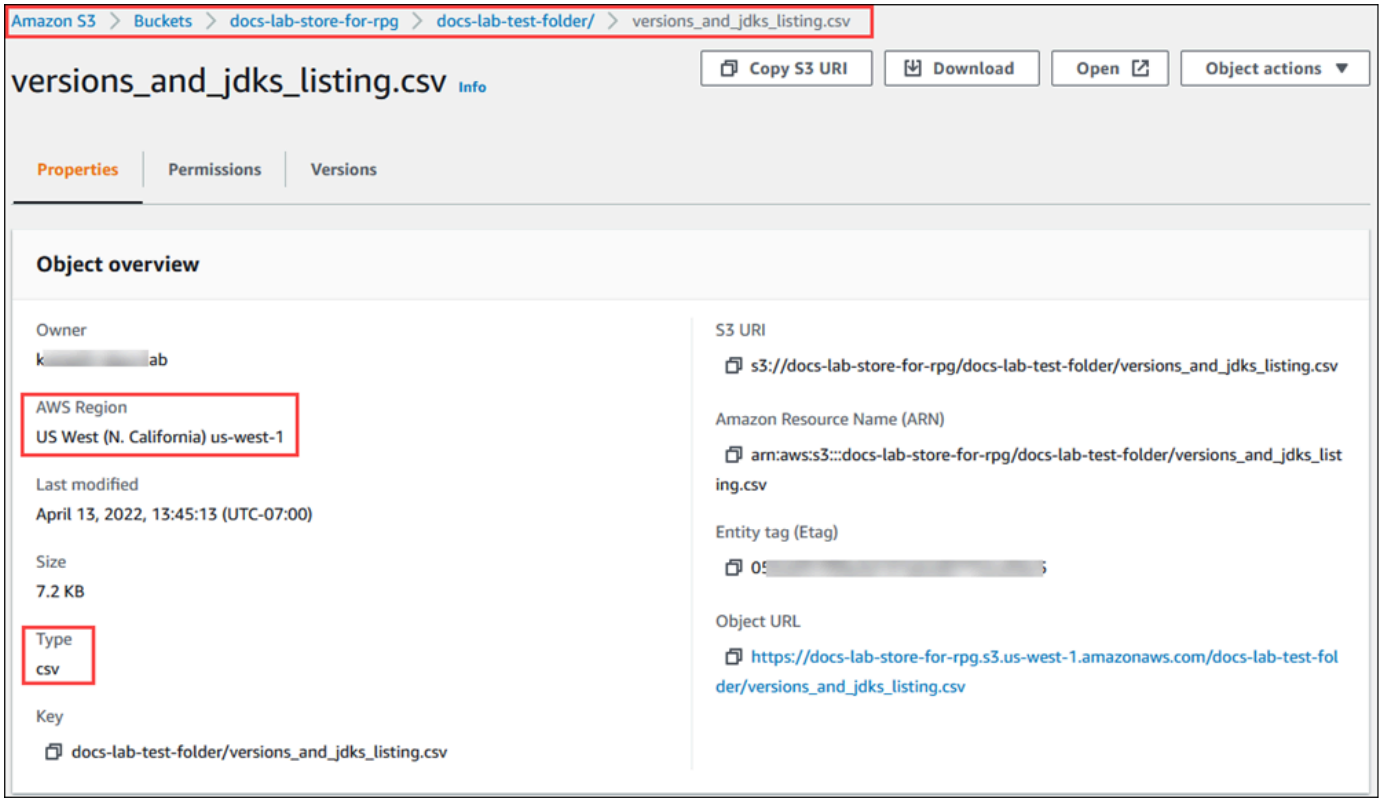
> **Note**
>
> L'importazione in più parti da Amazon S3 non è attualmente supportata.

1. Otteni il nome della tabella in cui la funzione `aws_s3.table_import_from_s3` deve importare dati. Il seguente comando, ad esempio, crea una tabella `t1` che può essere utilizzata in fasi successive.

   ```
   postgres=> CREATE TABLE t1
       (col1 varchar(80),
       col2 varchar(80),
       col3 varchar(80));
   ```

2. Ottieni i dettagli relativi al bucket Amazon S3 e i dati da importare. A tale scopo, apri la console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/ e scegli Buckets (Bucket). Individua il bucket contenente i dati nell'elenco. Scegli il bucket, apri la pagina Object overview (Panoramica degli oggetti) e quindi scegli Properties (Proprietà).

   Prendi nota del nome del bucket, del percorso, della Regione AWS e del tipo di file. Il nome della risorsa Amazon (ARN) è richiesto in un secondo momento per configurare l'accesso ad Amazon S3 tramite un ruolo IAM. Per ulteriori informazioni, consulta Configurazione dell'accesso a un bucket Amazon S3. Un esempio è illustrato nell'immagine seguente.



3. Puoi verificare il percorso ai dati sul bucket Amazon S3 utilizzando il comando AWS CLI `aws s3 cp`. Se le informazioni sono corrette, questo comando scarica una copia del file Amazon S3.

```
aws s3 cp s3://sample_s3_bucket/sample_file_path ./
```

4. Configura le autorizzazioni sull'istanza database RDS per PostgreSQL per consentire l'accesso al file sul bucket Amazon S3. A questo scopo, utilizza un ruolo AWS Identity and Access Management (IAM) o le credenziali di sicurezza. Per ulteriori informazioni, consulta Configurazione dell'accesso a un bucket Amazon S3.

5. Fornisci alla funzione `create_s3_uri` il percorso e gli altri dettagli dell'oggetto Amazon S3 raccolti (vedi passaggio 2) per costruire un oggetto URI Amazon S3. Per ulteriori informazioni su questa funzione, consulta aws_commons.create_s3_uri. Di seguito è riportato un esempio di costruzione dell'oggetto durante una sessione psql.

```
postgres=> SELECT aws_commons.create_s3_uri(
   'docs-lab-store-for-rpg',
   'versions_and_jdks_listing.csv',
   'us-west-1'
) AS s3_uri \gset
```

Nella fase seguente, si passa questo oggetto (`aws_commons._s3_uri_1`) alla funzione `aws_s3.table_import_from_s3` per importare i dati nella tabella.

6. Invoca la funzione `aws_s3.table_import_from_s3` per importare dati da Amazon S3 nella tabella. Per informazioni di riferimento, consulta aws_s3.table_import_from_s3. Per alcuni esempi, consulta Importazione di dati da Amazon S3 nell'istanza database RDS per PostgreSQL.

Configurazione dell'accesso a un bucket Amazon S3

Per importare i dati da un file Amazon S3, concedere al a RDS per un'istanza database PostgreSQL l'autorizzazione ad accedere al bucket Amazon S3 che contiene il file. Puoi concedere l'accesso a un bucket Amazon S3 in uno dei due modi descritti negli argomenti seguenti.

Argomenti

- Utilizzo di un ruolo IAM per accedere a un bucket Amazon S3.
- Utilizzo delle credenziali di sicurezza per accedere a un bucket Amazon S3
- Risoluzione dei problemi di accesso a Amazon S3

Utilizzo di un ruolo IAM per accedere a un bucket Amazon S3.

Prima di caricare i dati da un file Amazon S3, è necessario concedere a RDS per un'istanza database PostgreSQL l'autorizzazione per accedere al bucket Amazon S3 che contiene il file. In questo modo non dovrai gestire ulteriori informazioni sulle credenziali né fornirle nella chiamata della funzione aws_s3.table_import_from_s3.

Per svolgere questa operazione, creare una policy IAM che fornisca accesso al bucket Amazon S3. Creare un ruolo IAM e collegarvi la policy. Quindi, assegnare il ruolo IAM all'istanza database.

> **Note**
>
> Non è possibile associare un ruolo IAM a un cluster di database Aurora Serverless v1, quindi i seguenti passaggi non sono attinenti.

Per consentire a un'istanza database RDS for PostgreSQL l'accesso ad Amazon S3 tramite un ruolo IAM

1. Creare una policy IAM

   Questa policy fornisce le autorizzazioni bucket e di oggetto che consentono a RDS per un'istanza database PostgreSQL di accedere a Amazon S3.

   Includere nella policy le seguenti operazioni necessarie per consentire il trasferimento dei file da un bucket Amazon S3 a Amazon RDS:

   - `s3:GetObject`
   - `s3:ListBucket`

   Includere nella policy le seguenti risorse per identificare il bucket Amazon S3 e gli oggetti nel bucket. Questo mostra il formato Amazon Resource Name (ARN) per accedere a Amazon S3.

   - arn:aws:s3:::*`your-s3-bucket`*
   - arn:aws:s3:::*`your-s3-bucket`*/*

   Per ulteriori informazioni sulla creazione di una policy IAM per RDS per PostgreSQL, consulta Creazione e utilizzo di una policy IAM per l'accesso al database IAM. Consulta anche il Tutorial: Creare e collegare la prima policy gestita dal cliente nella Guida per l'utente di IAM.

   Il seguente comando dell'AWS CLI crea una policy IAM denominata `rds-s3-import-policy` con queste opzioni. Concede l'accesso a un bucket denominato `your-s3-bucket`.

   > **Note**
   >
   > Prendi nota del nome della risorsa Amazon (ARN) della policy restituita mediante questo comando. L'ARN sarà richiesto in una fase successiva quando si associa la policy a un ruolo IAM.

   Example

   Per LinuxmacOS, oUnix:

```
aws iam create-policy \
   --policy-name rds-s3-import-policy \
   --policy-document '{
     "Version": "2012-10-17",
     "Statement": [
       {
         "Sid": "s3import",
         "Action": [
           "s3:GetObject",
           "s3:ListBucket"
         ],
         "Effect": "Allow",
         "Resource": [
           "arn:aws:s3:::your-s3-bucket",
           "arn:aws:s3:::your-s3-bucket/*"
         ]
       }
     ]
   }'
```

Per Windows:

```
aws iam create-policy ^
   --policy-name rds-s3-import-policy ^
   --policy-document '{
     "Version": "2012-10-17",
     "Statement": [
       {
         "Sid": "s3import",
         "Action": [
           "s3:GetObject",
           "s3:ListBucket"
         ],
         "Effect": "Allow",
         "Resource": [
           "arn:aws:s3:::your-s3-bucket",
           "arn:aws:s3:::your-s3-bucket/*"
         ]
       }
     ]
   }'
```

2. Crea un ruolo IAM.

   In questo modo, Amazon RDS può assumere questo ruolo IAM per tuo conto, per accedere ai bucket Amazon S3. Per ulteriori informazioni, consulta la pagina relativa alla creazione di un ruolo per delegare le autorizzazioni a un utente IAM nella Guida per l'utente IAM.

   Si consiglia di utilizzare le chiavi di contesto delle condizioni globali `aws:SourceArn` e `aws:SourceAccount` nelle policy basate sulle risorse per limitare le autorizzazioni del servizio a una risorsa specifica. Questo è il modo più efficace per proteggersi dal problema di deputy confused.

   Se si utilizzano entrambe le chiavi di contesto delle condizioni globali e il valore `aws:SourceArn` contiene l'ID account, il valore `aws:SourceAccount` e l'account nel valore `aws:SourceArn` devono utilizzare lo stesso ID account quando viene utilizzato nella stessa dichiarazione di policy.

   - Utilizzare `aws:SourceArn` se si desidera un accesso cross-service per una singola risorsa.
   - Utilizzare `aws:SourceAccount` se si desidera consentire l'associazione di qualsiasi risorsa in tale account all'uso cross-service.

   Nella policy, assicurarsi di utilizzare la chiave di contesto della condizione globale `aws:SourceArn` con l'ARN completo della risorsa. L'esempio seguente mostra come utilizzare il comando AWS CLI per creare un ruolo denominato `rds-s3-import-role`.

   Example

   Per LinuxmacOS, oUnix:

```
aws iam create-role \
   --role-name rds-s3-import-role \
   --assume-role-policy-document '{
     "Version": "2012-10-17",
     "Statement": [
       {
         "Effect": "Allow",
         "Principal": {
            "Service": "rds.amazonaws.com"
          },
         "Action": "sts:AssumeRole",
         "Condition": {
```

```
            "StringEquals": {
               "aws:SourceAccount": "111122223333",
               "aws:SourceArn": "arn:aws:rds:us-east-1:111122223333:db:dbname"
               }
            }
      }
    ]
  }'
```

Per Windows:

```
aws iam create-role ^
   --role-name rds-s3-import-role ^
   --assume-role-policy-document '{
     "Version": "2012-10-17",
     "Statement": [
       {
         "Effect": "Allow",
         "Principal": {
            "Service": "rds.amazonaws.com"
          },
         "Action": "sts:AssumeRole",
         "Condition": {
             "StringEquals": {
                "aws:SourceAccount": "111122223333",
                "aws:SourceArn": "arn:aws:rds:us-east-1:111122223333:db:dbname"
                }
             }
       }
     ]
   }'
```

3. Collegare la policy IAM al ruolo IAM creato.

   Il comando AWS CLI seguente associa la policy creata in precedenza al ruolo denominato `rds-s3-import-role`Replace*`your-policy-arn`* con l'ARN della policy annotato nella fase precedente.

   Example

   Per LinuxmacOS, oUnix:

```
aws iam attach-role-policy \
   --policy-arn your-policy-arn \
   --role-name rds-s3-import-role
```

Per Windows:

```
aws iam attach-role-policy ^
   --policy-arn your-policy-arn ^
   --role-name rds-s3-import-role
```

4. Aggiungere il ruolo IAM all'istanza database.

   Per svolgere questa operazione, utilizzare la AWS Management Console o l'AWS CLI, come descritto di seguito.

Console

Per aggiungere un ruolo IAM all'istanza database PostgreSQL tramite la console

1. Accedi alla AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Scegliere il nome dell'istanza database PostgreSQL per visualizzarne i dettagli.
3. Nella scheda Connettività e sicurezza, nella sezione Gestisci ruoli IAM, scegli il ruolo da aggiungere in Aggiungi ruoli IAM a questa del cluster.
4. In Feature (Caratteristica), scegliere s3Import.
5. Scegliere Add role (Aggiungi ruolo).

AWS CLI

Per aggiungere un ruolo IAM a un'istanza database PostgreSQL tramite CLI

- Utilizzare il seguente comando per aggiungere il ruolo all'istanza database PostgreSQL denominata `my-db-instance`. Sostituire *`your-role-arn`* con l'ARN del ruolo annotato in precedenza. Utilizzare `s3Import` come valore dell'opzione `--feature-name`.

  Example

  Per LinuxmacOS, oUnix:

```
aws rds add-role-to-db-instance \
   --db-instance-identifier my-db-instance \
   --feature-name s3Import \
   --role-arn your-role-arn   \
   --region your-region
```

Per Windows:

```
aws rds add-role-to-db-instance ^
   --db-instance-identifier my-db-instance ^
   --feature-name s3Import ^
   --role-arn your-role-arn ^
   --region your-region
```

API RDS

Utilizzo delle credenziali di sicurezza per accedere a un bucket Amazon S3

Se preferisci, puoi utilizzare le credenziali di sicurezza per fornire accesso a un bucket Amazon S3 invece di fornire accesso con un ruolo IAM. A tale scopo, specifica il parametro `credentials` nella chiamata di funzione aws_s3.table_import_from_s3.

Il parametro `credentials` è una struttura di tipo `aws_commons._aws_credentials_1`, contenente le credenziali AWS. Utilizzare la funzione aws_commons.create_aws_credentials per impostare la chiave di accesso e la chiave segreta in una struttura `aws_commons._aws_credentials_1`, come illustrato di seguito.

```
postgres=> SELECT aws_commons.create_aws_credentials(
   'sample_access_key', 'sample_secret_key', '')
AS creds \gset
```

Dopo aver creato la struttura `aws_commons._aws_credentials_1` , utilizzare la funzione aws_s3.table_import_from_s3 con il parametro `credentials` per importare i dati, come illustrato di seguito.

```
postgres=> SELECT aws_s3.table_import_from_s3(
   't', '', '(format csv)',
```

```
   :'s3_uri',
   :'creds'
);
```

Oppure si può includere la chiamata inline di funzione aws_commons.create_aws_credentials all'interno della chiamata di funzione `aws_s3.table_import_from_s3`.

```
postgres=> SELECT aws_s3.table_import_from_s3(
   't', '', '(format csv)',
   :'s3_uri',
   aws_commons.create_aws_credentials('sample_access_key', 'sample_secret_key', '')
);
```

Risoluzione dei problemi di accesso a Amazon S3

Se riscontri problemi di connessione quanto tenti di importare i dati da Amazon S3, segui questi suggerimenti:

- Risoluzione dei problemi di identità e accesso in Amazon RDS
- Risoluzione dei problemi di Amazon S3 nella Guida per l'utente di Amazon Simple Storage Service
- Risoluzione dei problemi di Amazon S3 e IAM nella Guida per l'utente di IAM.

Importazione di dati da Amazon S3 nell'istanza database RDS per PostgreSQL

Importa i dati dal bucket Amazon S3 utilizzando la funzione `table_import_from_s3` dell'estensione aws_s3. Per informazioni di riferimento, consulta aws_s3.table_import_from_s3.

> **Note**
>
> Gli esempi seguenti utilizzano il metodo del ruolo IAM per consentire l'accesso al bucket Amazon S3. Pertanto, le chiamate della funzione `aws_s3.table_import_from_s3` non includono parametri di credenziali.

Di seguito viene illustrato un tipico esempio.

```
postgres=> SELECT aws_s3.table_import_from_s3(
   't1',
```

```
   '',
   '(format csv)',
   :'s3_uri'
);
```

I parametri sono i seguenti:

- `t1` – Il nome della tabella nell'istanza database PostgreSQL in cui copiare i dati.
- `''` – Un elenco opzionale di colonne nella tabella di database. Questo parametro può essere utilizzato per indicare quali colonne di dati S3 vanno in quali colonne della tabella. Se non viene specificata alcuna colonna, tutte le colonne vengono copiate nella tabella. Per un esempio di utilizzo di un elenco di colonne, consulta Importazione di un file Amazon S3 che utilizza un delimitatore personalizzato.
- `(format csv)` – Argomenti COPY di PostgreSQL. La procedura di copia utilizza gli argomenti e il formato del comando COPY di PostgreSQL per importare i dati. Le scelte di formato includono valori separati da virgole (CSV), come mostrato in questo esempio, testo e file binario. Il valore predefinito è testo.
- `s3_uri` – Una struttura contenente le informazioni che identificano il file Amazon S3. Per un esempio di utilizzo della funzione aws_commons.create_s3_uri per creare una struttura `s3_uri`, consulta Panoramica dell'importazione di dati dai dati di Amazon S3.

Per ulteriori informazioni su questa funzione, consulta aws_s3.table_import_from_s3.

La funzione restituisce `aws_s3.table_import_from_s3`. Per specificare altri tipi di file da importare da un bucket Amazon S3, consulta uno dei seguenti esempi.

> **Note**
>
> L'importazione di un file da 0 byte genererà un errore.

Argomenti

- Importazione di un file Amazon S3 che utilizza un delimitatore personalizzato
- Importazione di un file compresso (gzip) Amazon S3
- Importazione di un file Amazon S3 codificato

Importazione di un file Amazon S3 che utilizza un delimitatore personalizzato

Il seguente esempio mostra come importare un file che utilizza un delimitatore personalizzato. Mostra anche come controllare dove inserire i dati nella tabella di database utilizzando il parametro `column_list` della funzione aws_s3.table_import_from_s3.

In questo esempio si presuppone che le seguenti informazioni siano organizzate in colonne delimitate da pipe nel file Amazon S3.

```
1|foo1|bar1|elephant1
2|foo2|bar2|elephant2
3|foo3|bar3|elephant3
4|foo4|bar4|elephant4
...
```

Per importare un file che utilizza un delimitatore personalizzato

1. Creare una tabella nel database per i dati importati.

   ```
   postgres=> CREATE TABLE test (a text, b text, c text, d text, e text);
   ```

2. Utilizzare il seguente formato della funzione aws_s3.table_import_from_s3 per importare i dati dal file Amazon S3.

   Si può includere la chiamata inline di funzione aws_commons.create_s3_uri all'interno della chiamata di funzione `aws_s3.table_import_from_s3` per specificare il file.

   ```
   postgres=> SELECT aws_s3.table_import_from_s3(
      'test',
      'a,b,d,e',
      'DELIMITER ''|''',
      aws_commons.create_s3_uri('sampleBucket', 'pipeDelimitedSampleFile', 'us-
   east-2')
   );
   ```

I dati sono ora nella tabella nelle seguenti colonne.

```
postgres=> SELECT * FROM test;
a | b | c | d | e
---+------+---+---+------+-----------
```

```
1 | foo1 | | bar1 | elephant1
2 | foo2 | | bar2 | elephant2
3 | foo3 | | bar3 | elephant3
4 | foo4 | | bar4 | elephant4
```

Importazione di un file compresso (gzip) Amazon S3

Il seguente esempio mostra come importare da Amazon S3 un file compresso con gzip. Il file importato deve avere i seguenti metadati Amazon S3:

- Chiave: `Content-Encoding`
- Valore: `gzip`

Se carichi il file utilizzando la AWS Management Console, i metadati vengono in genere applicati dal sistema. Per informazioni sul caricamento di file in Amazon S3 utilizzando la AWS Management Console, la AWS CLI o l'API, consulta Caricamento degli oggetti nella Guida per l'utente di Amazon Simple Storage Service.

Per ulteriori informazioni sui metadati di Amazon S3 e i dettagli sui metadati forniti dal sistema, consulta la sezione Modifica dei metadati degli oggetti nella console Amazon S3 nella Guida per l'utente di Amazon Simple Storage Service.

Importare il file gzip in RDS per un'istanza database PostgreSQL, come illustrato di seguito.

```
postgres=> CREATE TABLE test_gzip(id int, a text, b text, c text, d text);
postgres=> SELECT aws_s3.table_import_from_s3(
 'test_gzip', '', '(format csv)',
 'myS3Bucket', 'test-data.gz', 'us-east-2'
);
```

Importazione di un file Amazon S3 codificato

Il seguente esempio mostra come importare da Amazon S3 un file codificato con Windows-1252

```
postgres=> SELECT aws_s3.table_import_from_s3(
 'test_table', '', 'encoding ''WIN1252''',
 aws_commons.create_s3_uri('sampleBucket', 'SampleFile', 'us-east-2')
);
```

Informazioni di riferimento sulle funzioni

Funzioni

- aws_s3.table_import_from_s3
- aws_commons.create_s3_uri
- aws_commons.create_aws_credentials

aws_s3.table_import_from_s3

Importa dati Amazon S3 in una tabella Amazon RDS. L'estensione `aws_s3` fornisce la funzione `aws_s3.table_import_from_s3`. Il valore restituito è testo.

Sintassi

I parametri richiesti sono `table_name`, `column_list` e `options`. Identificano la tabella di database e specificano il modo in cui i dati vengono copiati nella tabella

Puoi inoltre utilizzare i seguenti parametri:

- Il parametro `s3_info` specifica il file Amazon S3 da importare. Se si utilizza questo parametro, l'accesso a Amazon S3 è fornito da un ruolo IAM per l'istanza database PostgreSQL.

```
aws_s3.table_import_from_s3 (
   table_name text,
   column_list text,
   options text,
   s3_info aws_commons._s3_uri_1
)
```

- Il parametro `credentials` specifica le credenziali per accedere a Amazon S3. Se si utilizza questo parametro, non si utilizza il ruolo IAM.

```
aws_s3.table_import_from_s3 (
   table_name text,
   column_list text,
   options text,
   s3_info aws_commons._s3_uri_1,
   credentials aws_commons._aws_credentials_1
)
```

Parametri

table_name

Una stringa di testo obbligatoria contenente il nome della tabella di database PostgreSQL in cui importare i dati.

column_list

Una stringa di testo obbligatoria contenente un elenco opzionale delle colonne della tabella di database PostgreSQL nelle quali copiare i dati. Se la stringa è vuota, vengono utilizzate tutte le colonne della tabella. Per un esempio, consulta Importazione di un file Amazon S3 che utilizza un delimitatore personalizzato.

options

Una stringa di testo obbligatorio contenente gli argomenti del comando `COPY` di PostgreSQL. Tali argomenti specificano in che modo i dati vengono copiati nella tabella PostgreSQL. Per maggiori dettagli, consulta la documentazione di COPY PostgreSQL.

s3_info

Un tipo composito `aws_commons._s3_uri_1` contenente le seguenti informazioni sull'oggetto S3:

- `bucket` – Il nome del bucket Amazon S3 contenente il file.
- `file_path` – Il nome file di Amazon S3, incluso il percorso.
- `region`: la regione AWS in cui si trova il file. Per un elenco di nomi di regione AWS e dei valori associati, consulta Regioni, zone di disponibilità e Local Zones.

credenziali

Un tipo composito `aws_commons._aws_credentials_1` contenente le seguenti credenziali da utilizzare per l'operazione di importazione:

- Chiave di accesso
- Chiave segreta
- Token di sessione

Per informazioni sulla creazione di una struttura composita `aws_commons._aws_credentials_1`, consulta aws_commons.create_aws_credentials.

Sintassi alternativa

Per un aiuto nei test, si può utilizzare un set più ampio di parametri al posto dei parametri `s3_info` e `credentials`. Di seguito vengono riportate le variazioni di sintassi aggiuntive per la funzione `aws_s3.table_import_from_s3`.

- Invece di utilizzare il parametro `s3_info` per identificare un file Amazon S3, utilizzare la combinazione dei parametri `bucket`, `file_path` e `region`. Con questo formato della funzione, l'accesso a Amazon S3 viene fornito da un ruolo IAM nell'istanza database PostgreSQL.

```
aws_s3.table_import_from_s3 (
   table_name text,
   column_list text,
   options text,
   bucket text,
   file_path text,
   region text
)
```

- Invece di utilizzare il parametro `credentials` per specificare l'accesso a Amazon S3, utilizzare la combinazione dei parametri `access_key`, `session_key` e `session_token`.

```
aws_s3.table_import_from_s3 (
   table_name text,
   column_list text,
   options text,
   bucket text,
   file_path text,
   region text,
   access_key text,
   secret_key text,
   session_token text
)
```

Parametri alternativi

bucket

Una stringa di testo contenente il nome del bucket Amazon S3 che contiene il file

file_path

Una stringa di testo contenente il nome file di Amazon S3, incluso il percorso.

Regione

Una stringa di testo che identifica la posizione Regione AWS del file. Per un elenco di nomi di Regione AWS e di valori associati, consulta Regioni, zone di disponibilità e Local Zones.

chiave_accesso

Una stringa di testo contenente la chiave di accesso da utilizzare per l'operazione di importazione. Il valore predefinito è NULL.

secret_key

Una stringa di testo contenente la chiave segreta da utilizzare per l'operazione di importazione. Il valore predefinito è NULL.

session_token

(Opzionale) Una stringa di testo contenente la chiave di sessione da utilizzare per l'operazione di importazione. Il valore predefinito è NULL.

aws_commons.create_s3_uri

Crea una struttura `aws_commons._s3_uri_1` per conservare le informazioni relative al file Amazon S3. Si utilizzano i risultati della funzione `aws_commons.create_s3_uri` nel parametro `s3_info` della funzione aws_s3.table_import_from_s3.

Sintassi

```
aws_commons.create_s3_uri(
   bucket text,
   file_path text,
   region text
)
```

Parametri

bucket

Una stringa di testo obbligatoria contenente il nome del bucket Amazon S3 del file.

file_path

Una stringa di testo obbligatoria contenente il nome file di Amazon S3, incluso il percorso.

Regione

Una stringa di testo obbligatoria contenente la Regione AWS in cui si trova il file. Per un elenco di nomi di Regione AWS e di valori associati, consulta Regioni, zone di disponibilità e Local Zones.

aws_commons.create_aws_credentials

Imposta una chiave di accesso e una chiave segreta in una struttura `aws_commons._aws_credentials_1`. Si utilizzano i risultati della funzione `aws_commons.create_aws_credentials` nel parametro `credentials` della funzione aws_s3.table_import_from_s3.

Sintassi

```
aws_commons.create_aws_credentials(
   access_key text,
   secret_key text,
   session_token text
)
```

Parametri

chiave_accesso

Una stringa di testo obbligatoria contenente la chiave di accesso da utilizzare per l'importazione di un file Amazon S3. Il valore predefinito è NULL.

secret_key

Una stringa di testo obbligatoria contenente la chiave segreta da utilizzare per l'importazione di un file Amazon S3. Il valore predefinito è NULL.

session_token

Una stringa di testo opzionale contenente il token di sessione da utilizzare per l'importazione di un file Amazon S3. Il valore predefinito è NULL. Se si fornisce un `session_token` opzionale, è possibile utilizzare credenziali temporanee.

Trasporto dei database PostgreSQL tra istanze database

Utilizzando Transportable Database di PostgreSQL per Amazon RDS, puoi spostare un database PostgreSQL tra due istanze database. Si tratta di un modo molto rapido per migrare database di grandi dimensioni tra diverse istanze database. Per utilizzare questo approccio, le istanze database devono essere entrambe eseguite con la stessa versione principale di PostgreSQL.

Questa funzionalità richiede l'installazione dell'estensione `pg_transport` sull'istanza database di origine e destinazione. L'estensione `pg_transport` fornisce un meccanismo di trasporto fisico che consente di spostare i file del database con elaborazione minima. Questo meccanismo consente di spostare i dati più rapidamente rispetto ai tradizionali processi dump e load, con tempi di inattività molto ridotti.

> **Note**
>
> Transportable Database di PostgreSQL è disponibile in RDS for PostgreSQL versioni 11.5 e in RDS for PostgreSQL versioni 10.10 e successive.

Per trasportare un'istanza database PostgreSQL da un'istanza database RDS for PostgreSQL a un'altra, è necessario innanzitutto impostare le istanze di origine e di destinazione come descritto in Configurazione di un'istanza database per il trasporto. È quindi possibile trasportare il database utilizzando la funzione descritta in Trasporto di un database PostgreSQL.

Argomenti

- Limitazioni all'utilizzo di Transportable Database di PostgreSQL
- Configurazione del trasporto di un database PostgreSQL
- Trasporto di un database PostgreSQL alla destinazione dall'origine
- Cosa succede durante il trasporto del database
- Riferimento per la funzione Transportable Database
- Riferimento per i parametri di Transportable Database

Limitazioni all'utilizzo di Transportable Database di PostgreSQL

Transportable Database ha le limitazioni seguenti:

- Repliche di lettura – I database trasportabili non possono essere utilizzati su repliche di lettura o istanze padre di repliche di lettura.
- Tipi di colonne non supportati – Impossibile utilizzare i tipi di dati `reg` nelle tabelle di database che si intendono trasportare con questo metodo. Questi tipi dipendono dagli ID oggetto del catalogo di sistema (OID), che spesso vengono modificati durante il trasporto.
- Spazi tabelle – Tutti gli oggetti del database di origine devono trovarsi nello spazio tabelle `pg_default` predefinito.
- Compatibilità – Le istanze database di origine e di destinazione devono eseguire la stessa versione principale di PostgreSQL.
- Estensioni – L'istanza database di origine può avere solo `pg_transport` installato.
- Ruoli e liste di controllo accessi (ACL) – I privilegi di accesso al database di origine e le informazioni di proprietà non vengono trasferite nel database di destinazione. Tutti gli oggetti del database vengono creati e assegnati all'utente di destinazione locale del trasporto.
- Trasporti simultanei – Una singola istanza database può supportare fino a 32 trasporti simultanei, comprese le importazioni e le esportazioni, se i processi di lavoro sono stati configurati correttamente.
- Solo istanze database RDS for PostgreSQL – Sono supportati solo database trasportabili sulle istanze database RDS for PostgreSQL. Non è possibile utilizzarlo con database o database locali in esecuzione su Amazon EC2.

Configurazione del trasporto di un database PostgreSQL

Prima di iniziare, verifica che le istanze database RDS for PostgreSQL soddisfino i seguenti requisiti seguenti:

- Le istanze database RDS for PostgreSQL di origine e di destinazione devono eseguire la stessa versione di PostgreSQL.
- Il database di destinazione non può avere un database con lo stesso nome del database di origine che si desidera trasportare.
- L'account utilizzato per eseguire il trasporto necessita di privilegi `rds_superuser` sia sul database di origine che sul database di destinazione.
- Il gruppo di sicurezza per l'istanza database di origine deve consentire l'accesso in entrata dall'istanza database di destinazione. Questo potrebbe già accadere se le istanze database di origine e di destinazione si trovano nel VPC. Per ulteriori informazioni sui gruppi di sicurezza, consulta Controllo dell'accesso con i gruppi di sicurezza.

Il trasporto dei database da un'istanza database di origine a un'istanza database di destinazione richiede diverse modifiche al gruppo parametri del database associato a ciascuna istanza. Ciò significa che è necessario creare un gruppo parametri del database personalizzato per l'istanza database di origine e creare un gruppo parametri del database personalizzato per l'istanza database di destinazione.

> **Note**
>
> Se le istanze database sono già configurate utilizzando gruppi di parametri del database personalizzati, è possibile iniziare con il passaggio 2 della procedura seguente.

Per configurare il gruppo parametri del database personalizzato per il trasporto dei database

Per i seguenti passaggi, utilizza un account con privilegi `rds_superuser`.

1. Se le istanze DB di origine e di destinazione utilizzano un gruppo di parametri DB predefinito, è necessario creare un gruppo di parametri DB personalizzato utilizzando la versione appropriata per le istanze. Questa operazione consente di modificare i valori per diversi parametri. Per ulteriori informazioni, consulta Utilizzo di gruppi di parametri.
2. Nel gruppo parametri del database personalizzato, modifica i valori per i seguenti parametri:
   - `shared_preload_libraries` – Aggiungi `pg_transport` all'elenco delle librerie.
   - `pg_transport.num_workers` – Il valore predefinito è 3. Aumenta o riduci questo valore secondo necessità del tuo database. Per un database da 200 GB, consigliamo un valore non più alto di 8. Tieni presente che se aumenti il valore predefinito per questo parametro, dovresti anche aumentare il valore di `max_worker_processes`.
   - `pg_transport.work_mem` – Il valore predefinito è 128 MB o 256 MB, a seconda della versione di PostgreSQL. In genere, l'impostazione predefinita può essere lasciata invariata.
   - `max_worker_processes` - Il valore di questo parametro deve essere impostato utilizzando il seguente calcolo:

     ```
     (3 * pg_transport.num_workers) + 9
     ```

     Questo valore è necessario sulla destinazione per gestire vari processi di lavoro in background coinvolti nel trasporto. Per ulteriori informazioni su `max_worker_processes`,, consulta Consumo di risorse nella documentazione di PostgreSQL.

Per ulteriori informazioni sui parametri `pg_transport`, consulta Riferimento per i parametri di Transportable Database .

3. Riavvia l'istanza database RDS for PostgreSQL e l'istanza di destinazione in modo che le impostazioni per i parametri abbiano effetto.
4. Connettiti all'istanza database RDS for PostgreSQL di origine.

```
psql --host=source-instance.111122223333.aws-region.rds.amazonaws.com --port=5432
 --username=postgres --password
```

5. Rimuovi le estensioni estranee dallo schema pubblico dell'istanza database. Solo l'estensione `pg_transport` è consentita durante l'effettiva operazione di trasporto.
6. Installa l'estensione `pg_transport` come segue:

```
postgres=> CREATE EXTENSION pg_transport;
CREATE EXTENSION
```

7. Connettiti all'istanza database RDS for PostgreSQL di destinazione. Rimuovi eventuali estensioni estranee, quindi installa l'estensione `pg_transport`.

```
postgres=> CREATE EXTENSION pg_transport;
CREATE EXTENSION
```

Trasporto di un database PostgreSQL alla destinazione dall'origine

Dopo aver completato il processo descritto in Configurazione del trasporto di un database PostgreSQL, puoi avviare il trasporto. A questo scopo, esegui la funzione `transport.import_from_server` nell'istanza database di destinazione. Nella sintassi seguente puoi trovare i parametri della funzione.

```
SELECT transport.import_from_server(
   'source-db-instance-endpoint',
    source-db-instance-port,
   'source-db-instance-user',
   'source-user-password',
   'source-database-name',
   'destination-user-password',
   false);
```

Il valore `false` mostrato nell'esempio indica alla funzione che non si tratta di un test. Per testare la configurazione di trasporto, è possibile specificare `true` per l'opzione `dry_run` quando chiami la funzione, come illustrato di seguito:

```
postgres=> SELECT transport.import_from_server(
    'docs-lab-source-db.666666666666aws-region.rds.amazonaws.com', 5432,
    'postgres', '********', 'labdb', '******', true);
INFO:  Starting dry-run of import of database "labdb".
INFO:  Created connections to remote database        (took 0.03 seconds).
INFO:  Checked remote cluster compatibility          (took 0.05 seconds).
INFO:  Dry-run complete                           (took 0.08 seconds total).
 import_from_server
--------------------

(1 row)
```

Le linee INFO sono di output perché il parametro `pg_transport.timing` è impostato sul valore predefinito, `true`. Imposta la proprietà `dry_run` su `false` quando esegui il comando e il database di origine viene importato nella destinazione, come illustrato di seguito:

```
INFO:  Starting import of database "labdb".
INFO:  Created connections to remote database        (took 0.02 seconds).
INFO:  Marked remote database as read only           (took 0.13 seconds).
INFO:  Checked remote cluster compatibility          (took 0.03 seconds).
INFO:  Signaled creation of PITR blackout window     (took 2.01 seconds).
INFO:  Applied remote database schema pre-data       (took 0.50 seconds).
INFO:  Created connections to local cluster          (took 0.01 seconds).
INFO:  Locked down destination database              (took 0.00 seconds).
INFO:  Completed transfer of database files          (took 0.24 seconds).
INFO:  Completed clean up                            (took 1.02 seconds).
INFO:  Physical transport complete                (took 3.97 seconds total).
import_from_server
--------------------
(1 row)
```

Questa funzione richiede l'inserimento di password per l'utente del database. Quindi, ti consigliamo di modificare le password dei ruoli utente utilizzati dopo aver completato il trasporto. In alternativa, puoi utilizzare le variabili di associazione SQL per creare ruoli utente temporanei. Utilizza questi ruoli temporanei per il trasporto, quindi eliminali al termine.

Se il trasporto non ha esito positivo, potrebbe essere visualizzato un messaggio di errore simile al seguente:

```
pg_transport.num_workers=8 25% of files transported failed to download file data
```

Il messaggio di errore "Impossibile scaricare dati file" indica che il numero di processi di lavoro non è impostato correttamente per le dimensioni del database. Potrebbe essere necessario aumentare o diminuire il valore impostato per `pg_transport.num_workers`. Ogni errore segnala la percentuale di completamento, in modo da poter vedere l'impatto delle modifiche. Ad esempio, la modifica dell'impostazione da 8 a 4 in un caso ha comportato quanto segue:

```
pg_transport.num_workers=4 75% of files transported failed to download file data
```

Ricorda che il parametro `max_worker_processes` viene preso in considerazione anche durante il processo di trasporto. In altre parole, potrebbe essere necessario modificare sia `pg_transport.num_workers` che `max_worker_processes` per trasportare correttamente il database. L'esempio mostrato ha finalmente funzionato quando `pg_transport.num_workers` è stato impostato su 2:

```
pg_transport.num_workers=2 100% of files transported
```

Per ulteriori informazioni sulla funzione `transport.import_from_server` e sui relativi parametri, consulta Riferimento per la funzione Transportable Database.

Cosa succede durante il trasporto del database

La funzione Transportable Database di PostgreSQL utilizza un modello pull per importare il database dall'istanza database di origine alla destinazione. La funzione `transport.import_from_server` crea il database in transito nell'istanza database di destinazione. Il database in transito non è accessibile nell'istanza database di destinazione per tutta la durata del trasporto.

All'avvio del trasporto, tutte le sessioni correnti nel database di origine vengono terminate. Nessun database, oltre al database di origine nell'istanza database di origine, viene interessato dal trasporto.

Il database di origine passa in una modalità speciale di sola lettura. In questa modalità, puoi connetterti al database di origine ed eseguire query di sola lettura. Invece, le query abilitate per la scrittura e alcuni altri tipi di comandi sono bloccati. Solo lo specifico database di origine trasportato è sottoposto a queste limitazioni.

Durante il trasporto, non puoi ripristinare l'istanza database di destinazione a un point-in-time perché il trasporto non è transazionale e non utilizza il log write-ahead (WAL) di PostgreSQL per registrare le modifiche. Se nell'istanza database di destinazione sono abilitati i backup automatici, dopo il trasporto viene eseguito automaticamente un backup. I oint-in-time ripristini P sono disponibili per alcuni periodi successivi al termine del backup.

Se il trasporto non riesce, l'estensione `pg_transport` tenta di annullare tutte le modifiche alle istanze database di origine e di destinazione, inclusa la rimozione del database parzialmente trasportato nella destinazione. A seconda del tipo di errore, il database di origine potrebbe continuare a rifiutare le query abilitate per la scrittura. Se accade, utilizza il comando seguente per consentirle.

```
ALTER DATABASE db-name SET default_transaction_read_only = false;
```

Riferimento per la funzione Transportable Database

La funzione `transport.import_from_server` trasporta un database PostgreSQL importandolo da un'istanza database di origine a un'istanza database di destinazione. Per farlo, utilizza un meccanismo di trasporto con connessione al database fisico.

Prima di iniziare il trasporto, questa funzione verifica che le istanze database di origine e di destinazione siano della stessa versione e siano compatibili per la migrazione. Conferma inoltre che l'istanza database di destinazione abbia spazio sufficiente per l'origine.

Sintassi

```
transport.import_from_server(
   host text,
   port int,
   username text,
   password text,
   database text,
   local_password text,
   dry_run bool
)
```

Valore restituito

Nessuna.

Parametri

Le descrizioni dei parametri della funzione `transport.import_from_server` sono disponibili nella tabella seguente.

Parametro	Descrizione
`host`	L'endpoint dell'istanza database di origine.
`port`	Un numero intero che rappresenta la porta dell'istanza database di origine. Le istanze database di PostgreSQL spesso utilizzano la porta 5432.
`username`	L'utente dell'istanza database di origine. Questo utente deve essere membro del ruolo `rds_superuser` .
`password`	La password utente dell'istanza database di origine.
`database`	Il nome del database nell'istanza database di origine da trasportare.
`local_password`	La password locale dell'utente corrente per l'istanza database di destinazione. Questo utente deve essere membro del ruolo `rds_superuser` .
`dry_run`	Un valore booleano facoltativo che specifica se eseguire un test. L'impostazione predefinita è `false`, che indica che il trasporto procede. Per confermare la compatibilità tra le istanze database di origine e di destinazione senza eseguire effettivamente il trasporto, imposta `dry_run` su `true`.

Esempio

Per un esempio, consulta Trasporto di un database PostgreSQL alla destinazione dall'origine.

Riferimento per i parametri di Transportable Database

Diversi parametri controllano il comportamento dell'estensione `pg_transport`. Di seguito, sono disponibili le descrizioni di questi parametri.

`pg_transport.num_workers`

Il numero di dipendenti da utilizzare per il processo di trasporto. L'impostazione predefinita è 3. I valori validi sono 1–32. Anche i trasporti di database più grandi in genere richiedono un numero

di dipendenti inferiore a 8. Il valore di questa impostazione sull'istanza database di destinazione viene utilizzato sia dall'origine che dalla destinazione durante il trasporto.

`pg_transport.timing`

Specifica se riportare le informazioni temporali durante il trasporto. Il valore predefinito è `true`, il che significa che le informazioni temporali vengono riportate. Consigliamo di lasciare questo parametro impostato su `true` in modo da poter monitorare i progressi. Per output di esempio, vedi Trasporto di un database PostgreSQL alla destinazione dall'origine.

`pg_transport.work_mem`

La quantità massima di memoria da allocare per ogni processo di lavoro. Il valore predefinito è 131072 kilobyte (KB) o 262144 KB (256 MB), a seconda della versione di PostgreSQL. Il valore minimo è 64 megabyte (65536 KB). I valori validi sono unità in base 2 binarie espresse in kilobyte (KB), dove 1 KB = 1024 byte.

Il trasporto potrebbe utilizzare meno memoria rispetto a quella specificata in questo parametro. Anche i trasporti di database di dimensioni maggiori in genere richiedono meno di 256 MB (262144 KB) di memoria per dipendente.

Esportazione di dati da un'istanza di database del RDS per PostgreSQL a Amazon S3

È possibile eseguire query sui dati da un'istanza di database del RDS per PostgreSQL ed esportarli direttamente in file memorizzati in un bucket Amazon S3. A questo scopo, installa innanzitutto l'estensione RDS per PostgreSQL `aws_s3`. Questa estensione fornisce le funzioni utilizzate per esportare i risultati delle quey in Amazon S3. Di seguito, sono disponibili informazioni su come installare l'estensione ed esportare i dati in Amazon S3.

Puoi esportare da un'istanza database con provisioning o Aurora Serverless v2. Questi passaggi non sono supportati per Aurora Serverless v1.

> **Note**
>
> L'esportazione tra account in Amazon S3 non è supportata.

Tutte le versioni attualmente disponibili di RDS per PostgreSQL supportano l'esportazione dei dati in Servizio di archiviazione semplice Amazon. Per informazioni dettagliate sulla versione, consulta gli aggiornamenti di Amazon RDS per PostgreSQL nelle Note di rilascio di Amazon RDS per PostgreSQL.

Se non disponi di un bucket configurato per l'esportazione, consulta i seguenti argomenti nella Guida per l'utente di Servizio di archiviazione semplice Amazon.

- Configurazione di Amazon S3
- Creazione di un bucket

Per impostazione predefinita, i dati esportati da RDS per PostgreSQL ad Amazon S3 utilizzano la crittografia lato server con un. Chiave gestita da AWS Se utilizzi la crittografia a bucket, il bucket Amazon S3 deve essere crittografato con AWS Key Management Service una chiave AWS KMS() (SSE-KMS). Attualmente, i bucket crittografati con chiavi gestite di Amazon S3 (SSE-S3) non sono supportati.

> **Note**
>
> Puoi salvare i dati degli snapshot DB su Amazon S3 utilizzando AWS CLI, o AWS Management Console l'API Amazon RDS. Per ulteriori informazioni, consulta Esportazione dei dati dello snapshot DB in Simple Storage Service (Amazon S3).

Argomenti

Installazione dell'estensione aws_s3

Prima di poter utilizzare Servizio di archiviazione semplice Amazon con l'istanza database RDS per PostgreSQL, è necessario installare l'estensione `aws_s3`. Questa estensione fornisce funzioni per l'esportazione di dati un'istanza database RDS per PostgreSQL in un bucket Amazon S3. Fornisce inoltre funzioni per l'importazione dei dati da Amazon S3. Per ulteriori informazioni, consulta Importazione di dati da Amazon S3 in un'istanza database RDS per PostgreSQL. L'estensione `aws_s3` dipende da alcune delle funzioni helper nell'estensione `aws_commons`, che vengono installate automaticamente quando necessario.

Per installare l'estensione **`aws_s3`**

1. Usa psql (o pgAdmin) per connetterti all'istanza database RDS per PostgreSQL come un utente che dispone di privilegi `rds_superuser`. Se hai mantenuto il nome predefinito durante il processo di configurazione, esegui la connessione come `postgres`.

   ```
   psql --host=111122223333.aws-region.rds.amazonaws.com --port=5432 --
   username=postgres --password
   ```

2. Per installare l'estensione, esegui il comando seguente.

```
postgres=> CREATE EXTENSION aws_s3 CASCADE;
NOTICE: installing required extension "aws_commons"
CREATE EXTENSION
```

3. Per verificare che l'estensione sia installata, puoi usare il metacomando psql `\dx`.

```
postgres=> \dx
       List of installed extensions
    Name     | Version |   Schema   |                 Description
-------------+---------+------------+---------------------------------------------
 aws_commons | 1.2     | public     | Common data types across AWS services
 aws_s3      | 1.1     | public     | AWS S3 extension for importing data from S3
 plpgsql     | 1.0     | pg_catalog | PL/pgSQL procedural language
(3 rows)
```

Le funzioni per importare dati da Amazon S3 ed esportare dati in Amazon S3 sono ora disponibili per l'uso.

Verifica che la versione di RDS per PostgreSQL supporti le esportazioni in Amazon S3

Per verificare che la versione di RDS per PostgreSQL supporti l'esportazione in Amazon S3, puoi utilizzare il comando `describe-db-engine-versions`. Nell'esempio seguente viene descritto come verificare il supporto per la versione 10.14.

```
aws rds describe-db-engine-versions --region us-east-1
--engine postgres --engine-version 10.14 | grep s3Export
```

Se l'output include la stringa `"s3Export"`, allora il motore supporta le esportazioni Amazon S3. In caso contrario, il motore non le supporta.

Panoramica dell'esportazione di dati in Amazon S3

Per esportare i dati archiviati in un database RDS per PostgreSQL verso un bucket Amazon S3, attenersi alla procedura descritta di seguito.

Per esportare i dati RDS per PostgreSQL in S3.

1. Identifica un percorso del file Amazon S3 da utilizzare per l'esportazione dei dati. Per informazioni dettagliate su questo processo, consulta Specifica del percorso del file Amazon S3 in cui eseguire l'esportazione.
2. Fornisci l'autorizzazione ad accedere al bucket Amazon S3.

   Per esportare i dati in un file Amazon S3, concedi al RDS per istanza database PostgreSQL l'autorizzazione per accedere al bucket Amazon S3 che verrà utilizzato dall'esportazione per lo storage. Questa operazione include le seguenti fasi:

   1. Crea una policy IAM che fornisce l'accesso a un bucket Amazon S3 in cui desideri eseguire l'esportazione.
   2. Creare un ruolo IAM.
   3. Collega la policy creata al ruolo creato.
   4. Aggiungi questo ruolo IAM al istanza di database.

   Per informazioni dettagliate su questo processo, consulta Configurazione dell'accesso a un bucket Amazon S3.
3. Identifica una query del database per ottenere i dati. Esporta i dati della query chiamando la funzione `aws_s3.query_export_to_s3`.

   Dopo aver completato le attività di preparazione precedenti, utilizza la funzione aws_s3.query_export_to_s3 per esportare i risultati della query in Amazon S3. Per informazioni dettagliate su questo processo, consulta Esportazione dei dati della query utilizzando la funzione aws_s3.query_export_to_s3.

Specifica del percorso del file Amazon S3 in cui eseguire l'esportazione

Specifica le seguenti informazioni per identificare la posizione in Amazon S3 in cui desideri esportare i dati:

- Nome del bucket – Un bucket è un container di oggetti o file Amazon S3.

  Per ulteriori informazioni sull'archiviazione dei dati con Amazon S3, consulta Creazione di un bucket e Visualizzazione di un oggetto nella Guida per l'utente di Amazon Simple Storage Service.

- Percorso del file – Il percorso del file identifica la posizione di archiviazione dell'esportazione nel bucket Amazon S3. Il percorso del file comprende:
  - Un prefisso del percorso facoltativo che identifica un percorso di cartella virtuale.
  - Un prefisso del file che identifica uno o più file da archiviare. Esportazioni di dimensioni maggiori vengono archiviate in più file, ciascuno con una dimensione massima di circa 6 GB. I nomi di file aggiuntivi hanno lo stesso prefisso di file ma con l'aggiunta di `_part`*XX*. *XX* rappresenta 2, poi 3 e così via.

  Ad esempio, un percorso del file con una cartella `exports` e un prefisso del file `query-1-export` è `/exports/query-1-export`.
- AWS Regione (opzionale): la AWS regione in cui si trova il bucket Amazon S3.

> **Note**
>
> Per un elenco dei nomi delle AWS regioni e dei valori associati, vedereRegioni, zone di disponibilità e Local Zones.

Per conservare le informazioni sul file Amazon S3 relative alla posizione di archiviazione dell'esportazione, puoi utilizzare la funzione aws_commons.create_s3_uri per creare una struttura composita `aws_commons._s3_uri_1` come descritto di seguito.

```
psql=> SELECT aws_commons.create_s3_uri(
   'sample-bucket',
   'sample-filepath',
   'us-west-2'
) AS s3_uri_1 \gset
```

In seguito fornisci questo valore `s3_uri_1` come un parametro nella chiamata alla funzione aws_s3.query_export_to_s3. Per alcuni esempi, consulta Esportazione dei dati della query utilizzando la funzione aws_s3.query_export_to_s3.

Configurazione dell'accesso a un bucket Amazon S3

Per esportare i dati in Amazon S3, concedi all'istanza del di database PostgreSQL l'autorizzazione per accedere al bucket Amazon S3 di destinazione dei file.

A tale scopo, procedi come indicato di seguito.

Per concedere a un'istanza database PostgreSQL l'accesso ad Amazon S3 tramite un ruolo IAM

1. Creare una policy IAM

   Questa policy fornisce le autorizzazioni bucket e di oggetto che consentono all'istanza del di database PostgreSQL di accedere a Amazon S3.

   Come parte della creazione di questa policy, attenersi alla seguente procedura:

   a. Includere nella policy le seguenti operazioni obbligatorie per consentire il trasferimento dei file dall'istanza del del database PostgreSQL a un bucket Amazon S3:

      - `s3:PutObject`
      - `s3:AbortMultipartUpload`

   b. Includere l'Amazon Resource Name (ARN) che identifica il bucket Amazon S3 e gli oggetti nel bucket. Il formato ARN per l'accesso a Amazon S3 è: `arn:aws:s3:::`*`your-s3-bucket`*`/*`

   Per ulteriori informazioni sulla creazione di una policy IAM per Amazon RDS for PostgreSQL, consulta Creazione e utilizzo di una policy IAM per l'accesso al database IAM. Consulta anche il Tutorial: Creare e collegare la prima policy gestita dal cliente nella Guida per l'utente di IAM.

   Il AWS CLI comando seguente crea una policy IAM denominata `rds-s3-export-policy` con queste opzioni. Concede l'accesso a un bucket denominato `your-s3-bucket`.

   > **Warning**
   >
   > Si consiglia di impostare il database all'interno di un VPC privato con policy di endpoint configurate per accedere a bucket specifici. Per ulteriori informazioni, consulta Utilizzo delle policy dell'endpoint per Amazon S3 nella Guida per l'utente di Amazon VPC.
   > Si consiglia di non creare una policy con accesso a tutte le risorse. Questo accesso può rappresentare una minaccia per la sicurezza dei dati. Se si crea una policy che consente a `S3:PutObject` di accedere a tutte le risorse utilizzando `"Resource":"*"`, un utente con privilegi di esportazione può esportare i dati in tutti i bucket dell'account. Inoltre, l'utente può esportare i dati in qualsiasi bucket pubblicamente scrivibile all'interno della regione AWS .

Dopo aver creato la policy, annotarne l'Amazon Resource Name (ARN). Per la fase successiva, in cui si associa la policy a un ruolo IAM, è necessario l'ARN.

```
aws iam create-policy  --policy-name rds-s3-export-policy  --policy-document '{
     "Version": "2012-10-17",
     "Statement": [
       {
         "Sid": "s3export",
         "Action": [
           "s3:PutObject",
           "s3:AbortMultipartUpload"
         ],
         "Effect": "Allow",
         "Resource": [
           "arn:aws:s3:::your-s3-bucket/*"
         ]
       }
     ]
   }'
```

2. Creare un ruolo IAM.

   In questo modo, Amazon RDS può assumere questo ruolo IAM per tuo conto, per accedere ai bucket Amazon S3. Per ulteriori informazioni, consulta la pagina relativa alla creazione di un ruolo per delegare le autorizzazioni a un utente IAM nella Guida per l'utente IAM.

   Si consiglia di utilizzare le chiavi di contesto delle condizioni globali `aws:SourceArn` e `aws:SourceAccount` nelle policy basate sulle risorse per limitare le autorizzazioni del servizio a una risorsa specifica. Questo è il modo più efficace per proteggersi dal problema di deputy confused.

   Se si utilizzano entrambe le chiavi di contesto delle condizioni globali e il valore `aws:SourceArn` contiene l'ID account, il valore `aws:SourceAccount` e l'account nel valore `aws:SourceArn` devono utilizzare lo stesso ID account quando viene utilizzato nella stessa dichiarazione di policy.

   - Utilizzare `aws:SourceArn` se si desidera un accesso cross-service per una singola risorsa.
   - Utilizzare `aws:SourceAccount` se si desidera consentire l'associazione di qualsiasi risorsa in tale account all'uso cross-service.

Nella policy, assicurarsi di utilizzare la chiave di contesto della condizione globale `aws:SourceArn` con l'ARN completo della risorsa. L'esempio seguente mostra come eseguire questa operazione utilizzando il AWS CLI comando per creare un ruolo denominato`rds-s3-export-role`.

Example

Per LinuxmacOS, oUnix:

```
aws iam create-role  \
    --role-name rds-s3-export-role  \
    --assume-role-policy-document '{
     "Version": "2012-10-17",
     "Statement": [
       {
         "Effect": "Allow",
         "Principal": {
            "Service": "rds.amazonaws.com"
          },
         "Action": "sts:AssumeRole",
         "Condition": {
             "StringEquals": {
                "aws:SourceAccount": "111122223333",
                "aws:SourceArn": "arn:aws:rds:us-east-1:111122223333:db:dbname"
                }
             }
       }
     ]
   }'
```

Per Windows:

```
aws iam create-role  ^
    --role-name rds-s3-export-role  ^
    --assume-role-policy-document '{
     "Version": "2012-10-17",
     "Statement": [
       {
         "Effect": "Allow",
         "Principal": {
```

```
            "Service": "rds.amazonaws.com"
          },
          "Action": "sts:AssumeRole",
          "Condition": {
              "StringEquals": {
                 "aws:SourceAccount": "111122223333",
                 "aws:SourceArn": "arn:aws:rds:us-east-1:111122223333:db:dbname"
                 }
              }
       }
     ]
   }'
```

3. Collegare la policy IAM al ruolo IAM creato.

   Il AWS CLI comando seguente associa la policy creata in precedenza al ruolo denominato `rds-s3-export-role`. Replace *`your-policy-arn`* con l'ARN della policy annotato in un passaggio precedente.

   ```
   aws iam attach-role-policy  --policy-arn your-policy-arn  --role-name rds-s3-
   export-role
   ```

4. Aggiungere il ruolo IAM all'istanza database. A tale scopo, utilizzare AWS Management Console o AWS CLI, come descritto di seguito.

Console

Per aggiungere un ruolo IAM all'istanza database PostgreSQL tramite la console

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Scegliere il nome dell'istanza database PostgreSQL per visualizzarne i dettagli.
3. Nella scheda Connectivity & security (Connettività e sicurezza), nella sezione Manage IAM roles (Gestisci ruoli IAM), selezionare il ruolo da aggiungere sotto Add IAM roles to this instance (Aggiungi ruoli IAM a questa istanza).
4. In Feature (Caratteristica), scegliere s3Export.
5. Scegliere Add role (Aggiungi ruolo).

AWS CLI

Per aggiungere un ruolo IAM a un'istanza database PostgreSQL tramite CLI

- Utilizzare il seguente comando per aggiungere il ruolo all'istanza database PostgreSQL denominata `my-db-instance`. Sostituire *`your-role-arn`* con l'ARN del ruolo annotato in precedenza. Utilizzare `s3Export` come valore dell'opzione `--feature-name`.

  Example

  Per LinuxmacOS, oUnix:

  ```
  aws rds add-role-to-db-instance \
     --db-instance-identifier my-db-instance \
     --feature-name s3Export \
     --role-arn your-role-arn   \
     --region your-region
  ```

  Per Windows:

  ```
  aws rds add-role-to-db-instance ^
     --db-instance-identifier my-db-instance ^
     --feature-name s3Export ^
     --role-arn your-role-arn ^
     --region your-region
  ```

Esportazione dei dati della query utilizzando la funzione aws_s3.query_export_to_s3

Esporta i dati PostgreSQL in Amazon S3 chiamando la funzione aws_s3.query_export_to_s3.

Argomenti

- Prerequisiti
- Chiamare aws_s3.query_export_to_s3
- Esportazione in un file CSV che utilizza un delimitatore personalizzato
- Esportazione in un file binario con codifica

Prerequisiti

Prima di utilizzare la funzione `aws_s3.query_export_to_s3`, assicurati di completare i seguenti prerequisiti:

- Installa le estensioni PostgreSQL richieste come descritto in Panoramica dell'esportazione di dati in Amazon S3.
- Determina dove esportare i dati in Amazon S3 come descritto in Specifica del percorso del file Amazon S3 in cui eseguire l'esportazione.
- Assicurati che l'istanza database abbia accesso di esportazione a Amazon S3 come descritto in Configurazione dell'accesso a un bucket Amazon S3.

Gli esempi seguenti utilizzano una tabella del database denominata `sample_table`. Questi esempi esportano i dati in un bucket denominato `sample-bucket`. La tabella e i dati di esempio vengono creati con le seguenti istruzioni SQL in psql.

```
psql=> CREATE TABLE sample_table (bid bigint PRIMARY KEY, name varchar(80));
psql=> INSERT INTO sample_table (bid,name) VALUES (1, 'Monday'), (2,'Tuesday'), (3,
 'Wednesday');
```

Chiamare aws_s3.query_export_to_s3

Di seguito vengono illustrati le modalità di base per chiamare la funzione aws_s3.query_export_to_s3.

In questi esempi viene utilizzata la variabile `s3_uri_1` per identificare una struttura contenente le informazioni che identificano il file Amazon S3. Utilizzare la funzione aws_commons.create_s3_uri per creare la struttura.

```
psql=> SELECT aws_commons.create_s3_uri(
   'sample-bucket',
   'sample-filepath',
   'us-west-2'
) AS s3_uri_1 \gset
```

Anche se i parametri variano per le due chiamate di funzione `aws_s3.query_export_to_s3` seguenti, i risultati sono gli stessi per questi esempi. Tutte le righe della tabella `sample_table` vengono esportate in un bucket denominato `sample-bucket`.

```
psql=> SELECT * FROM aws_s3.query_export_to_s3('SELECT * FROM
 sample_table', :'s3_uri_1');

psql=> SELECT * FROM aws_s3.query_export_to_s3('SELECT * FROM
 sample_table', :'s3_uri_1', options :='format text');
```

I parametri sono descritti come segue:

- `'SELECT * FROM sample_table'` – Il primo parametro è una stringa di testo obbligatoria contenente una query SQL. Il motore PostgreSQL esegue questa query. I risultati della query vengono copiati nel bucket S3 identificato in altri parametri.
- `:'s3_uri_1'` – Questo parametro è una struttura che identifica il file Amazon S3. In questo esempio viene utilizzata una variabile per identificare la struttura creata in precedenza. È invece possibile creare la struttura includendo la chiamata di funzione `aws_commons.create_s3_uri` in linea all'interno della chiamata di funzione `aws_s3.query_export_to_s3` come segue.

  ```
  SELECT * from aws_s3.query_export_to_s3('select * from sample_table',
     aws_commons.create_s3_uri('sample-bucket', 'sample-filepath', 'us-west-2')
  );
  ```
- `options :='format text'` – Il parametro `options` è una stringa di testo opzionale contenente argomenti `COPY` PostgreSQL. La procedura di copia utilizza gli argomenti e il formato del comando COPY di PostgreSQL.

Se il file specificato non esiste nel bucket Amazon S3, viene creato. Se il file esiste già, viene sovrascritto. La sintassi per accedere ai dati esportati Amazon S3 è la seguente.

```
s3-region://bucket-name[/path-prefix]/file-prefix
```

Esportazioni di dimensioni maggiori vengono archiviate in più file, ciascuno con una dimensione massima di circa 6 GB. I nomi di file aggiuntivi hanno lo stesso prefisso di file ma con l'aggiunta di `_part`*XX*. *XX* rappresenta 2, poi 3 e così via. Ad esempio, supponi di specificare il percorso in cui archivi i file di dati come segue.

```
s3-us-west-2://my-bucket/my-prefix
```

Se l'esportazione deve creare tre file di dati, il bucket Amazon S3 contiene i seguenti file di dati.

```
s3-us-west-2://my-bucket/my-prefix
s3-us-west-2://my-bucket/my-prefix_part2
s3-us-west-2://my-bucket/my-prefix_part3
```

Per il riferimento completo per questa funzione e altri modi per chiamarla, consulta aws_s3.query_export_to_s3. Per ulteriori informazioni sull'accesso ai file in Amazon S3, consulta Visualizzazione di un oggetto nella Guida per l'utente di Amazon Simple Storage Service.

Esportazione in un file CSV che utilizza un delimitatore personalizzato

Nell'esempio seguente viene illustrato come chiamare la funzione aws_s3.query_export_to_s3 per esportare i dati in un file che utilizza un delimitatore personalizzato. Nell'esempio vengono utilizzati gli argomenti del comando PostgreSQL COPY per specificare il formato CSV (valori delimitati da virgole) e un delimitatore : (due punti).

```
SELECT * from aws_s3.query_export_to_s3('select * from basic_test', :'s3_uri_1',
 options :='format csv, delimiter $$:$$');
```

Esportazione in un file binario con codifica

Nell'esempio seguente viene illustrato come chiamare la funzione aws_s3.query_export_to_s3 per esportare i dati in un file binario con codifica Windows-1253.

```
SELECT * from aws_s3.query_export_to_s3('select * from basic_test', :'s3_uri_1',
 options :='format binary, encoding WIN1253');
```

Risoluzione dei problemi di accesso a Amazon S3

Se si verificano problemi di connessione durante il tentativo di esportare i dati in Amazon S3, conferma innanzi tutto che le regole di accesso in uscita per il gruppo di sicurezza VPC associato all'istanza DB consentano la connettività di rete. In particolare, il gruppo di sicurezza deve disporre di una regola che consenta all'istanza database di inviare traffico TCP alla porta 443 e a qualsiasi indirizzo IPv4 (0.0.0.0/0). Per ulteriori informazioni, consulta Fornisci accesso alla istanza database nel VPC creando un gruppo di sicurezza.

Consulta anche quanto segue per i suggerimenti:

- Risoluzione dei problemi di identità e accesso in Amazon RDS
- Risoluzione dei problemi di Amazon S3 nella Guida per l'utente di Amazon Simple Storage Service

- Risoluzione dei problemi di Amazon S3 e IAM nella Guida per l'utente di IAM.

Informazioni di riferimento sulle funzioni

Funzioni

- aws_s3.query_export_to_s3
- aws_commons.create_s3_uri

aws_s3.query_export_to_s3

Esporta un risultato della query PostgreSQL in un bucket Amazon S3. L'estensione `aws_s3` fornisce la funzione `aws_s3.query_export_to_s3`.

I due parametri richiesti sono `query` e `s3_info`. Questi definiscono la query da esportare e identificano il bucket Amazon S3 in cui eseguire l'esportazione. Un parametro opzionale chiamato `options` fornisce la definizione di vari parametri di esportazione. Per esempi di utilizzo della funzione `aws_s3.query_export_to_s3`, consulta Esportazione dei dati della query utilizzando la funzione aws_s3.query_export_to_s3.

Sintassi

```
aws_s3.query_export_to_s3(
    query text,
    s3_info aws_commons._s3_uri_1,
    options text,
    kms_key text
)
```

Parametri di input

query

Una stringa di testo obbligatoria contenente una query SQL eseguita dal motore PostgreSQL. I risultati di questa query vengono copiati in un bucket S3 identificato nel parametro `s3_info`.

s3_info

Un tipo composito `aws_commons._s3_uri_1` contenente le seguenti informazioni sull'oggetto S3:

- `bucket` – Il nome del bucket Amazon S3 per contenere il file.

- `file_path` – Il nome e il percorso del file Amazon S3.
- `region`— La AWS regione in cui si trova il bucket. Per un elenco dei nomi delle AWS regioni e dei valori associati, vedereRegioni, zone di disponibilità e Local Zones.

  Attualmente, questo valore deve essere la stessa AWS regione dell'istanza DB del che esporta. L'impostazione predefinita è la AWS regione dell'istanza DB del che esporta.

Per creare una struttura composita `aws_commons._s3_uri_1`, consulta la funzione aws_commons.create_s3_uri.

options

Una stringa di testo opzionale contenente gli argomenti del comando `COPY` di PostgreSQL. Questi argomenti specificano come i dati devono essere copiati quando vengono esportati. Per maggiori dettagli, consulta la documentazione di COPY PostgreSQL.

Parametri di input alternativi

Per facilitare il testing, puoi utilizzare un set esteso di parametri al posto del parametro `s3_info`. Di seguito vengono riportate le variazioni di sintassi aggiuntive per la funzione `aws_s3.query_export_to_s3`.

Invece di utilizzare il parametro `s3_info` per identificare un file Amazon S3, utilizzare la combinazione dei parametri `bucket`, `file_path` e `region`.

```
aws_s3.query_export_to_s3(
    query text,
    bucket text,
    file_path text,
    region text,
    options text,
)
```

query

Una stringa di testo obbligatoria contenente una query SQL eseguita dal motore PostgreSQL. I risultati di questa query vengono copiati in un bucket S3 identificato nel parametro `s3_info`.

bucket

Una stringa di testo obbligatoria contenente il nome del bucket Amazon S3 che contiene il file

file_path

Una stringa di testo obbligatoria contenente il nome file di Amazon S3, incluso il percorso.

Regione

Una stringa di testo opzionale contenente la AWS regione in cui si trova il bucket. Per un elenco dei nomi delle AWS regioni e dei valori associati, vedereRegioni, zone di disponibilità e Local Zones.

Attualmente, questo valore deve essere la stessa AWS regione dell'istanza DB del che esporta. L'impostazione predefinita è la AWS regione dell'istanza DB del che esporta.

options

Una stringa di testo opzionale contenente gli argomenti del comando `COPY` di PostgreSQL. Questi argomenti specificano come i dati devono essere copiati quando vengono esportati. Per maggiori dettagli, consulta la documentazione di COPY PostgreSQL.

Parametri di output

```
aws_s3.query_export_to_s3(
    OUT rows_uploaded bigint,
    OUT files_uploaded bigint,
    OUT bytes_uploaded bigint
)
```

rows_uploaded

Il numero di righe della tabella che sono state caricate correttamente in Amazon S3 per la query specificata.

files_uploaded

Il numero di file caricati in Amazon S3. I file vengono creati in dimensioni di circa 6 GB. A ogni file aggiuntivo creato è stato aggiunto `_part`*XX* al nome. *XX* rappresenta 2, poi 3 e così via, se necessario.

bytes_uploaded

Il numero totale di byte caricati in Amazon S3.

Esempi

```
psql=> SELECT * from aws_s3.query_export_to_s3('select * from sample_table', 'sample-
bucket', 'sample-filepath');
psql=> SELECT * from aws_s3.query_export_to_s3('select * from sample_table', 'sample-
bucket', 'sample-filepath','us-west-2');
psql=> SELECT * from aws_s3.query_export_to_s3('select * from sample_table', 'sample-
bucket', 'sample-filepath','us-west-2','format text');
```

aws_commons.create_s3_uri

Crea una struttura `aws_commons._s3_uri_1` per conservare le informazioni relative al file Amazon S3. I risultati della funzione `aws_commons.create_s3_uri` vengono utilizzati nel parametro `s3_info` della funzione aws_s3.query_export_to_s3. Per un esempio di utilizzo della funzione `aws_commons.create_s3_uri`, consulta Specifica del percorso del file Amazon S3 in cui eseguire l'esportazione.

Sintassi

```
aws_commons.create_s3_uri(
   bucket text,
   file_path text,
   region text
)
```

Parametri di input

bucket

Una stringa di testo obbligatoria contenente il nome del bucket Amazon S3 del file.

file_path

Una stringa di testo obbligatoria contenente il nome file di Amazon S3, incluso il percorso.

Regione

Una stringa di testo obbligatoria contenente la AWS regione in cui si trova il file. Per un elenco dei nomi delle AWS regioni e dei valori associati, vedereRegioni, zone di disponibilità e Local Zones.

Richiamo di una AWS Lambda funzione da un'istanza RDS del cluster

AWS Lambda è un servizio di elaborazione basato sugli eventi che consente di eseguire codice senza fornire o gestire server. È disponibile per l'uso con molti AWS servizi, tra cui . Ad esempio, è possibile utilizzare le funzioni Lambda per elaborare le notifiche di eventi da un database o per caricare dati da file ogni volta che un nuovo file viene caricato su Amazon S3. Per ulteriori informazioni su Lambda, vedi Cos'è? AWS Lambda nella Guida per gli AWS Lambda sviluppatori.

> **Note**
>
> L'invocazione di una AWS Lambda funzione è supportata in queste versioni di RDS per PostgreSQL:
>
> - Tutte le versioni di PostgreSQL 16
> - Tutte le versioni di PostgreSQL 15
> - PostgreSQL 14.1 e versioni secondarie successive
> - PostgreSQL 13.2 e versioni secondarie successive
> - PostgreSQL 12.6 e versioni secondarie successive

Di seguito sono riportati i riepiloghi dei passaggi necessari.

Per ulteriori informazioni sulle funzioni Lambda, consulta Nozioni di base su Lambda e Fondamenti di AWS Lambda nella Guida per gli sviluppatori di AWS Lambda .

Argomenti

- Messaggi di errore della funzione Lambda
- AWS Lambdariferimento a funzioni e parametri

Fase 1: configurare l'istanza DB RDS del per le connessioni in uscita a AWS Lambda

Le funzioni Lambda vengono sempre eseguite all'interno di un Amazon VPC di proprietà del servizio. AWS Lambda Lambda applica le regole di accesso alla rete e di sicurezza a questo VPC e mantiene e monitora il VPC automaticamente. L'istanza database RDS per PostgreSQL deve inviare traffico di rete al VPC del servizio Lambda. La modalità di configurazione dipende dal fatto che l'istanza database sia pubblica o privata.

- : l'istanza DB principale di un cluster DB sottorete pubblica sul VPC e se la proprietà "" dell'istanza è. PubliclyAccessible `true` Per trovare il valore di questa proprietà, puoi usare il comando. describe-db-instances AWS CLI In alternativa, puoi utilizzare la AWS Management Console per aprire la scheda Connectivity & security (Connettività e sicurezza) e controllare che l'opzione Publicly accessible (Accessibile pubblicamente) sia impostata su Yes (Sì). Per verificare se l'istanza si trova nella sottorete pubblica del VPC, puoi utilizzare la AWS Management Console o la AWS CLI.

  Per configurare l'accesso a Lambda, usi AWS Management Console o AWS CLI per creare una regola in uscita sul gruppo di sicurezza del tuo VPC. La regola in uscita specifica che TCP può utilizzare la porta 443 per inviare pacchetti a qualsiasi indirizzo IPv4 (0.0.0.0/0).
- Istanza privata del : in questo caso, la proprietà "" dell'istanza si trova o si trova in una sottorete privata. PubliclyAccessible `false` Per consentire all'istanza di funzionare con Lambda, è possibile utilizzare un gateway Network Address Translation (NAT). Per ulteriori informazioni, consulta Gateway NAT. In alternativa, puoi configurare il VPC con un endpoint VPC per Lamda. Per ulteriori informazioni, consultare Endpoint VPC nella Guida per l'utente di Amazon VPC. L'endpoint risponde alle chiamate effettuate dall'istanza database RDS per PostgreSQL alle funzioni Lambda. L'endpoint VPC utilizza la propria risoluzione DNS privata. RDS for PostgreSQL non può utilizzare l'endpoint VPC di Lambda fino a quando non modifichi il valore di `rds.custom_dns_resolution` dal valore predefinito di 0 (non abilitato) a 1. A tale scopo:
  - Crea un gruppo parametri del database personalizzato.
  - Cambia il valore del parametro `rds.custom_dns_resolution` da `0` (valore predefinito) a `1`.
  - Modifica l'istanza database per utilizzare il gruppo parametri del database personalizzato.

- Affinché il parametro modificato abbia effetto, riavvia l'istanza.

Il tuo VPC può ora interagire con il AWS Lambda VPC a livello di rete. Configura quindi le autorizzazioni utilizzando IAM.

Passaggio 2: configura IAM per l'istanza RDS del eAWS Lambda

Il richiamo di funzioni Lambda dall'istanza database RDS for PostgreSQL richiede determinati privilegi. Per configurare i privilegi necessari, si consiglia di creare una policy IAM che consenta di richiamare le funzioni Lambda, assegnare tale policy a un ruolo e quindi applicare il ruolo all'istanza database. Questo approccio fornisce all'istanza database privilegi per richiamare la funzione Lambda specificata per tuo conto. La procedura seguente mostra come eseguire questa operazione in AWS CLI.

Configurare le autorizzazioni IAM per l'utilizzo dell'istanza Amazon RDS con Lambda

1. Usa il AWS CLI comando create-policy per creare una policy IAM che consenta all'istanza database Aurora di richiamare la funzione Lambda specificata. (L'ID dichiarazione (Sid) è una descrizione facoltativa per la dichiarazione di policy e non ha alcun effetto sull'utilizzo). Questa policy fornisce all'istanza database le autorizzazioni minime necessarie per richiamare la funzione Lambda specificata.

   ```
   aws iam create-policy  --policy-name rds-lambda-policy --policy-document '{
       "Version": "2012-10-17",
       "Statement": [
           {
           "Sid": "AllowAccessToExampleFunction",
           "Effect": "Allow",
           "Action": "lambda:InvokeFunction",
           "Resource": "arn:aws:lambda:aws-region:444455556666:function:my-function"
           }
       ]
   }'
   ```

   In alternativa, puoi utilizzare la policy `AWSLambdaRole` predefinita che ti consente di richiamare una qualsiasi delle tue funzioni Lambda. Per ulteriori informazioni, consulta Policy IAM basate sull'identità per Lambda
2. Utilizza il comando create-role per creare un ruolo IAM che la policy può assumere in fase di esecuzione AWS CLI .

```
aws iam create-role  --role-name rds-lambda-role --assume-role-policy-document '{
    "Version": "2012-10-17",
    "Statement": [
        {
        "Effect": "Allow",
        "Principal": {
            "Service": "rds.amazonaws.com"
        },
        "Action": "sts:AssumeRole"
        }
    ]
}'
```

3. Applica la policy al ruolo utilizzando il attach-role-policy AWS CLI comando.

```
aws iam attach-role-policy \
    --policy-arn arn:aws:iam::444455556666:policy/rds-lambda-policy \
    --role-name rds-lambda-role --region aws-region
```

4. AWS CLI Quest'ultimo passaggio consente agli utenti dell'istanza database di richiamare le funzioni Lambda.

```
aws rds add-role-to-db-instance \
       --db-instance-identifier my-instance-name \
       --feature-name Lambda \
       --role-arn  arn:aws:iam::444455556666:role/rds-lambda-role   \
       --region aws-region
```

Con le configurazioni di VPC e IAM completate, puoi ora installare l'estensione `aws_lambda`. (Puoi installare l'estensione in qualsiasi momento, ma fino a quando non configuri il supporto VPC e i privilegi IAM corretti, l'estensione `aws_lambda`non aggiunge nulla alle funzionalità dell'istanza database RDS for PostgreSQL).

Fase 3: installazione dell'estensione **`aws_lambda`** per un'istanza database RDS for PostgreSQL

Questa estensione fornisce all'istanza database RDS for PostgreSQL la possibilità di chiamare le funzioni Lambda da PostgreSQL.

Installare l'estensione **aws_lambda**nell'istanza database RDS for PostgreSQL

Utilizza la riga di comando PostgreSQL `psql` o lo strumento pgAdmin per connetterti al all'istanza database RDS for PostgreSQL.

1. Connettiti all'istanza database RDS for PostgreSQL come utente con privilegi `rds_superuser`. L'utente `postgres` predefinito viene visualizzato nell'esempio.

   ```
   psql -h instance.444455556666.aws-region.rds.amazonaws.com -U postgres -p 5432
   ```

2. Installa l'estensione `aws_lambda`. Anche l'estensione `aws_commons` è obbligatoria. Fornisce funzioni di supporto ad `aws_lambda` e molte altre estensioni Aurora per PostgreSQL. Se non si trova già nella tua istanza database RDS for PostgreSQL, viene installata con `aws_lambda` come mostrato di seguito.

   ```
   CREATE EXTENSION IF NOT EXISTS aws_lambda CASCADE;
   NOTICE:  installing required extension "aws_commons"
   CREATE EXTENSION
   ```

L'estensione `aws_lambda` è installata nell'istanza database principale . Ora puoi creare strutture utili per richiamare le tue funzioni Lambda.

Fase 4: utilizzo delle funzioni di supporto Lambda con l'istanza database RDS for PostgreSQL (facoltativo)

Puoi utilizzare le funzioni di supporto nell'estensione `aws_commons` per preparare entità che è possibile richiamare più facilmente da PostgreSQL. Per farlo, devi disporre delle seguenti informazioni sulle funzioni Lambda:

- Nome funzione: il nome, l'Amazon Resource Name (ARN), la versione o l'alias della funzione Lambda. La policy IAM creata in Fase 2: configurazione di IAM per l'istanza e Lambda richiede l'ARN, quindi ti consigliamo di utilizzare l'ARN della tua funzione.
- AWS Regione: (Facoltativo) La AWS regione in cui si trova la funzione Lambda se non si trova nella stessa regione dell'istanza DB RDS per PostgreSQL del .

Per mantenere le informazioni sul nome della funzione Lambda, puoi utilizzare la funzione aws_commons.create_lambda_function_arn. Questa funzione di supporto crea una struttura

composita `aws_commons._lambda_function_arn_1` con i dettagli necessari alla funzione di richiamo. Di seguito, puoi trovare tre approcci alternativi per l'impostazione di questa struttura composita.

```
SELECT aws_commons.create_lambda_function_arn(
   'my-function',
   'aws-region'
) AS aws_lambda_arn_1 \gset
```

```
SELECT aws_commons.create_lambda_function_arn(
   '111122223333:function:my-function',
   'aws-region'
) AS lambda_partial_arn_1 \gset
```

```
SELECT aws_commons.create_lambda_function_arn(
   'arn:aws:lambda:aws-region:111122223333:function:my-function'
) AS lambda_arn_1 \gset
```

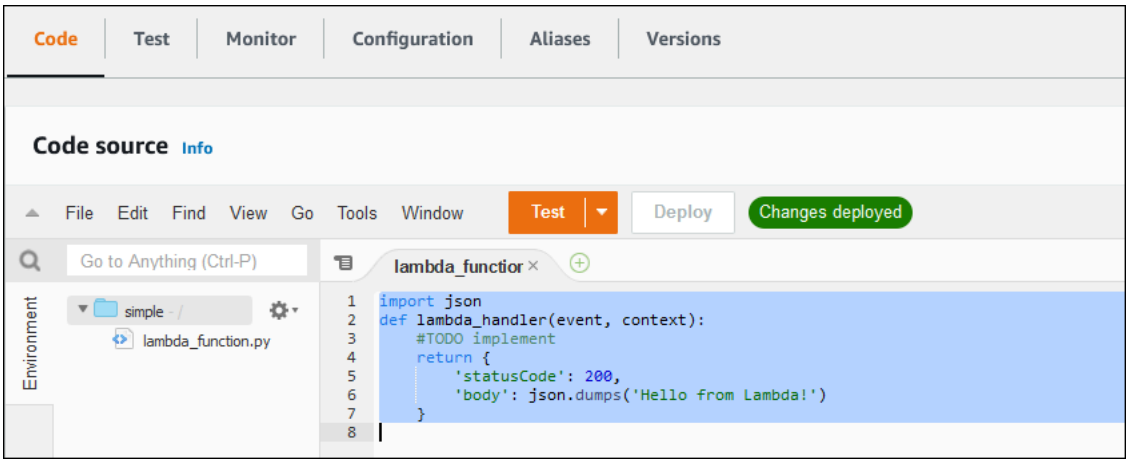
Ognuno di questi valori può essere utilizzato nelle chiamate alla funzione aws_lambda.invoke. Per alcuni esempi, consulta Fase 5: richiamo di una funzione Lambda dall'istanza database RDS for PostgreSQL.

Fase 5: richiamo di una funzione Lambda dall'istanza database RDS for PostgreSQL

La funzione `aws_lambda.invoke` si comporta in modo sincrono o asincrono, a seconda del `invocation_type`. Le due alternative per questo parametro sono `RequestResponse` (il valore predefinito) e `Event`, come di seguito riportato.

- **`RequestResponse`**: questo tipo di richiamo è sincrono. Questo è il comportamento predefinito quando viene effettuata la chiamata senza specificare un tipo di chiamata. Il payload di risposta include i risultati della funzione `aws_lambda.invoke`. Utilizza questo tipo di chiamata quando il flusso di lavoro richiede la ricezione dei risultati della funzione Lambda prima di procedere.
- **`Event`**: questo tipo di richiamo è asincrono. La risposta non include un payload contenente i risultati. Utilizza questo tipo di richiamo quando il flusso di lavoro non ha bisogno di un risultato della funzione Lambda per continuare l'elaborazione.

Come semplice test della configurazione, puoi connetterti all'istanza database utilizzando `psql` e richiamare una funzione di esempio dalla riga di comando. Supponiamo di avere una delle funzioni di base impostate sul tuo servizio Lambda, come la semplice funzione Python mostrata nello screenshot di seguito.



Per richiamare una funzione di esempio

1. Connettiti all'istanza database utilizzando `psql` o pgAdmin.

   ```
   psql -h instance.444455556666.aws-region.rds.amazonaws.com -U postgres -p 5432
   ```

2. Richiama la funzione utilizzando il relativo ARN.

   ```
   SELECT * from
    aws_lambda.invoke(aws_commons.create_lambda_function_arn('arn:aws:lambda:aws-
   region:444455556666:function:simple', 'us-west-1'), '{"body": "Hello from
    Postgres!"}'::json );
   ```

   La risposta è la seguente.

   ```
   status_code |                         payload                          |
    executed_version | log_result
   -------------+----------------------------------------------------------
   +------------------+------------
            200 | {"statusCode": 200, "body": "\"Hello from Lambda!\""} | $LATEST
        |
   (1 row)
   ```

Se il tuo tentativo di richiamo non ha esito positivo, vedi Messaggi di errore della funzione Lambda .

Fase 6: concessione delle autorizzazioni ad altri utenti per richiamare le funzioni Lambda

A questo punto della procedura, solo tu in qualità di `rds_superuser` puoi richiamare le funzioni Lambda. Per consentire ad altri utenti di richiamare le funzioni che crei, è necessario concedere loro l'autorizzazione.

Per concedere ad altri utenti l'autorizzazione per richiamare le funzioni Lambda

1. Connettiti all'istanza database utilizzando `psql` o pgAdmin.

```
psql -h instance.444455556666.aws-region.rds.amazonaws.com -U postgres -p 5432
```

2. Esegui i seguenti comandi SQL:

```
postgres=>  GRANT USAGE ON SCHEMA aws_lambda TO db_username;
GRANT EXECUTE ON ALL FUNCTIONS IN SCHEMA aws_lambda TO db_username;
```

Esempi: Richiamo delle funzioni Lambda dall'istanza database RDS for PostgreSQL

Di seguito, è possibile trovare diversi esempi di chiamate alla funzione aws_lambda.invoke. Nella maggior parte degli esempi viene utilizzata la struttura composita `aws_lambda_arn_1` che crei in Fase 4: utilizzo delle funzioni di supporto Lambda con l'istanza database RDS for PostgreSQL (facoltativo) per semplificare il passaggio dei dettagli della funzione. Per un esempio di chiamata asincrona, consulta Esempio: richiamo (di eventi) asincroni di funzioni Lambda. Tutti gli altri esempi elencati utilizzano il richiamo sincrono.

Per ulteriori informazioni sui tipi di chiamata Lambda, consulta Richiamo di funzioni Lambda nella Guida per gli sviluppatori di AWS Lambda . Per ulteriori informazioni su `aws_lambda_arn_1`, consulta aws_commons.create_lambda_function_arn.

Elenco di esempi

- Esempio: invocazione sincrona (RequestResponse) di funzioni Lambda
- Esempio: richiamo (di eventi) asincroni di funzioni Lambda
- Esempio: acquisizione del registro di esecuzione Lambda in una risposta di funzione
- Esempio: inclusione del contesto client in una funzione Lambda

- Esempio: richiamo di una versione specifica di una funzione Lambda

Esempio: invocazione sincrona (RequestResponse) di funzioni Lambda

Di seguito sono riportati due esempi di una chiamata di funzione Lambda sincrona. I risultati di queste chiamate di funzione `aws_lambda.invoke` sono gli stessi.

```
SELECT * FROM aws_lambda.invoke('aws_lambda_arn_1', '{"body": "Hello from
 Postgres!"}'::json);
```

```
SELECT * FROM aws_lambda.invoke('aws_lambda_arn_1', '{"body": "Hello from
 Postgres!"}'::json, 'RequestResponse');
```

I parametri sono descritti come segue:

- `:'aws_lambda_arn_1'`: questo parametro identifica la struttura composita creata in Fase 4: utilizzo delle funzioni di supporto Lambda con l'istanza database RDS for PostgreSQL (facoltativo) con la funzione di supporto di `aws_commons.create_lambda_function_arn`. Puoi anche creare questa struttura in linea all'interno della chiamata `aws_lambda.invoke` come segue:

  ```
  SELECT * FROM aws_lambda.invoke(aws_commons.create_lambda_function_arn('my-function',
   'aws-region'),
  '{"body": "Hello from Postgres!"}'::json
  );
  ```

- `'{"body": "Hello from PostgreSQL!"}'::json`: il payload JSON da passare alla funzione Lambda.
- `'RequestResponse'`: il tipo di richiamo Lambda.

Esempio: richiamo (di eventi) asincroni di funzioni Lambda

Di seguito è riportato un esempio di una chiamata di funzione Lambda asincrona. Il tipo di richiamo `Event` pianifica il richiamo della funzione Lambda con il payload di input specificato e restituisce immediatamente un risultato. Utilizza il tipo di chiamata `Event` in determinati flussi di lavoro che non dipendono dai risultati della funzione Lambda.

```
SELECT * FROM aws_lambda.invoke('aws_lambda_arn_1', '{"body": "Hello from
 Postgres!"}'::json, 'Event');
```

Esempio: acquisizione del registro di esecuzione Lambda in una risposta di funzione

È possibile includere gli ultimi 4 KB del registro di esecuzione nella risposta della funzione utilizzando il parametro `log_type` nella chiamata di funzione `aws_lambda.invoke`. Per impostazione predefinita, questo parametro è impostato su `None`, ma puoi specificare `Tail` per acquisire i risultati del registro di esecuzione Lambda nella risposta, come illustrato di seguito.

```
SELECT *, select convert_from(decode(log_result, 'base64'), 'utf-8') as log FROM
 aws_lambda.invoke(:'aws_lambda_arn_1', '{"body": "Hello from Postgres!"}'::json,
 'RequestResponse', 'Tail');
```

Impostare il parametro aws_lambda.invoke della funzione `log_type` su `Tail` per includere il log di esecuzione nella risposta. Il valore predefinito per il parametro `log_type` è `None`.

Il `log_result` che viene restituito è una stringa `base64` codificata. È possibile decodificare i contenuti utilizzando una combinazione delle funzioni PostgreSQL `decode` e `convert_from`.

Per ulteriori informazioni su `log_type`, consulta aws_lambda.invoke.

Esempio: inclusione del contesto client in una funzione Lambda

La funzione `aws_lambda.invoke` ha un parametro `context` che puoi utilizzare per passare le informazioni separate dal payload, come illustrato di seguito.

```
SELECT *, convert_from(decode(log_result, 'base64'), 'utf-8') as log FROM
 aws_lambda.invoke(:'aws_lambda_arn_1', '{"body": "Hello from Postgres!"}'::json,
 'RequestResponse', 'Tail');
```

Per includere il contesto client, utilizzare un oggetto JSON per il parametro aws_lambda.invoke della funzione `context`.

Per ulteriori informazioni sul parametro `context`, consulta la documentazione di riferimento di aws_lambda.invoke.

Esempio: richiamo di una versione specifica di una funzione Lambda

Puoi specificare una determinata versione di una funzione Lambda includendo il parametro `qualifier` con la chiamata `aws_lambda.invoke`. Di seguito puoi trovare un esempio in cui viene utilizzato '*`custom_version`*' come alias per la versione.

```
SELECT * FROM aws_lambda.invoke('aws_lambda_arn_1', '{"body": "Hello from
 Postgres!"}'::json, 'RequestResponse', 'None', NULL, 'custom_version');
```

Puoi inoltre fornire un qualificatore di funzione Lambda con i dettagli relativi al nome della funzione, come mostrato di seguito.

```
SELECT * FROM aws_lambda.invoke(aws_commons.create_lambda_function_arn('my-
function:custom_version', 'us-west-2'),
'{"body": "Hello from Postgres!"}'::json);
```

Per ulteriori informazioni su `qualifier` e altri parametri, consulta documentazione di riferimento di aws_lambda.invoke.

Messaggi di errore della funzione Lambda

Nell'elenco seguente sono disponibili informazioni sui messaggi di errore, con le possibili cause e soluzioni.

- Problemi di configurazione del VPC

  I problemi di configurazione del VPC possono generare i seguenti messaggi di errore al momento della connessione:

  ```
  ERROR:  invoke API failed
  DETAIL: AWS Lambda client returned 'Unable to connect to endpoint'.
  CONTEXT:  SQL function "invoke" statement 1
  ```

  Una causa comune di questo errore è il gruppo di sicurezza VPC configurato in modo errato. Assicurati di avere una regola in uscita per TCP aperta sulla porta 443 del gruppo di sicurezza VPC in modo che il VPC possa connettersi al VPC Lambda.

  Se la tua istanza database è privata, controlla la configurazione DNS privata per il tuo VPC. Assicurati di impostare il `rds.custom_dns_resolution` parametro su 1 e di AWS PrivateLink configurarlo come descritto in. Fase 1: configurare l'istanza DB RDS del per le connessioni in uscita a AWS Lambda Per ulteriori informazioni, consulta Interface VPC endpoints ().AWS PrivateLink

- Mancanza delle autorizzazioni necessarie per richiamare le funzioni Lambda

Se viene visualizzato uno dei seguenti messaggi di errore, l'utente (ruolo) che richiama la funzione non dispone delle autorizzazioni appropriate.

```
ERROR:  permission denied for schema aws_lambda
```

```
ERROR:  permission denied for function invoke
```

A un utente (ruolo) devono essere concesse autorizzazioni specifiche per richiamare le funzioni Lambda. Per ulteriori informazioni, consulta Fase 6: concessione delle autorizzazioni ad altri utenti per richiamare le funzioni Lambda.

- Gestione impropria degli errori nelle funzioni Lambda

  Se una funzione Lambda genera un'eccezione durante l'elaborazione della richiesta, `aws_lambda.invoke` non riesce e indica un errore PostgreSQL come quello seguente.

```
SELECT * FROM aws_lambda.invoke('aws_lambda_arn_1', '{"body": "Hello from
 Postgres!"}'::json);
ERROR:  lambda invocation failed
DETAIL:  "arn:aws:lambda:us-west-2:555555555555:function:my-function" returned error
 "Unhandled", details: "<Error details string>".
```

  Assicurati di gestire gli errori nelle funzioni Lambda o nell'applicazione PostgreSQL.

AWS Lambdariferimento a funzioni e parametri

Di seguito è riportato il riferimento per le funzioni e i parametri da utilizzare per richiamare Lambda con PostgreSQL RDS per PostgreSQL.

Funzioni e parametri

- aws_lambda.invoke
- aws_commons.create_lambda_function_arn
- parametri aws_lambda

aws_lambda.invoke

Esegue una funzione Lambda per una istanza database RDS for PostgreSQL.

Per ulteriori dettagli sul richiamo delle funzioni Lambda, consulta anche Invoke nella Guida per gli sviluppatori di AWS Lambda.

Sintassi

JSON

```
aws_lambda.invoke(
IN function_name TEXT,
IN payload JSON,
IN region TEXT DEFAULT NULL,
IN invocation_type TEXT DEFAULT 'RequestResponse',
IN log_type TEXT DEFAULT 'None',
IN context JSON DEFAULT NULL,
IN qualifier VARCHAR(128) DEFAULT NULL,
OUT status_code INT,
OUT payload JSON,
OUT executed_version TEXT,
OUT log_result TEXT)
```

```
aws_lambda.invoke(
IN function_name aws_commons._lambda_function_arn_1,
IN payload JSON,
IN invocation_type TEXT DEFAULT 'RequestResponse',
IN log_type TEXT DEFAULT 'None',
IN context JSON DEFAULT NULL,
IN qualifier VARCHAR(128) DEFAULT NULL,
OUT status_code INT,
OUT payload JSON,
OUT executed_version TEXT,
OUT log_result TEXT)
```

JSONB

```
aws_lambda.invoke(
IN function_name TEXT,
IN payload JSONB,
IN region TEXT DEFAULT NULL,
IN invocation_type TEXT DEFAULT 'RequestResponse',
IN log_type TEXT DEFAULT 'None',
IN context JSONB DEFAULT NULL,
IN qualifier VARCHAR(128) DEFAULT NULL,
```

```
OUT status_code INT,
OUT payload JSONB,
OUT executed_version TEXT,
OUT log_result TEXT)
```

```
aws_lambda.invoke(
IN function_name aws_commons._lambda_function_arn_1,
IN payload JSONB,
IN invocation_type TEXT DEFAULT 'RequestResponse',
IN log_type TEXT DEFAULT 'None',
IN context JSONB DEFAULT NULL,
IN qualifier VARCHAR(128) DEFAULT NULL,
OUT status_code INT,
OUT payload JSONB,
OUT executed_version TEXT,
OUT log_result TEXT
)
```

Parametri di input

function_name

Nome identificativo della funzione Lambda. Il valore può essere il nome della funzione, un ARN o un ARN parziale. Per un elenco dei formati possibili, consulta Formati dei nomi delle funzioni Lambda nella Guida per gli sviluppatori di AWS Lambda.

payload

L'input per la funzione Lambda. Il formato può essere JSON o JSONB. Per ulteriori informazioni, consulta Tipi di JSON nella documentazione di PostgreSQL.

region

(Facoltativo) La regione Lambda per la funzione. Per impostazione predefinita, RDS risolve la regione AWS dall'ARN completo nella `function_name` oppure utilizza la regione dell'istanza database RDS for PostgreSQL. Se il valore di questa Regione è in conflitto con quello fornito nell'ARN `function_name`, viene generato un errore.

invocation_type

Il tipo di chiamata della funzione Lambda. Il valore prevede la distinzione tra lettere maiuscole e minuscole. I valori possibili sono:

- `RequestResponse` – Il valore predefinito. Questo tipo di chiamata per una funzione Lambda è sincrono e restituisce un payload di risposta nel risultato. Utilizzare il tipo di chiamata `RequestResponse` quando il flusso di lavoro dipende dalla ricezione immediata del risultato della funzione Lambda.
- `Event` – Questo tipo di chiamata per una funzione Lambda è asincrono e restituisce immediatamente una risposta senza un payload restituito. Utilizzare il tipo di chiamata `Event` quando non sono necessari i risultati della funzione Lambda prima che il flusso di lavoro proceda.
- `DryRun` – Questo tipo di chiamata verifica l'accesso senza eseguire la funzione Lambda.

log_type

Il tipo di log Lambda da restituire nel parametro di output `log_result`. Il valore prevede la distinzione tra lettere maiuscole e minuscole. I valori possibili sono:

- Tail – Il parametro di output `log_result` restituito includerà gli ultimi 4 KB del registro di esecuzione.
- None – Non viene restituita nessuna informazione di log Lambda.

context

Contesto client in formato JSON o JSONB. I campi da utilizzare includono `custom` e `env`.

qualifier

Un qualificatore che identifica la versione di una funzione Lambda da richiamare. Se questo valore è in conflitto con quello fornito nell' `function_name` ARN, viene generato un errore.

Parametri di output

status_code

Un codice di risposta allo stato HTTP. Per ulteriori informazioni, consulta Elementi di risposta del richiamo di Lambda nella Guida per gli sviluppatori di AWS Lambda.

payload

Le informazioni restituite dalla funzione Lambda eseguita. Il formato è in JSON o JSONB.

executed_version

La versione della funzione Lambda eseguita.

log_result

Le informazioni del registro di esecuzione restituite se il valore `log_type` è `Tail` quando è stata richiamata la funzione Lambda. Il risultato contiene gli ultimi 4 KB del registro di esecuzione codificato in Base64.

aws_commons.create_lambda_function_arn

Crea una struttura `aws_commons._lambda_function_arn_1` per contenere le informazioni sul nome della funzione Lambda. È possibile utilizzare i risultati della `aws_commons.create_lambda_function_arn` funzione nel parametro `function_name` della funzione aws_lambda.invoke.

Sintassi

```
aws_commons.create_lambda_function_arn(
    function_name TEXT,
    region TEXT DEFAULT NULL
    )
    RETURNS aws_commons._lambda_function_arn_1
```

Parametri di input

function_name

Stringa di testo obbligatoria contenente il nome della funzione Lambda. Il valore può essere un nome di funzione, un ARN parziale o un ARN completo.

region

Una stringa di testo facoltativa contenente la regione AWS in cui si trova la funzione Lambda. Per un elenco di nomi di regione e dei valori associati, consulta Regioni, zone di disponibilità e Local Zones.

parametri aws_lambda

In questa tabella, puoi trovare i parametri associati alla funzione. `aws_lambda`

Parametro	Descrizione
`aws_lambda.connect_timeout_ms`	Si tratta di un parametro dinamico che imposta il tempo di attesa massimo durante la connessione a AWS Lambda. I valori predefiniti sono`1000`. I valori consentiti per questo parametro sono compresi tra 1 e 900000.
`aws_lambda.request_timeout_ms`	Si tratta di un parametro dinamico che imposta il tempo di attesa massimo in attesa della risposta da AWS Lambda. I valori predefiniti sono`3000`. I valori consentiti per questo parametro sono compresi tra 1 e 900000.
`aws_lambda.endpoint_override`	Speciifica l'endpoint che può essere utilizzato per connettersi a AWS Lambda. Una stringa vuota seleziona l'endpoint AWS Lambda predefinito per la regione. È necessario riavviare il database affinché questa modifica statica dei parametri abbia effetto.

Attività DBA comuni per Amazon RDS for PostgreSQL

Gli amministratori di database (DBA) eseguono una serie di attività durante l'amministrazione di un'istanza database Amazon RDS for PostgreSQL. Se sei un amministratore di database (DBA) che già conosce PostgreSQL, devi essere a conoscenza di alcune delle principali differenze tra l'esecuzione di PostgreSQL sul tuo hardware e RDS per PostgreSQL. Ad esempio, poiché si tratta di un servizio gestito, Amazon RDS non consente alla shell di accedere alle istanze database. Ciò significa che non puoi accedere direttamente a `pg_hba.conf` e altri file di configurazione. Per RDS per PostgreSQL, le modifiche che vengono in genere apportate al file di configurazione PostgreSQL di un'istanza on-premise vengono applicate a un gruppo di parametri database personalizzato associato all'istanza database RDS per PostgreSQL. Per ulteriori informazioni, consulta Utilizzo di gruppi di parametri.

Non è possibile accedere ai file di log come con un'istanza PostgreSQL on-premise. Per ulteriori informazioni sulla registrazione, consulta File di log del database RDS per PostgreSQL.

Come altro esempio, non è possibile accedere all'account PostgreSQL `superuser`. Su RDS per PostgreSQL, il ruolo `rds_superuser` è quello più privilegiato e viene concesso a `postgres` al momento della configurazione. Sia che tu abbia familiarità con l'utilizzo di PostgreSQL on-premise o che tu sia un nuovo utente di RDS per PostgreSQL, ti consigliamo di familiarizzare con il ruolo `rds_superuser` e approfondire l'utilizzo di ruoli, utenti, gruppi e autorizzazioni. Per ulteriori informazioni, consulta Informazioni su ruoli e autorizzazioni di PostgreSQL.

Di seguito sono riportate alcune attività DBA comuni per RDS for PostgreSQL.

Argomenti

- Regole di confronto supportate in RDS per PostgreSQL
- Informazioni su ruoli e autorizzazioni di PostgreSQL
- Utilizzo della funzione di autovacuum di PostgreSQL in Amazon RDS for PostgreSQL
- Utilizzo dei meccanismi di registrazione supportati da RDS for PostgreSQL
- Gestione dei file temporanei con PostgreSQL
- Utilizzo di pgBadger per l'analisi del registro con PostgreSQL
- Utilizzo di PGSnapper per il monitoraggio di PostgreSQL
- Utilizzo dei parametri sull'istanza database RDS for PostgreSQL

Regole di confronto supportate in RDS per PostgreSQL

Le regole di confronto sono un insieme di regole che determinano il modo in cui le stringhe di caratteri archiviate nel database vengono ordinate e confrontate. Le regole di confronto svolgono un ruolo fondamentale nel sistema del computer e sono incluse come parte del sistema operativo. Le regole di confronto cambiano nel tempo quando vengono aggiunti nuovi caratteri alle lingue o quando vengono modificate le regole di ordinamento.

Le librerie di regole di confronto definiscono regole e algoritmi specifici per una regola di confronto. Le librerie di regole di confronto più popolari utilizzate in PostgreSQL sono GNU C (glibc) e Internationalization components for Unicode (ICU). Per impostazione predefinita, RDS per PostgreSQL utilizza la regola di confronto glibc che include le sequenze di ordinamento dei caratteri unicode per sequenze di caratteri multibyte.

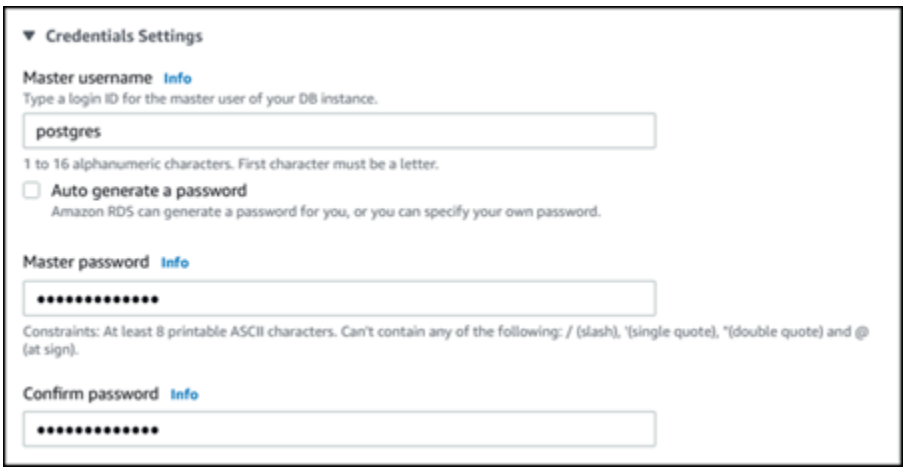
Quando si crea una nuova istanza database in RDS per PostgreSQL, viene cercata la regola di confronto disponibile nel sistema operativo. I parametri PostgreSQL `LC_COLLATE` e `LC_CTYPE` del comando `CREATE DATABASE` vengono utilizzati per specificare una regola di confronto, che rappresenta la regola di confronto predefinita nel database. In alternativa, puoi anche usare il parametro `LOCALE` in `CREATE DATABASE` per impostare questi parametri e determinare la regole di confronto predefinita per le stringhe di caratteri nel database e le regole per classificare i caratteri come lettere, numeri o simboli. Puoi anche scegliere una regola di confronto da utilizzare per una colonna, un indice o una query.

RDS per PostgreSQL dipende dalla libreria glibc del sistema operativo per il supporto della regola di confronto. L'istanza RDS per PostgreSQL viene aggiornata periodicamente con le versioni più recenti del sistema operativo. Questi aggiornamenti a volte includono una nuova versione della libreria glibc. Raramente, le versioni più recenti della libreria glibc modificano l'ordinamento o la regola di confronto di alcuni caratteri e pertanto i dati possono essere ordinati in modo diverso o produrre voci di indice non valide. Se si riscontrano problemi di ordinamento per la regola di confronto durante un aggiornamento, potrebbe essere necessario ricostruire gli indici.

Per ridurre il possibile impatto degli aggiornamenti della libreria glibc, RDS per PostgreSQL ora include una libreria di regole di confronto predefinita indipendente. Questa libreria di regole confronto è disponibile in RDS for PostgreSQL 14.6, 13.9, 12.13, 11.18, 10.23 e versioni secondarie successive. È compatibile con glibc 2.26-59.amzn2 e fornisce stabilità dell'ordinamento per evitare risultati errati delle query.

Informazioni su ruoli e autorizzazioni di PostgreSQL

Quando si crea un' per PostgreSQL utilizzando, viene creato contemporaneamente un account amministratore. AWS Management Console Per impostazione predefinita, verrà chiamato `postgres`, come mostrato nello screenshot seguente:



Anziché accettare il valore predefinito (`postgres`) è possibile scegliere un nome diverso. In tal caso, il nome scelto deve iniziare con una lettera e contenere da 1 a 16 caratteri alfanumerici. Per semplicità, facciamo riferimento a questo account utente principale utilizzando il suo valore predefinito (`postgres`) in tutta la Guida.

Se si utilizza il `create-db-instance` AWS CLI anziché il AWS Management Console, si crea il nome passandolo con il parametro nel comando. `master-username` Per ulteriori informazioni, consulta Creazione di un'istanza database Amazon RDS.

Sia che utilizzi l' AWS Management ConsoleAPI Amazon RDS AWS CLI, che utilizzi il `postgres` nome predefinito o scelga un nome diverso, questo primo account utente del database è membro del `rds_superuser` gruppo e dispone di `rds_superuser` privilegi.

Argomenti

- Comprendere il ruolo rds_superuser
- Controllo dell'accesso utente al database PostgreSQL
- Delega e controllo della gestione delle password utente
- Utilizzo delle crittografia password SCRAM per PostgreSQL

Comprendere il ruolo rds_superuser

In PostgreSQL, un ruolo può definire un utente, un gruppo o un insieme di autorizzazioni specifiche concesse a un gruppo o a un utente per vari oggetti nel database. I comandi PostgreSQL `CREATE USER` e `CREATE GROUP` sono stati sostituiti dal comando `CREATE ROLE` più generico, ma con proprietà specifiche per distinguere gli utenti del database. Un utente del database può essere paragonato a un ruolo con il privilegio LOGIN.

> **Note**
>
> È comunque possibile continuare a utilizzare i comandi `CREATE USER` e `CREATE GROUP`. Per ulteriori informazioni, consulta la sezione relativa ai ruoli di database nella documentazione di PostgreSQL.

L'utente `postgres` è l'utente di database più privilegiato nell'istanza database di RDS per PostgreSQL. Ha le caratteristiche definite dalla seguente istruzione `CREATE ROLE`.

```
CREATE ROLE postgres WITH LOGIN NOSUPERUSER INHERIT CREATEDB CREATEROLE NOREPLICATION
 VALID UNTIL 'infinity'
```

Le proprietà `NOSUPERUSER`, `NOREPLICATION`, `INHERIT` e `VALID UNTIL 'infinity'` sono le opzioni predefinite per CREATE ROLE, se non diversamente specificato.

Per impostazione predefinita, `postgres` dispone dei privilegi concessi al `rds_superuser` ruolo e delle autorizzazioni per creare ruoli e database. Il ruolo `rds_superuser` consente all'utente `postgres` di eseguire le seguenti operazioni:

- Aggiungere le estensioni che sono disponibili per l'uso con Amazon RDS. Per ulteriori informazioni, consulta Utilizzo delle caratteristiche di PostgreSQL supportate da Amazon RDS for PostgreSQL
- Creare ruoli per gli utenti e concedere i relativi privilegi. Per ulteriori informazioni, consulta CREATE ROLE e GRANTnella documentazione di PostgreSQL.
- Creare database. Per ulteriori informazioni, consulta CREATE DATABASE nella documentazione di PostgreSQL.
- Concedere privilegi `rds_superuser` a ruoli utente che non dispongono di questi privilegi e revocare i privilegi, se necessario. Si consiglia di concedere questo ruolo solo agli utenti che eseguono attività superuser. In altre parole, è possibile concedere questo ruolo agli amministratori di database (DBA) o agli amministratori di sistema.

- Concedere (e revocare) il ruolo `rds_replication` per gli utenti del database che non hanno il ruolo `rds_superuser`.
- Concedere (e revocare) il ruolo `rds_password` per gli utenti del database che non hanno il ruolo `rds_superuser`.
- Ottenere informazioni sullo stato di tutte le connessioni al database utilizzando la vista `pg_stat_activity`. Quando necessario, il ruolo `rds_superuser` può arrestare qualsiasi connessione utilizzando il comando `pg_terminate_backend` o `pg_cancel_backend`.

Nell'istruzione `CREATE ROLE postgres...`, si può vedere che il ruolo utente `postgres` non concede specificamente autorizzazioni PostgreSQL `superuser`. RDS per PostgreSQL è un servizio gestito e pertanto non è possibile accedere al sistema operativo host, né connettersi utilizzando l'account PostgreSQL`superuser`. Molte delle attività che richiedono l'accesso di tipo `superuser` su un PostgreSQL autonomo viene gestito automaticamente da Amazon RDS.

Per ulteriori informazioni sulla concessione dei privilegi, consulta la sezione relativa al comando GRANT nella documentazione di PostgreSQL.

Il ruolo `rds_superuser` è uno dei diversi ruoli predefinito in un Istanza database di RDS per PostgreSQL.

> **Note**
>
> In PostgreSQL 13 e versioni precedenti, i ruoli di default sono conosciuti come ruoli predefiniti.

L'elenco seguente fornisce alcuni degli altri ruoli predefiniti creati automaticamente per un nuovo . Istanza database di RDS per PostgreSQL. I ruoli predefiniti e i relativi privilegi non possono essere modificati. Non è possibile eliminare, rinominare o modificare i privilegi per questi ruoli predefiniti. Qualsiasi tentativo comporta la generazione di un errore.

- rds_password - Un ruolo in grado di modificare le password e configurare vincoli di password per gli utenti del database. Il `rds_superuser` ruolo viene concesso con questo ruolo per impostazione predefinita e può concedere il ruolo agli utenti del database. Per ulteriori informazioni, consulta Controllo dell'accesso utente al database PostgreSQL.
  - Per le versioni di RDS per PostgreSQL precedenti alla 14`rds_password`, role può modificare le password e impostare vincoli di password per gli utenti del database e gli utenti con ruolo.

`rds_superuser` A partire dalla versione 14 di RDS per PostgreSQL`rds_password`, role può modificare le password e impostare vincoli di password solo per gli utenti del database. Solo gli utenti con `rds_superuser` ruolo possono eseguire queste azioni su altri utenti con ruolo. `rds_superuser`

- rdsadmin – Un ruolo creato per gestire molte delle attività di gestione che l'amministratore con privilegi `superuser` esegue su un database PostgreSQL autonomo. Questo ruolo viene utilizzato internamente da RDS per PostgreSQL per molte attività di gestione.
- rdstopmgr: un ruolo utilizzato internamente da Amazon RDS per supportare le implementazioni multi-AZ.

Per visualizzare tutti i ruoli predefiniti, è possibile connettersi all'istanza database RDS per PostgreSQL e usare il metacomando `psql \du`. L'output è simile al seguente.

```
List of roles
  Role name   |      Attributes                    |      Member of
--------------+------------------------------------+-----------------------------------
postgres      | Create role, Create DB            +| {rds_superuser}
              | Password valid until infinity      |
rds_superuser | Cannot login                       | {pg_monitor,pg_signal_backend,
              |                                   +|   rds_replication,rds_password}
...
```

Nell'output, si vede che `rds_superuser` non è un ruolo utente del database (non può effettuare il login), ma ha i privilegi di molti altri ruoli. È inoltre possibile vedere che l'utente di database `postgres` è membro del ruolo `rds_superuser`. Come accennato in precedenza, `postgres` è il valore predefinito nella pagina Crea database della console Amazon RDS. Se si sceglie un altro nome, tale nome viene visualizzato nell'elenco dei ruoli.

Controllo dell'accesso utente al database PostgreSQL

I nuovi database in PostgreSQL vengono sempre creati con un set predefinito di privilegi nel schema `public` del database, che consente a tutti gli utenti e i ruoli del database di creare oggetti. I privilegi predefiniti permettono agli utenti del database di connettersi al database e di creare tabelle temporanee durante la connessione.

Per controllare meglio l'accesso degli utenti alle istanze database create sull'istanza database RDS per PostgreSQL, si consiglia di revocare questi privilegi `public` predefiniti. Dopo averlo fatto,

è consigliabile concedere privilegi specifici agli utenti del database su base più granulare, come mostrato nella procedura seguente.

Per impostare ruoli e privilegi per una nuova istanza database

Si supponga di aver configurando un database in un'istanza database RDS per PostgreSQL per poter essere usato da diversi ricercatori, che dovranno avere l'accesso in lettura-scrittura al database.

1. Utilizzare `psql` (o pgAdmin) per connettersi all'istanza database RDS per PostgreSQL:

   ```
   psql --host=your-db-instance.666666666666.aws-region.rds.amazonaws.com --port=5432
    --username=postgres --password
   ```

   Specifica la password, quando richiesto. Il client `psql` si connette e visualizza il database di connessione amministrativa predefinito `postgres=>` come prompt.
2. Per impedire agli utenti del database di creare oggetti nello schema `public`, eseguire le seguenti operazioni:

   ```
   postgres=> REVOKE CREATE ON SCHEMA public FROM PUBLIC;
   REVOKE
   ```
3. Creare quindi una nuova istanza database:

   ```
   postgres=> CREATE DATABASE lab_db;
   CREATE DATABASE
   ```
4. Revocare tutti i privilegi dallo schema `PUBLIC` in questo nuovo database.

   ```
   postgres=> REVOKE ALL ON DATABASE lab_db FROM public;
   REVOKE
   ```
5. Creare un ruolo per gli utenti del database.

   ```
   postgres=> CREATE ROLE lab_tech;
   CREATE ROLE
   ```
6. Concedere agli utenti del database con questo ruolo la possibilità di connettersi al database.

   ```
   postgres=> GRANT CONNECT ON DATABASE lab_db TO lab_tech;
   GRANT
   ```

7. Concedere a tutti gli utenti con il ruolo `lab_tech` tutti i privilegi per questo database.

```
postgres=> GRANT ALL PRIVILEGES ON DATABASE lab_db TO lab_tech;
GRANT
```

8. Creare utenti del database, come segue:

```
postgres=> CREATE ROLE lab_user1 LOGIN PASSWORD 'change_me';
CREATE ROLE
postgres=> CREATE ROLE lab_user2 LOGIN PASSWORD 'change_me';
CREATE ROLE
```

9. Concedere a questi due utenti i privilegi associati al ruolo lab_tech:

```
postgres=> GRANT lab_tech TO lab_user1;
GRANT ROLE
postgres=> GRANT lab_tech TO lab_user2;
GRANT ROLE
```

A questo punto, `lab_user1` e `lab_user2` possono connettersi al database `lab_db`. Questo esempio non segue le best practice per l'utilizzo aziendale, che potrebbero includere la creazione di più istanze database, schemi diversi e la concessione di autorizzazioni limitate. Per informazioni più complete e scenari aggiuntivi, consulta Gestione di utenti e ruoli PostgreSQL.

Per ulteriori informazioni sui privilegi in database PostgreSQL, consulta la sezione relativa al comando GRANT nella documentazione di PostgreSQL.

Delega e controllo della gestione delle password utente

Un amministratore di database (DBA) potrebbe voler delegare la gestione delle password utente. In alternativa, è possibile impedire agli utenti del database di modificare le password o di riconfigurare i vincoli delle password, ad esempio la durata della password. Per garantire che solo gli utenti del database scelti possano modificare le impostazioni della password, è possibile attivare la funzione di gestione delle password con restrizioni. Quando si attiva questa funzione, solo gli utenti del database a cui è stato concesso il ruolo `rds_password` saranno in grado di gestire le password.

> **Note**
>
> Per utilizzare la gestione delle password limitate, l'istanza database RDS per PostgreSQL deve eseguire PostgreSQL 10.6 o superiore.

Per impostazione predefinita, questa funzione è impostata su `off`, come mostrato di seguito:

```
postgres=> SHOW rds.restrict_password_commands;
  rds.restrict_password_commands
--------------------------------
 off
(1 row)
```

Per attivare questa funzione, utilizzare un gruppo di parametri personalizzato e modificare l'impostazione per `rds.restrict_password_commands` su 1. Assicurarsi di riavviare l'istanza database RDS per PostgreSQL per implementare l'impostazione.

Con questa funzione attiva, i privilegi `rds_password` sono obbligatori per i seguenti comandi SQL:

```
CREATE ROLE myrole WITH PASSWORD 'mypassword';
CREATE ROLE myrole WITH PASSWORD 'mypassword' VALID UNTIL '2023-01-01';
ALTER ROLE myrole WITH PASSWORD 'mypassword' VALID UNTIL '2023-01-01';
ALTER ROLE myrole WITH PASSWORD 'mypassword';
ALTER ROLE myrole VALID UNTIL '2023-01-01';
ALTER ROLE myrole RENAME TO myrole2;
```

Anche la ridenominazione di un ruolo (`ALTER ROLE myrole RENAME TO newname`) è limitata se la password utilizza l'algoritmo di hashing MD5.

Con questa funzionalità attiva, se si tenta di eseguire uno di questi comandi SQL senza le autorizzazioni di ruolo `rds_password`, viene generato il seguente errore:

```
ERROR: must be a member of rds_password to alter passwords
```

Si consiglia di concedere i privilegi `rds_password` solo a ruoli utilizzati esclusivamente per la gestione delle password. Se si concedono i privilegi `rds_password` agli utenti del database sprovvisti dei privilegi `rds_superuser`, è necessario concedere loro anche l'attributo `CREATEROLE`.

Assicurarsi di verificare i requisiti della password come la scadenza e la complessità necessaria sul lato client. Se si utilizza la propria utilità lato client per le modifiche relative alla password, l'utilità deve essere membro di `rds_password` e avere i privilegi `CREATE ROLE`.

Utilizzo delle crittografia password SCRAM per PostgreSQL

Il meccanismo SCRAM (Salted Challenge Response Authentication Mechanism) è un'alternativa all'algoritmo predefinito MD5 (Message Digest) di PostgreSQL per la crittografia delle password. Il meccanismo di autenticazione SCRAM è considerato più sicuro di MD5. Per ulteriori informazioni su questi due diversi approcci di protezione delle password, consulta la sezione relativa alla autenticazione password nella documentazione di PostgreSQL.

Si consiglia di utilizzare SCRAM anziché MD5 come schema di crittografia password per il . l'istanza database RDS per PostgreSQL. È un meccanismo crittografico di richiesta/risposta che utilizza l'algoritmo scram-sha-256 per l'autenticazione e la crittografia delle password.

Potrebbe essere necessario aggiornare le librerie per le applicazioni client per supportare SCRAM. Ad esempio, le versioni JDBC precedenti alla 42.2.0 non supportano SCRAM. Per ulteriori informazioni, consulta PostgreSQL JDBC Driver nella documentazione di PostgreSQL JDBC Driver. Per un elenco di altri driver PostgreSQL e il supporto SCRAM, consulta Elenco dei driver nella documentazione di PostgreSQL.

> **Note**
>
> RDS per PostgreSQL versione 13.1 e successive supportano scram-sha-256. Queste versioni consentono inoltre di configurare l'istanza database per richiedere SCRAM, come illustrato nelle procedure seguenti.

Configurazione di istanza database di RDS per PostgreSQL per richiedere SCRAM

puoi richiedere che l'istanza database di RDS per PostgreSQL accetti solo password che utilizzano l'algoritmo scram-sha-256.

> **Important**
>
> Per i proxy RDS esistenti con database PostgreSQL, se si modifica l'autenticazione del database in modo da utilizzare solo `SCRAM`, il proxy diventa non disponibile per un massimo di 60 secondi. Per evitare il problema, procedi in uno dei seguenti modi:

- Assicurati che il database consenta entrambe le autenticazioni `SCRAM` e `MD5`.
- Per utilizzare solo l'autenticazione `SCRAM`, crea un nuovo proxy, esegui la migrazione del traffico dell'applicazione sul nuovo proxy, quindi elimina il proxy precedentemente associato al database.

Prima di apportare modifiche al sistema, assicurati di comprendere il processo completo, come segue:

- Ottieni informazioni su tutti i ruoli e la crittografia password per tutti gli utenti del database.
- Verifica le impostazioni dei parametri per l'istanza database di RDS per PostgreSQL per i parametri che controllano la crittografia password.
- Se l'istanza database di RDS per PostgreSQL utilizza un gruppo di parametri predefinito, devi creare un gruppo di parametri database e applicarlo all'istanza database di RDS per PostgreSQL in modo da poter modificare i parametri quando necessario. Se l'istanza database di RDS per PostgreSQL utilizza un gruppo di parametri personalizzati, puoi modificare i parametri necessari in seguito nel processo, in base alle esigenze.
- Modifica il parametro `password_encryption` in `scram-sha-256`.
- Invia una notifica a tutti gli utenti del database per informarli che devono aggiornare le password. Esegui la stessa operazione per l'account `postgres`. Le nuove password sono crittografate e archiviate utilizzando l'algoritmo scram-sha-256.
- Verifica che tutte le password siano crittografate utilizzando come il tipo di crittografia.
- Se tutte le password utilizzano scram-sha-256, puoi modificare il `rds.accepted_password_auth_method` da `md5+scram` a `scram-sha-256`.

> **Warning**
>
> Dopo aver modificato `rds.accepted_password_auth_method` in scram-sha-256, gli eventuali utenti (ruoli) con password crittografate `md5` non potranno connettersi.

Preparazione alla richiesta di SCRAM per l'istanza database di RDS per PostgreSQL

Prima di apportare modifiche all'istanza database RDS per PostgreSQL, controlla tutti gli account utente del database esistenti. Inoltre, controlla il tipo di crittografia utilizzato per le password. Puoi

eseguire queste attività utilizzando l'estensione `rds_tools`. Questa estensione è supportata su RDS per PostgreSQL 13.1 e versioni successive.

Per ottenere un elenco di utenti del database (ruoli) e metodi di crittografia password

1. Utilizza `psql` per connetterti all'istanza database RDS per PostgreSQL, come mostrato di seguito.

```
psql --host=db-name.111122223333.aws-region.rds.amazonaws.com --port=5432 --
username=postgres --password
```

2. Installa l'estensione `rds_tools`.

```
postgres=> CREATE EXTENSION rds_tools;
CREATE EXTENSION
```

3. Ottieni un elenco di ruoli e crittografia.

```
postgres=> SELECT * FROM
      rds_tools.role_password_encryption_type();
```

L'output visualizzato è simile al seguente.

```
       rolname        | encryption_type
----------------------+-----------------
 pg_monitor           |
 pg_read_all_settings |
 pg_read_all_stats    |
 pg_stat_scan_tables  |
 pg_signal_backend    |
 lab_tester           | md5
 user_465             | md5
 postgres             | md5
(8 rows)
```

Creazione di un gruppo di parametri DB personalizzato

> **Note**
>
> Se l'istanza database di RDS per PostgreSQL utilizza già un gruppo di parametri personalizzati, non è necessario crearne uno nuovo.

Per una panoramica dei gruppi di parametri per Amazon RDS, consulta Utilizzo dei parametri sull'istanza database RDS for PostgreSQL.

Il tipo di crittografia password utilizzato per le password è impostato in un parametro, `password_encryption`. La crittografia consentita dall'istanza database di RDS per PostgreSQL è impostata in un altro parametro, `rds.accepted_password_auth_method`. La modifica di uno di questi rispetto ai valori predefiniti richiede la creazione di un gruppo di parametri database personalizzato e l'applicazione all'istanza.

Puoi anche utilizzare l'API RDS AWS Management Console o l'API RDS per creare un gruppo di parametri del). Per ulteriori informazioni, consulta

Ora puoi associare il gruppo di parametri personalizzati all'istanza database.

Creare un gruppo di parametri database personalizzato

1. Utilizza il comando CLI `create-db-parameter-group` per creare il gruppo di parametri database personalizzato. Questo esempio utilizza `postgres13` come l'origine per questo gruppo di parametri personalizzati.

   Per LinuxmacOS, oUnix:

   ```
   aws rds create-db-parameter-group --db-parameter-group-name 'docs-lab-scram-passwords' \
     --db-parameter-group-family postgres13  --description 'Custom parameter group for SCRAM'
   ```

   Per Windows:

   ```
   aws rds create-db-parameter-group --db-parameter-group-name "docs-lab-scram-passwords" ^
   ```

```
  --db-parameter-group-family postgres13  --description "Custom DB parameter group
for SCRAM"
```

2. Utilizza il comando CLI `modify-db-instance` per applicare questo gruppo di parametri personalizzati al cluster database RDS per PostgreSQL.

   Per LinuxmacOS, oUnix:

   ```
   aws rds modify-db-instance --db-instance-identifier 'your-instance-name' \
           --db-parameter-group-name "docs-lab-scram-passwords
   ```

   Per Windows:

   ```
   aws rds modify-db-instance --db-instance-identifier "your-instance-name" ^
           --db-parameter-group-name "docs-lab-scram-passwords
   ```

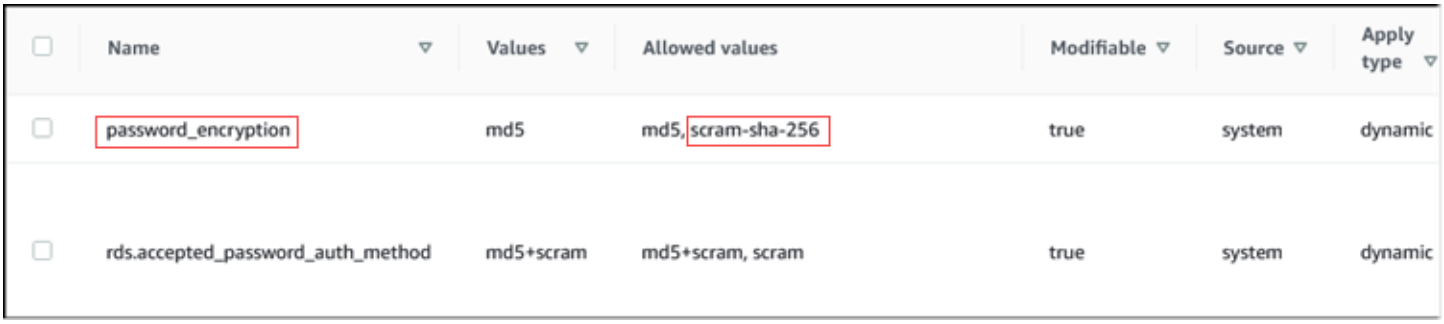
   Per ripetere la sincronizzazione dell'istanza database RDS per PostgreSQL con il gruppo di parametri DB personalizzato, è necessario riavviare l'istanza principale e tutte le altre istanze del cluster. Per ridurre al minimo l'impatto sugli utenti, pianifica questa operazione in modo che si verifichi durante la normale finestra di manutenzione.

Configurazione della crittografia password per utilizzare SCRAM

Il meccanismo di crittografia password utilizzato da un'istanza database RDS per PostgreSQL è impostato nel gruppo di parametri DB nel parametro `password_encryption`. I valori consentiti sono unset, `md5` o `scram-sha-256`. Il valore predefinito dipende dalla versione di RDS per PostgreSQL come segue:

- RDS per PostgreSQL 14 e versioni successive: l'impostazione predefinita è `scram-sha-256`
- RDS per PostgreSQL 13: l'impostazione predefinita è `md5`

Con un gruppo di parametri database personalizzato collegato all'istanza database di RDS per PostgreSQL, puoi modificare i valori per il parametro di crittografia password.



Name	Values	Allowed values	Modifiable	Source	Apply type
password_encryption	md5	md5, scram-sha-256	true	system	dynamic
rds.accepted_password_auth_method	md5+scram	md5+scram, scram	true	system	dynamic

Per modificare l'impostazione di crittografia password in scram-sha-256

- Modifica il valore della crittografia password in scram-sha-256, come mostrato di seguito. La modifica può essere applicata immediatamente perché il parametro è dinamico, quindi non è necessario un riavvio per rendere effettiva la modifica.

  Per LinuxmacOS, oUnix:

```
aws rds modify-db-parameter-group --db-parameter-group-name \
  'docs-lab-scram-passwords' --parameters
 'ParameterName=password_encryption,ParameterValue=scram-
sha-256,ApplyMethod=immediate'
```

  Per Windows:

```
aws rds modify-db-parameter-group --db-parameter-group-name ^
  "docs-lab-scram-passwords" --parameters
 "ParameterName=password_encryption,ParameterValue=scram-
sha-256,ApplyMethod=immediate"
```

Migrazione delle password per i ruoli utente in SCRAM

Puoi migrare le password per i ruoli utente a SCRAM come descritto di seguito.

Eseguire la migrazione delle password utente (ruolo) del database da MD5 a SCRAM

1. Accedi come utente amministratore (nome utente predefinito, `postgres`) come mostrato di seguito.

```
psql --host=db-name.111122223333.aws-region.rds.amazonaws.com --port=5432 --
username=postgres --password
```

2. Controlla l'impostazione del parametro `password_encryption` sull'istanza database RDS per PostgreSQL utilizzando il comando seguente.

```
postgres=> SHOW password_encryption;
 password_encryption
---------------------
 md5
 (1 row)
```

3. Modifica il valore di questo parametro in scram-sha-256. Si tratta di un parametro dinamico, quindi non è necessario riavviare l'istanza dopo aver apportato questa modifica. Controlla nuovamente il valore per essere certo che ora sia impostato su `scram-sha-256`, come descritto di seguito.

```
postgres=> SHOW password_encryption;
 password_encryption
---------------------
 scram-sha-256
 (1 row)
```

4. Invia una notifica a tutti gli utenti del database con la richiesta di modificare le password. Assicurati di modificare anche la password per l'account `postgres` (l'utente del database con privilegi `rds_superuser`).

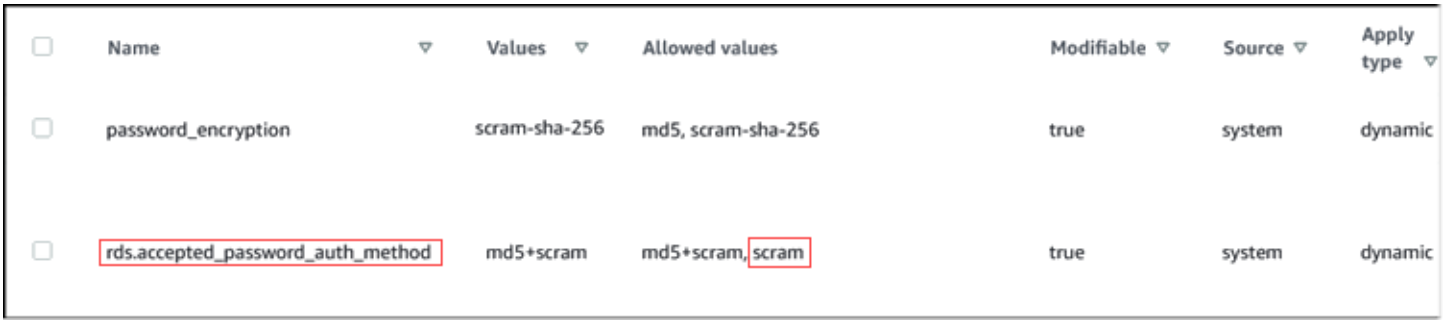
```
labdb=> ALTER ROLE postgres WITH LOGIN PASSWORD 'change_me';
ALTER ROLE
```

5. Ripeti il processo per tutti i database sul l'istanza database RDS per PostgreSQL.

Modifica del parametro per richiedere SCRAM

Questo è il passaggio finale del processo. Dopo aver apportato la modifica nella procedura seguente, gli eventuali account utente (ruoli) che ancora utilizzano la crittografia `md5` per le password non possono accedere al l'istanza database RDS per PostgreSQL.

Il `rds.accepted_password_auth_method` specifica il metodo di crittografia accettato dall'istanza database di RDS per PostgreSQL per una password utente durante il processo di accesso. Il valore predefinito è `md5+scram`, il che significa che entrambi i metodi sono accettati. Nell'immagine seguente, è disponibile l'impostazione predefinita per questo parametro.



Name	Values	Allowed values	Modifiable	Source	Apply type
password_encryption	scram-sha-256	md5, scram-sha-256	true	system	dynamic
rds.accepted_password_auth_method	md5+scram	md5+scram, scram	true	system	dynamic

I valori consentiti per questo parametro sono `md5+scram` o `scram`. La modifica del valore di questo parametro in `scram` lo rende un requisito.

Modificare il valore del parametro per richiedere l'autenticazione SCRAM per le password

1. Verifica che tutte le password degli utenti del database per tutti i database sull'istanza database di RDS per PostgreSQL utilizzino `scram-sha-256` per la crittografia password. A questo proposito, esegui la query su `rds_tools` per il ruolo (utente) e il tipo di crittografia, come segue.

```
postgres=> SELECT * FROM rds_tools.role_password_encryption_type();
  rolname        | encryption_type
  ---------------------+-----------------
  pg_monitor           |
  pg_read_all_settings |
  pg_read_all_stats    |
  pg_stat_scan_tables  |
  pg_signal_backend    |
  lab_tester           | scram-sha-256
  user_465             | scram-sha-256
  postgres             | scram-sha-256
  ( rows)
```

2. Ripeti la query su tutte le istanze database nel l'istanza database RDS per PostgreSQL.

   Se tutte le password utilizzano scram-sha-256, puoi procedere.

3. Modifica il valore dell'autenticazione password accettata in scram-sha-256, come riportato di seguito.

   Per LinuxmacOS, oUnix:

```
aws rds modify-db-parameter-group --db-parameter-group-name 'docs-lab-scram-
passwords' \
```

```
  --parameters
 'ParameterName=rds.accepted_password_auth_method,ParameterValue=scram,ApplyMethod=immediat
```

Per Windows:

```
aws rds modify-db-parameter-group --db-parameter-group-name "docs-lab-scram-
passwords" ^
  --parameters
 "ParameterName=rds.accepted_password_auth_method,ParameterValue=scram,ApplyMethod=immediat
```

Utilizzo della funzione di autovacuum di PostgreSQL in Amazon RDS for PostgreSQL

Consigliamo vivamente di usare la caratteristica di autovacuum per mantenere l'integrità dell'istanza database di PostgreSQL. La funzione di autovaacuum automatizza l'esecuzione del comando VACUUM e ANALYZE. Verifica la presenza di tabelle con un numero elevato di tuple inserite, aggiornate o eliminate. Dopo questa verifica, recupera lo storage rimuovendo i dati obsoleti o le tuple da database PostgreSQL.

Per impostazione predefinita, la funzione di autovacuum è attivata per le istanze database di Amazon RDS for PostgreSQL create utilizzando uno dei gruppi parametri del database PostgreSQL di default. Queste includono `default.postgres10`, `default.postgres11` e così via. Tutti i gruppi parametri del database PostgreSQL di default hanno un parametro `rds.adaptive_autovacuum` impostato su `1`, in questo modo viene quindi attivata la caratteristica. Per impostazione predefinita vengono impostati anche altri parametri di configurazione associati alla caratteristica di autovacuum. Poiché questi valori di default sono in qualche modo generici, è possibile trarre vantaggio dalla regolazione di alcuni parametri associati alla caratteristica di autovacuum per il carico di lavoro specifico.

Di seguito, puoi trovare ulteriori informazioni sulla funzione di autovacuum e su come regolare alcuni dei relativi parametri sulla tua istanza database di RDS for PostgreSQL. Per informazioni generali, consulta Best practice per l'utilizzo di PostgreSQL.

Argomenti

- Allocazione di memoria per il vacuum
- Riduzione della probabilità che si verifichi il wraparound dell'ID delle transazioni
- Determinare se le tabelle nel database devono essere sottoposte a vacuum

- Determinare quali tabelle sono attualmente idonee per l'Autovacuum
- Determinare se l'Auotvacuum è attualmente in esecuzione e per quanto tempo
- Esecuzione di un congelamento manuale del vacuum
- Indicizzare di nuovo una tabella quando l'autovacuum è in esecuzione
- Gestione di autovacuum con indici di grandi dimensioni
- Altri parametri che influenzano l'autovacuum
- Impostazione dei parametri di autovacuum a livello tabella
- Registrazione delle attività di autovacuum e vacuum

Allocazione di memoria per il vacuum

Uno dei parametri più importanti che influenzano le prestazioni della funzione di autovacuum è il parametro maintenance_work_mem. Questo parametro determina la quantità di memoria allocata per l'autovacuum da utilizzare per la scansione di una tabella di database e per contenere tutti gli ID di riga che verranno sottoposti ad autovacuum. Se si imposta il valore del parametro `maintenance_work_mem` troppo basso, il processo di vacuum potrebbe dover eseguire la scansione della tabella più volte per completare il lavoro. Queste scansioni multiple possono avere un impatto negativo sulle prestazioni

Quando si eseguono i calcoli per determinare il valore del parametro `maintenance_work_mem` tenere a mente due cose:

- L'unità predefinita per questo parametro è il kilobyte (KB).
- Il parametro `maintenance_work_mem` funziona insieme al parametro `autovacuum_max_workers`. Se si dispone di molte tabelle di piccole dimensioni, assegna più `autovacuum_max_workers` e meno `maintenance_work_mem`. Se si dispone di tabelle di grandi dimensioni (ad esempio, maggiori di 100 GB), allocare più memoria e meno processi di lavoro. Si deve avere abbastanza memoria allocata affinché si abbia esito positivo sulle tabelle più grandi. Ogni `autovacuum_max_workers` può utilizzare la memoria allocata. Quindi assicurati che la combinazione processi dei dipendenti e della memoria sia uguale alla memoria totale che si desidera allocare.

In termini generali, per gli host di grandi dimensioni, impostare il parametro `maintenance_work_mem` su un valore compreso tra uno e due gigabyte (tra 1.048.576 e 2.097.152 KB). Per gli host di dimensioni estremamente grandi, impostare il parametro su un valore compreso

tra due e quattro gigabyte (tra 2.097.152 e 4.194.304 KB). Il valore impostato per questo parametro dipende dal carico di lavoro. Amazon RDS ha aggiornato il valore di default per questo parametro in kilobyte come segue.

`GREATEST({DBInstanceClassMemory/63963136*1024},65536)`.

Riduzione della probabilità che si verifichi il wraparound dell'ID delle transazioni

In alcuni casi, le impostazioni del gruppo di parametri correlate all'autovacuum potrebbero non essere abbastanza aggressive da prevenire il wraparound dell'ID delle transazioni. Per risolvere questo problema, RDS for PostgreSQL offre un meccanismo che adatta automaticamente i valori di parametro autovacuum. Ottimizzazione adattiva del parametro di autovacuum è una caratteristica per RDS for PostgreSQL. Una spiegazione dettagliata di Wraparound della transazione si trova nella documentazione di PostgreSQL.

La regolazione adattiva del parametro autovacuum è attivata per impostazione predefinita per le istanze RDS for PostgreSQL con il parametro dinamico `rds.adaptive_autovacuum` impostato su ON. Si consiglia di tenere questa opzione attivata. Tuttavia, per disattivare l'ottimizzazione adattiva del parametro di autovacuum, impostare il parametro `rds.adaptive_autovacuum` su 0 o OFF.

Il wraparound dell'ID delle transazioni è ancora possibile quando Amazon RDS regola i parametri di autovacuum. Ti invitiamo a implementare un CloudWatch allarme Amazon per il wraparound degli ID delle transazioni. Per ulteriori informazioni, consulta il post Implementazione di un sistema di avvisi rapidi per il wraparound degli ID transazione in RDS for PostgreSQL del blog sui database di AWS.

Con l'ottimizzazione adattiva dei parametri dell'autovacuum attivata, Amazon RDS inizia a regolare i parametri dell'autovacuum quando la CloudWatch metrica `MaximumUsedTransactionIDs` raggiunge il valore del parametro o 500.000.000, a seconda di quale sia il maggiore. `autovacuum_freeze_max_age`

Amazon RDS continua ad adattare i parametri per l'autovacuum se una tabella continua a tendere verso i wraparound dell'ID della transazione. Ognuno di questi aggiustamenti dedica più risorse all'autovacuum per evitare il wraparound. Amazon RDS aggiorna i seguenti parametri correlati all'autovacuum:

- autovacuum_vacuum_cost_delay
- autovacuum_vacuum_cost_limit
- autovacuum_work_mem
- autovacuum_naptime

RDS modifica questi parametri solo se il nuovo valore rende l'autovacuum più aggressivo. I parametri vengono modificati nella memoria sull'istanza database. I valori nel gruppo di parametri non vengono modificati. Per visualizzare le impostazioni in memoria correnti, utilizzare il comando SQL PostgreSQL SHOW.

Quando Amazon RDS modifica uno qualsiasi dei parametri autovacuum, genera un evento per l'istanza database interessata. Questo evento è visibile sulla AWS Management Console e tramite l'API Amazon RDS. Dopo che la `MaximumUsedTransactionIDs` CloudWatch metrica è tornata al di sotto della soglia, Amazon RDS ripristina i parametri relativi all'autovacuum in memoria ai valori specificati nel gruppo di parametri. Quindi genera un altro evento corrispondente a questa modifica.

Determinare se le tabelle nel database devono essere sottoposte a vacuum

La seguente query può essere usata per mostrare il numero di transazioni non sottoposte a vacuum in un database. La colonna `datfrozenxid` di una riga di database `pg_database` è il margine inferiore sui normali ID di transazione che appaiono nel database. Questa colonna è il minimo dei valori `relfrozenxid` per tabella all'interno del database.

```
SELECT datname, age(datfrozenxid) FROM pg_database ORDER BY age(datfrozenxid) desc
 limit 20;
```

Ad esempio, i risultati dell'esecuzione della query precedente potrebbero essere i seguenti.

```
datname    | age
mydb       | 1771757888
template0  | 1721757888
template1  | 1721757888
rdsadmin   | 1694008527
postgres   | 1693881061
(5 rows)
```

Quando l'età di un database raggiunge i due miliardi di ID di transazioni, si verifica il wraparound ddell'ID della transazione (XID) e il database entra in modalità di sola lettura. Puoi utilizzare questa query per produrre un parametro ed eseguirla alcune volte al giorno. Per impostazione predefinita, l'autovacuum è impostato per mantenere l'età delle transazioni a non più di 200,000,000 (`autovacuum_freeze_max_age`).

Un esempio di strategia di monitoraggio potrebbe avere questo aspetto:

- Impostare il valore `autovacuum_freeze_max_age` su 200 milioni di transazioni.

- Se una tabella raggiunge 500 milioni di transazioni senza vacuum, viene attivato un allarme a bassa gravità. Questo non è un valore irragionevole, ma può indicare che l'autovacuum non riesce a mantenere il passo.
- Se una tabella invecchia a un miliardo, questo dovrebbe essere trattato come un allarme per cui intervenire. In generale, si desidera mantenere le età più vicine a `autovacuum_freeze_max_age` per motivi di prestazioni. Si consiglia di investigare utilizzando le raccomandazioni che seguono.
- Se una tabella raggiunge 1,5 milioni di transazioni senza vacuum, viene attivato un allarme a gravità elevata. A seconda della velocità con cui il database utilizza gli ID delle transazioni, questo allarme può indicare che il tempo del sistema per eseguire l'autovacuum sta per scadere. In questo caso, consigliamo di risolvere il problema immediatamente.

Se una tabella superando costantemente queste soglie, modifica ulteriormente i parametri dell'autovacuum. Per impostazione predefinita, l'utilizzo manuale di VACUUM (che ha disabilitato i ritardi basati sui costi) è più aggressivo dell'autovacuum predefinito, ma è anche più intrusivo per il sistema nel suo complesso.

Consigliamo quanto segue:

- Attiva un meccanismo di monitoraggio in modo da essere consapevole dell'età delle transazioni più vecchie.

  Per informazioni sulla creazione di un processo che fornisce avvisi sul wraparound degli ID transazione, consulta il post nel blog di AWS Database Implementazione di un sistema di avvisi rapidi per il wraparound degli ID transazione in Amazon RDS for PostgreSQL.
- Per le tabelle più occupate, eseguire regolarmente un congelamento manuale del vacuum durante una finestra di manutenzione, oltre a fare affidamento sull'autovacuum. Per informazioni sull'esecuzione di un congelamento manuale del vacuum, consulta Esecuzione di un congelamento manuale del vacuum.

Determinare quali tabelle sono attualmente idonee per l'Autovacuum

Spesso, una o due tabelle hanno bisogno del vacuum. Le tabelle il cui valore `relfrozenxid` sia maggiore del numero di transazioni in `autovacuum_freeze_max_age` sono sempre destinate all'autovacuum. Altrimenti, se il numero di tuple reso obsoleto dall'ultimo VACUUM supera la soglia del vacuum, la tabella viene sottoposta a vacuum.

La autovacuum threshold (soglia di autovacuum) viene definita come:

```
Vacuum-threshold = vacuum-base-threshold + vacuum-scale-factor * number-of-tuples
```

dove è, è e `vacuum base threshold` è`autovacuum_vacuum_threshold`. `vacuum scale factor` `autovacuum_vacuum_scale_factor` `number of tuples` `pg_class.reltuples`

Mentre sei connesso al database, esegui la seguente query per visualizzare un elenco di tabelle che Autovacuum vede come idonee per il vacuuming.

```
WITH vbt AS (SELECT setting AS autovacuum_vacuum_threshold FROM
pg_settings WHERE name = 'autovacuum_vacuum_threshold'),
vsf AS (SELECT setting AS autovacuum_vacuum_scale_factor FROM
pg_settings WHERE name = 'autovacuum_vacuum_scale_factor'),
fma AS (SELECT setting AS autovacuum_freeze_max_age FROM pg_settings WHERE name =
 'autovacuum_freeze_max_age'),
sto AS (select opt_oid, split_part(setting, '=', 1) as param,
split_part(setting, '=', 2) as value from (select oid opt_oid, unnest(reloptions)
 setting from pg_class) opt)
SELECT '"'||ns.nspname||'"."'||c.relname||'"' as relation,
pg_size_pretty(pg_table_size(c.oid)) as table_size,
age(relfrozenxid) as xid_age,
coalesce(cfma.value::float, autovacuum_freeze_max_age::float)
 autovacuum_freeze_max_age,
(coalesce(cvbt.value::float, autovacuum_vacuum_threshold::float) +
coalesce(cvsf.value::float,autovacuum_vacuum_scale_factor::float) * c.reltuples)
AS autovacuum_vacuum_tuples, n_dead_tup as dead_tuples FROM
pg_class c join pg_namespace ns on ns.oid = c.relnamespace
join pg_stat_all_tables stat on stat.relid = c.oid join vbt on (1=1) join vsf on (1=1)
 join fma on (1=1)
left join sto cvbt on cvbt.param = 'autovacuum_vacuum_threshold' and c.oid =
 cvbt.opt_oid
left join sto cvsf on cvsf.param = 'autovacuum_vacuum_scale_factor' and c.oid =
 cvsf.opt_oid
left join sto cfma on cfma.param = 'autovacuum_freeze_max_age' and c.oid = cfma.opt_oid
WHERE c.relkind = 'r' and nspname <> 'pg_catalog'
AND (age(relfrozenxid) >= coalesce(cfma.value::float, autovacuum_freeze_max_age::float)
OR coalesce(cvbt.value::float, autovacuum_vacuum_threshold::float) +
coalesce(cvsf.value::float,autovacuum_vacuum_scale_factor::float) *
c.reltuples <= n_dead_tup)
ORDER BY age(relfrozenxid) DESC LIMIT 50;
```

Determinare se l'Auotvacuum è attualmente in esecuzione e per quanto tempo

Se è necessario procedere manualmente con il vacuum in una tabella, devi determinare se l'autovacuum è attualmente in esecuzione. In tal caso, potrebbe essere necessario regolare i parametri per farlo eseguire in modo più efficiente oppure disattivare l'autovacuum temporaneamente in modo da poter eseguire manualmente il VACUUM.

Utilizzare la seguente query per determinare se l'autovacuum è in esecuzione, da quanto tempo è in esecuzione e se è in attesa su un'altra sessione.

```
SELECT datname, usename, pid, state, wait_event, current_timestamp - xact_start AS
 xact_runtime, query
FROM pg_stat_activity
WHERE upper(query) LIKE '%VACUUM%'
ORDER BY xact_start;
```

Dopo l'esecuzione della query, si dovrebbe visualizzare un output simile a quello riportato di seguito.

```
 datname | usename  |  pid  | state  | wait_event |      xact_runtime       | query
 --------+----------+-------+--------+------------+-------------------------
+-----------------------------------------------------------------------------------------------
 mydb    | rdsadmin | 16473 | active |            | 33 days 16:32:11.600656 |
 autovacuum: VACUUM ANALYZE public.mytable1 (to prevent wraparound)
 mydb    | rdsadmin | 22553 | active |            | 14 days 09:15:34.073141 |
 autovacuum: VACUUM ANALYZE public.mytable2 (to prevent wraparound)
 mydb    | rdsadmin | 41909 | active |            | 3 days 02:43:54.203349  |
 autovacuum: VACUUM ANALYZE public.mytable3
 mydb    | rdsadmin |   618 | active |            | 00:00:00                |
 SELECT datname, usename, pid, state, wait_event, current_timestamp - xact_start AS
 xact_runtime, query+
         |          |       |        |            |                         | FROM
 pg_stat_activity
           +
         |          |       |        |            |                         | WHERE
 query like '%VACUUM%'
          +
         |          |       |        |            |                         | ORDER BY
 xact_start;
       +
```

Diversi problemi possono provocare sessioni di autovacuum di lunga esecuzione (che durano più giorni). Il problema più comune è che il valore del parametro `maintenance_work_mem` è impostato come troppo basso per la dimensione della tabella o la frequenza degli aggiornamenti.

Consigliamo di utilizzare la seguente formula per impostare il valore del parametro `maintenance_work_mem`.

```
GREATEST({DBInstanceClassMemory/63963136*1024},65536)
```

Le sessioni autovacuum a breve esecuzioni possono anche indicare dei problemi:

- Può indicare che non ci sono abbastanza `autovacuum_max_workers` per il carico di lavoro. In questo caso, sarà necessario indicare il numero di lavoratori.
- Può indicare che esiste un danneggiamento dell'indice (l'autovacuum si blocca e si riavvia sulla stessa relazione ma non ci sono progressi). In questo caso, esegui un `vacuum freeze verbose` *`table`* manuale per vedere la causa esatta.

Esecuzione di un congelamento manuale del vacuum

Si potrebbe voler eseguire un vacuum manuale su una tabella che ha già un processo di vacuum in esecuzione. Ciò è utile se hai identificato una tabella con un'età che si avvicina a 2 miliardi (o al di sopra di qualsiasi soglia monitorata).

I seguenti passaggi sono delle linea guida, con diverse varianti del processo. Ad esempio, durante la verifica, supporre che il valore del parametro `maintenance_work_mem` sia stato impostato come troppo piccolo e che sia necessario agire immediatamente su una tabella. Tuttavia, probabilmente al momento non desideri che l'istanza non venga recapitata. Utilizzando le query delle sezioni precedenti, si determina quale tabella rappresenta il problema e si nota una sessione autovacuum a lunga esecuzione. Si sa che è necessario cambiare l'impostazione del parametro `maintenance_work_mem`, ma è necessario anche agire immediatamente ed eseguire il vacuum della tabella in questione. La procedura seguente mostra cosa fare in questa situazione.

Per eseguire manualmente un congelamento del vacuum

1. Aprire due sessioni nel database che contiene la tabella che si desidera sottoporre a vacuum. Per la seconda sessione, utilizzare "screen" o un'altra utility che mantiene la sessione se la connessione viene interrotta.

2. Nella sessione uno, ottieni l'ID di processo (PID) della sessione di autovacuum in esecuzione sulla tabella.

   Eseguire la query seguente per ottenere il PID della sessione di autovacuum.

```
SELECT datname, usename, pid, current_timestamp - xact_start
AS xact_runtime, query
FROM pg_stat_activity WHERE upper(query) LIKE '%VACUUM%' ORDER BY
xact_start;
```

3. Nella sessione due, calcoli la quantità di memoria necessaria per questa operazione. In questo esempio, stabiliamo che è possibile permettersi di utilizzare fino a 2 GB di memoria per questa operazione, pertanto abbiamo impostato `maintenance_work_mem` per la sessione corrente su 2 GB.

```
SET maintenance_work_mem='2 GB';
SET
```

4. Nella sessione 2, inviare un comando `vacuum freeze verbose` per la tabella. L'impostazione di verbose è utile perché, anche se al momento non vi è alcun rapporto sullo stato di avanzamento in PostgreSQL, è possibile visualizzare l'attività.

```
\timing on
Timing is on.
vacuum freeze verbose pgbench_branches;
```

```
INFO:  vacuuming "public.pgbench_branches"
INFO:  index "pgbench_branches_pkey" now contains 50 row versions in 2 pages
DETAIL:  0 index row versions were removed.
0 index pages have been deleted, 0 are currently reusable.
CPU 0.00s/0.00u sec elapsed 0.00 sec.
INFO:  index "pgbench_branches_test_index" now contains 50 row versions in 2 pages
DETAIL:  0 index row versions were removed.
0 index pages have been deleted, 0 are currently reusable.
CPU 0.00s/0.00u sec elapsed 0.00 sec.
INFO:  "pgbench_branches": found 0 removable, 50 nonremovable row versions
     in 43 out of 43 pages
DETAIL:  0 dead row versions cannot be removed yet.
There were 9347 unused item pointers.
0 pages are entirely empty.
CPU 0.00s/0.00u sec elapsed 0.00 sec.
```

```
VACUUM
Time: 2.765 ms
```

5. Nella sessione uno, se l'autovacuum bloccava la sessione di vacuum, in `pg_stat_activity` vedi che l'attesa è "T" per la sessione di vacuum. In questo caso, è necessario terminare il processo di autovacuum come segue.

   ```
   SELECT pg_terminate_backend('the_pid');
   ```

   In questo momento, inizia la sessione. È importante notare che l'autovacuum si riavvia immediatamente poiché questa tabella è probabilmente la più alta nella lista di lavori.

6. Avvia il comando `vacuum freeze verbose` nella sessione due, quindi termina il processo di autovacuum nella sessione uno.

Indicizzare di nuovo una tabella quando l'autovacuum è in esecuzione

Se un indice diventa corrotto, l'autovacuum continua a elaborare la tabella e avrà esito negativo. Setenti di eseguire un vacuum manuale in questa situazione, riceverai un messaggio di errore come il seguente.

```
postgres=>  vacuum freeze pgbench_branches;
ERROR: index "pgbench_branches_test_index" contains unexpected
   zero page at block 30521
HINT: Please REINDEX it.
```

Quando l'indice è corrotto e l'autovacuum sta tentando l'esecuzione sulla tabella, ci sarà una contesa con una sessione di autovacuum già in esecuzione. Quando si immette un comando REINDEX, si richiede un blocco esclusivo sulla tabella. Le operazioni in scrittura sono bloccate e anche quelle in lettura che utilizzano l'indice specifico.

Per indicizzare di nuovo una tabella quando l'autovacuum è in esecuzione sulla tabella

1. Apri due sessioni nel database che contiene la tabella da sottoporre a vacuum. Per la seconda sessione, utilizzare "screen" o un'altra utility che mantiene la sessione se la connessione viene interrotta.
2. Nella sessione numero uno, ottenere il PID della sessione di autovacuum in esecuzione sulla tabella.

Eseguire la query seguente per ottenere il PID della sessione di autovacuum.

```
SELECT datname, usename, pid, current_timestamp - xact_start
AS xact_runtime, query
FROM pg_stat_activity WHERE upper(query) like '%VACUUM%' ORDER BY
xact_start;
```

3. Nella sessione due, rilasciare il comando di reindicizzazione.

```
\timing on
Timing is on.
reindex index pgbench_branches_test_index;
REINDEX
  Time: 9.966 ms
```

4. Nella sessione uno, se l'autovacuum bloccava il processo, in `pg_stat_activity` vedi che l'attesa è "T" per la sessione di vacuum. In questo caso, termina il processo di autovacuum.

```
SELECT pg_terminate_backend('the_pid');
```

   In questo momento, inizia la sessione. È importante notare che l'autovacuum si riavvia immediatamente poiché questa tabella è probabilmente la più alta nella lista di lavori.

5. Avvia il comando nella sessione due, quindi termina il processo di autovacuum nella sessione 1.

Gestione di autovacuum con indici di grandi dimensioni

Come parte del funzionamento, autovacuum esegue diverse fasi di vacuum mentre viene eseguito su una tabella. Prima che la tabella venga pulita, tutti i suoi indici vengono prima sottoposti al vacuum. Quando si rimuovono più indici di grandi dimensioni, questa fase richiede una notevole quantità di tempo e risorse. Pertanto, come best practice, assicurati di controllare il numero di indici in una tabella ed eliminare gli indici non utilizzati.

Per questo processo, controlla innanzitutto la dimensione complessiva degli indici. Quindi, determina se ci sono indici potenzialmente inutilizzati da rimuovere come mostrato negli esempi seguenti.

Per verificare la dimensione della tabella e dei relativi indici

```
postgres=> select pg_size_pretty(pg_relation_size('pgbench_accounts'));
pg_size_pretty
```

```
6404 MB
(1 row)
```

```
postgres=> select pg_size_pretty(pg_indexes_size('pgbench_accounts'));
pg_size_pretty
11 GB
(1 row)
```

In questo esempio, la dimensione degli indici è maggiore della tabella. Questa differenza può causare problemi di prestazioni perché gli indici sono aumentati in dimensioni o inutilizzati, il che influisce sull'autovacuum e sulle operazioni di inserimento.

Per verificare la presenza di indici non utilizzati

Utilizzando la visualizzazione pg_stat_user_indexes, è possibile verificare la frequenza con cui viene utilizzato un indice con la colonna `idx_scan`. Nell'esempio seguente, gli indici non utilizzati hanno `idx_scan` con il valore `0`.

```
postgres=> select * from pg_stat_user_indexes where relname = 'pgbench_accounts' order
 by idx_scan desc;

relid  | indexrelid | schemaname | relname          | indexrelname          | idx_scan
 | idx_tup_read | idx_tup_fetch
-------+------------+------------+------------------+-----------------------+----------
+--------------+--------------
16433  | 16454      | public     | pgbench_accounts | index_f               | 6
 | 6            | 0
16433  | 16450      | public     | pgbench_accounts | index_b               | 3
 | 199999       | 0
16433  | 16447      | public     | pgbench_accounts | pgbench_accounts_pkey | 0
 | 0            | 0
16433  | 16452      | public     | pgbench_accounts | index_d               | 0
 | 0            | 0
16433  | 16453      | public     | pgbench_accounts | index_e               | 0
 | 0            | 0
16433  | 16451      | public     | pgbench_accounts | index_c               | 0
 | 0            | 0
16433  | 16449      | public     | pgbench_accounts | index_a               | 0
 | 0            | 0
(7 rows)
```

```
postgres=> select schemaname, relname, indexrelname, idx_scan from pg_stat_user_indexes
 where relname = 'pgbench_accounts' order by idx_scan desc;

schemaname  | relname          | indexrelname          | idx_scan
------------+------------------+-----------------------+----------
public      | pgbench_accounts | index_f               | 6
public      | pgbench_accounts | index_b               | 3
public      | pgbench_accounts | pgbench_accounts_pkey | 0
public      | pgbench_accounts | index_d               | 0
public      | pgbench_accounts | index_e               | 0
public      | pgbench_accounts | index_c               | 0
public      | pgbench_accounts | index_a               | 0
(7 rows)
```

Note

Queste statistiche sono incrementali dal momento in cui vengono ripristinate. Supponi di avere un indice utilizzato solo alla fine di un trimestre lavorativo o solo per un report specifico. È possibile che questo indice non sia stato utilizzato da quando le statistiche sono state ripristinate. Per ulteriori informazioni, consulta Funzioni statistiche. Gli indici utilizzati per garantire l'univocità non vengono sottoposti ad analisi e non devono essere identificati come indici non utilizzati. Per identificare gli indici non utilizzati, è necessario avere una conoscenza approfondita dell'applicazione e delle relative query.

Per verificare quando le statistiche sono state ripristinate l'ultima volta per un database, usa `pg_stat_database`

```
postgres=> select datname, stats_reset from pg_stat_database where datname =
 'postgres';

datname   | stats_reset
----------+-------------------------------
postgres  | 2022-11-17 08:58:11.427224+00
(1 row)
```

Vacuum di una tabella il più rapidamente possibile

RDS per PostgreSQL 12 e versioni successive

Se sono presenti troppi indici in una tabella di grandi dimensioni, l'istanza database potrebbe essere vicina al wraparound dell'ID di transazione (XID), ovvero quando il contatore XID arriva a zero. Se non controllata, questa situazione potrebbe causare la perdita di dati. Tuttavia, è possibile eseguire rapidamente il vacuum della tabella senza ripulire gli indici. In RDS per PostgreSQL 12 e versioni successive, puoi usare VACUUM con la clausola `INDEX_CLEANUP`.

```
postgres=> VACUUM (INDEX_CLEANUP FALSE, VERBOSE TRUE) pgbench_accounts;

INFO: vacuuming "public.pgbench_accounts"
INFO: table "pgbench_accounts": found 0 removable, 8 nonremovable row versions in 1 out
 of 819673 pages
DETAIL: 0 dead row versions cannot be removed yet, oldest xmin: 7517
Skipped 0 pages due to buffer pins, 0 frozen pages.
CPU: user: 0.01 s, system: 0.00 s, elapsed: 0.01 s.
```

Se è già in esecuzione una sessione di autovacuum, è necessario interromperla per iniziare il VACUUM manuale. Per informazioni sull'esecuzione di un congelamento manuale del vacuum, consulta Esecuzione di un congelamento manuale del vacuum.

> **Note**
>
> Se si evita di eseguire regolarmente la pulizia, le dimensioni dell'indice potrebbero aumentare, con ripercussioni sulle prestazioni complessive dell'analisi. Come best practice, utilizza la procedura precedente solo per evitare il wraparound dell'ID di transazione.

RDS per PostgreSQL 11 e versioni precedenti

Tuttavia, in RDS per PostgreSQL 11 e versioni precedenti, l'unico modo per eseguire il vacuum più rapidamente è riducendo il numero di indici su una tabella. L'eliminazione di un indice può influire sui piani di query. Ti consigliamo di eliminare prima gli indici inutilizzati, quindi quelli che hanno il wraparound XID molto vicino. Una volta completato il processo di vacuum, è possibile ricreare questi indici.

Altri parametri che influenzano l'autovacuum

La query seguente mostra i valori di alcuni dei parametri che influenzano direttamente l'autovacuum e il suo comportamento. I parametri di autovacuum vengono descritti in forma completa nella documentazione di PostgreSQL.

```
SELECT name, setting, unit, short_desc
FROM pg_settings
WHERE name IN (
'autovacuum_max_workers',
'autovacuum_analyze_scale_factor',
'autovacuum_naptime',
'autovacuum_analyze_threshold',
'autovacuum_analyze_scale_factor',
'autovacuum_vacuum_threshold',
'autovacuum_vacuum_scale_factor',
'autovacuum_vacuum_threshold',
'autovacuum_vacuum_cost_delay',
'autovacuum_vacuum_cost_limit',
'vacuum_cost_limit',
'autovacuum_freeze_max_age',
'maintenance_work_mem',
'vacuum_freeze_min_age');
```

Mentre questi riguardano tutti l'autovacuum, alcuni dei più importanti sono:

- maintenance_work_mem
- autovacuum_freeze_max_age
- autovacuum_max_workers
- autovacuum_vacuum_cost_delay
- autovacuum_vacuum_cost_limit

Impostazione dei parametri di autovacuum a livello tabella

Puoi impostare i parametri di archiviazione correlati all'autovacuum a livello di tabella, che può essere meglio di alterare il comportamento dell'intero database. Per le tabelle di grandi dimensioni, potrebbe essere necessario regolare impostazioni aggressive e si potrebbe non desiderare di eseguire l'autovacuum in questo modo per tutte le tabelle.

La query seguente mostra quali tabelle attualmente dispongono di opzioni a livello di tabella.

```
SELECT relname, reloptions
FROM pg_class
WHERE reloptions IS NOT null;
```

Un esempio in cui ciò potrebbe essere utile è per tabelle che sono molto più grandi rispetto al resto delle tabelle. Supponi di disporre di una tabella da 300 GB e di altre 30 tabelle da meno di un GB. Se disponi di una tabella da 300 GB e di altre 30 tabelle da meno di 1 GB, puoi impostare alcuni parametri specifici per la tabella di grandi dimensioni in modo da non alterare il comportamento dell'intero sistema.

```
ALTER TABLE mytable set (autovacuum_vacuum_cost_delay=0);
```

In questo modo si disattiva il ritardo dell'autovacuum basato sul costo per questa tabella a spese di un maggiore utilizzo delle risorse sul sistema. Normalmente, l'autovacuum si ferma per `autovacuum_vacuum_cost_delay` ogni volta che viene raggiunto `autovacuum_cost_limit`. Per ulteriori dettagli, consulta la documentazione di PostegreSQL relativa al vacuuming basato sul costo.

Registrazione delle attività di autovacuum e vacuum

Le informazioni sulle attività dell'autovacuum vengono inviate a `postgresql.log` in base al livello specificato nel parametro `rds.force_autovacuum_logging_level`. Di seguito sono riportati i valori consentiti per questo parametro e le versioni di PostgreSQL per le quali tale valore è l'impostazione predefinita:

- `disabled` (PostgreSQL 10, PostgreSQL 9.6)
- `debug5`, `debug4`, `debug3`, `debug2`, `debug1`
- `info` (PostgreSQL 12, PostgreSQL 11)
- `notice`
- `warning` (PostgreSQL 13 e versioni successive)
- `error`, log, `fatal`, `panic`

`rds.force_autovacuum_logging_level` funziona con il parametro `log_autovacuum_min_duration`. Il valore del parametro `log_autovacuum_min_duration` è la soglia (in millisecondi) al di sopra della quale vengono registrate le azioni dell'autovacuum. Un ambiente di `-1` non registra nulla, mentre un'impostazione di 0 registra tutte le

azioni. Come con `rds.force_autovacuum_logging_level`, i valori predefiniti per `log_autovacuum_min_duration` dipendono dalla versione, come segue:

- `10000 ms` – PostgreSQL 14, PostgreSQL 13, PostgreSQL 12 e PostgreSQL 11
- `(empty)` – Nessun valore predefinito per PostgreSQL 10 e PostgreSQL 9.6

Consigliamo di impostare `rds.force_autovacuum_logging_level` su `WARNING`. Consigliamo anche di impostare `log_autovacuum_min_duration` su un valore compreso tra 1000 e 5000. Un'impostazione di 5000 registri di attività che richiede più di 5000 millisecondi. Qualsiasi impostazione diversa da -1 registra anche i messaggi se l'azione dell'autovacuum viene ignorata a causa di un blocco in conflitto o di relazioni interrotte simultaneamente. Per ulteriori informazioni, consulta la pagina relativa al vacuum automatico nella documentazione di PostgreSQL.

Per risolvere i problemi, è possibile modificare il parametro `rds.force_autovacuum_logging_level` in uno dei livelli di debug, da `debug1` fino a `debug5` per le informazioni più dettagliate. Si consiglia di utilizzare le impostazioni di debug per brevi periodi di tempo e solo per la risoluzione dei problemi. Per ulteriori informazioni, consulta Quando registrare nella documentazione di PostgreSQL.

> **Note**
>
> PostgreSQL consente all'account `rds_superuser` di visualizzare le sessioni di autovacuum in `pg_stat_activity`. Ad esempio, è possibile identificare e terminare una sessione di autovacuum che blocca l'esecuzione di un comando o l'esecuzione più lenta di un comando vacuum emesso manualmente.

Utilizzo dei meccanismi di registrazione supportati da RDS for PostgreSQL

Esistono diversi parametri, estensioni e altri elementi configurabili che è possibile impostare per registrare le attività che avvengono sull'istanza database PostgreSQL. Questi sono i seguenti:

- Il parametro `log_statement` può essere utilizzato per registrare l'attività dell'utente nel database di PostgreSQL. Per ulteriori informazioni sulla registrazione di RDS per PostgreSQL e su come monitorare i registri, consulta File di log del database RDS per PostgreSQL.
- Il parametro `rds.force_admin_logging_level` registra le azioni dall'utente interno Amazon RDS (rdsadmin) nei database sull'istanza database. Scrive l'output nel registro degli errori

PostgreSQL. I valori consentiti sono `disabled`, `debug5`, `debug4`, `debug3`, `debug2`, `debug1`, `info`, `notice`, `warning`, `error`, log, `fatal` e `panic`. Il valore predefinito è `disabled`.

- Il parametro `rds.force_autovacuum_logging_level` può essere impostato per acquisire varie operazioni di pulizia automatica nel registro degli errori PostgreSQL. Per ulteriori informazioni, consulta Registrazione delle attività di autovacuum e vacuum.
- L'estensione PostgreSQL Audit (pgAudit) può essere installata e configurata per acquisire attività a livello di sessione o a livello di oggetto. Per ulteriori informazioni, consulta Utilizzo di pgAudit per registrare l'attività del database.
- L'estensione `log_fdw` consente di accedere al registro del motore del database utilizzando SQL. Per ulteriori informazioni, consulta Utilizzo dell'estensione log_fdw per accedere al registro di database utilizzando SQL.
- La libreria `pg_stat_statements` è specificata come predefinita per il parametro `shared_preload_libraries` in RDS per PostgreSQL versione 10 e successive. È questa libreria che puoi utilizzare per analizzare le query in esecuzione. Assicurati che `pg_stat_statements` sia impostato nel gruppo parametri del database. Per ulteriori informazioni sul monitoraggio dell'istanza database RDS for PostgreSQL utilizzando le informazioni fornite da questa libreria, consulta Statistiche SQL per RDS PostgreSQL.
- Il parametro `log_hostname` acquisisce nel log il nome host di ogni connessione client. Per RDS per PostgreSQL versione 12 e successive, questo parametro è impostato su `off` per impostazione predefinita. Se lo attivi, assicurati di monitorare i tempi di connessione della sessione. Quando è attivo, il servizio utilizza la richiesta di ricerca inversa del sistema dei nomi di dominio (DNS) per ottenere il nome host del client che sta effettuando la connessione e aggiungerlo al log di PostgreSQL. Ciò ha un impatto notevole sulla connessione della sessione. Ti consigliamo di attivare questo parametro solo a scopo di risoluzione dei problemi.

In termini generali, lo scopo della registrazione è consentire a DBA di monitorare, ottimizzare le prestazioni e risolvere i problemi. Molti dei log vengono caricati automaticamente su Amazon CloudWatch o Performance Insights. Qui vengono ordinati e raggruppati per fornire parametri completi per l'istanza database. Per ulteriori informazioni sul monitoraggio e sui parametri di Amazon RDS, consulta Monitoraggio di parametri in un'istanza Amazon RDS.

Gestione dei file temporanei con PostgreSQL

In PostgreSQL, una query che esegue operazioni di ordinamento e hash utilizza la memoria dell'istanza per archiviare i risultati fino al valore specificato nel parametro `work_mem`. Quando la memoria dell'istanza non è sufficiente, vengono creati file temporanei per archiviare i risultati.

Questi vengono scritti su disco per completare l'esecuzione della query. Successivamente, questi file vengono rimossi automaticamente al completamento della query. In RDS per PostgreSQL, questi file vengono archiviati in Amazon EBS sul volume di dati. Per ulteriori informazioni, consulta Storage delle istanze di database Amazon RDS. Puoi monitorare la metrica `FreeStorageSpace` pubblicata in CloudWatch per assicurarti che l'istanza database disponga di spazio di archiviazione libero sufficiente. Per ulteriori informazioni, consulta `FreeStorageSpace` .

Consigliamo di utilizzare istanze Letture ottimizzate per Amazon RDS per i carichi di lavoro che comportano più query simultanee che aumentano l'utilizzo di file temporanei. Queste istanze utilizzano l'archiviazione locale a livello di blocchi SSD basata su NVMe (Non-Volatile Memory Express). Per ulteriori informazioni, consulta Prestazioni delle query migliorate per RDS per PostgreSQL con Letture ottimizzate per Amazon RDS.

È possibile utilizzare i seguenti parametri e funzioni per gestire i file temporanei nell'istanza.

- **`temp_file_limit`**: questo parametro annulla qualsiasi query che superi la dimensione definita in KB dal parametro temp_files. Questo limite impedisce a qualsiasi query di essere eseguita all'infinito e di consumare spazio su disco con file temporanei. È possibile stimare il valore utilizzando i risultati del parametro `log_temp_files`. È consigliabile esaminare il comportamento del carico di lavoro e impostare il limite in base alla stima. Gli esempi seguenti mostrano come viene annullata una query quando supera il limite.

  ```
  postgres=> select * from pgbench_accounts, pg_class, big_table;
  ```

  ```
  ERROR: temporary file size exceeds temp_file_limit (64kB)
  ```

- **`log_temp_files`**: questo parametro invia messaggi a postgresql.log quando i file temporanei di una sessione vengono rimossi. Questo parametro produce log dopo che una query è stata completata correttamente. Pertanto, potrebbe non essere utile nella risoluzione dei problemi delle query attive e con tempi di esecuzione lunghi.

  L'esempio seguente mostra che quando la query viene completata correttamente, le voci vengono registrate nel file postgresql.log mentre i file temporanei vengono eliminati.

  ```
  2023-02-06 23:48:35 UTC:205.251.233.182(12456):adminuser@postgres:[31236]:LOG:
   temporary file: path "base/pgsql_tmp/pgsql_tmp31236.5", size 140353536
  ```

```
2023-02-06 23:48:35 UTC:205.251.233.182(12456):adminuser@postgres:[31236]:STATEMENT:
  select a.aid from pgbench_accounts a, pgbench_accounts b where a.bid=b.bid order by
 a.bid limit 10;
2023-02-06 23:48:35 UTC:205.251.233.182(12456):adminuser@postgres:[31236]:LOG:
 temporary file: path "base/pgsql_tmp/pgsql_tmp31236.4", size 180428800
2023-02-06 23:48:35 UTC:205.251.233.182(12456):adminuser@postgres:[31236]:STATEMENT:
  select a.aid from pgbench_accounts a, pgbench_accounts b where a.bid=b.bid order by
 a.bid limit 10;
```

- **pg_ls_tmpdir**: questa funzione disponibile in RDS per PostgreSQL 13 e versioni successive fornisce visibilità sull'attuale utilizzo dei file temporanei. La query completata non viene visualizzata nei risultati della funzione. Nell'esempio seguente, è possibile visualizzare i risultati di questa funzione.

```
postgres=> select * from pg_ls_tmpdir();
```

```
      name       |    size    |      modification
-----------------+------------+------------------------
 pgsql_tmp8355.1 | 1072250880 | 2023-02-06 22:54:56+00
 pgsql_tmp8351.0 | 1072250880 | 2023-02-06 22:54:43+00
 pgsql_tmp8327.0 | 1072250880 | 2023-02-06 22:54:56+00
 pgsql_tmp8351.1 |  703168512 | 2023-02-06 22:54:56+00
 pgsql_tmp8355.0 | 1072250880 | 2023-02-06 22:54:00+00
 pgsql_tmp8328.1 |  835031040 | 2023-02-06 22:54:56+00
 pgsql_tmp8328.0 | 1072250880 | 2023-02-06 22:54:40+00
(7 rows)
```

```
postgres=> select query from pg_stat_activity where pid = 8355;

query
-------------------------------------------------------------------------------------
select a.aid from pgbench_accounts a, pgbench_accounts b where a.bid=b.bid order by
 a.bid
(1 row)
```

Il nome del file include l'ID di elaborazione (PID) della sessione che ha generato il file temporaneo. Una query più avanzata, come nell'esempio seguente, esegue la somma dei file temporanei per ogni PID.

```
postgres=> select replace(left(name, strpos(name, '.')-1),'pgsql_tmp','') as pid,
 count(*), sum(size) from pg_ls_tmpdir() group by pid;
```

```
 pid  | count |   sum
------+-------+------------------
 8355 |     2 | 2144501760
 8351 |     2 | 2090770432
 8327 |     1 | 1072250880
 8328 |     2 | 2144501760
(4 rows)
```

- **pg_stat_statements**: se attivi il parametro pg_stat_statements, puoi visualizzare l'utilizzo medio dei file temporanei per chiamata. È possibile identificare il valore query_id della query e utilizzarlo per esaminare l'utilizzo dei file temporanei, come illustrato nell'esempio seguente.

```
postgres=> select queryid from pg_stat_statements where query like 'select a.aid from
 pgbench%';
```

```
       queryid
----------------------
 -7170349228837045701
(1 row)
```

```
postgres=> select queryid, substr(query,1,25), calls, temp_blks_read/calls
 temp_blks_read_per_call, temp_blks_written/calls temp_blks_written_per_call from
 pg_stat_statements where queryid = -7170349228837045701;
```

```
       queryid        |          substr           | calls | temp_blks_read_per_call |
 temp_blks_written_per_call
```

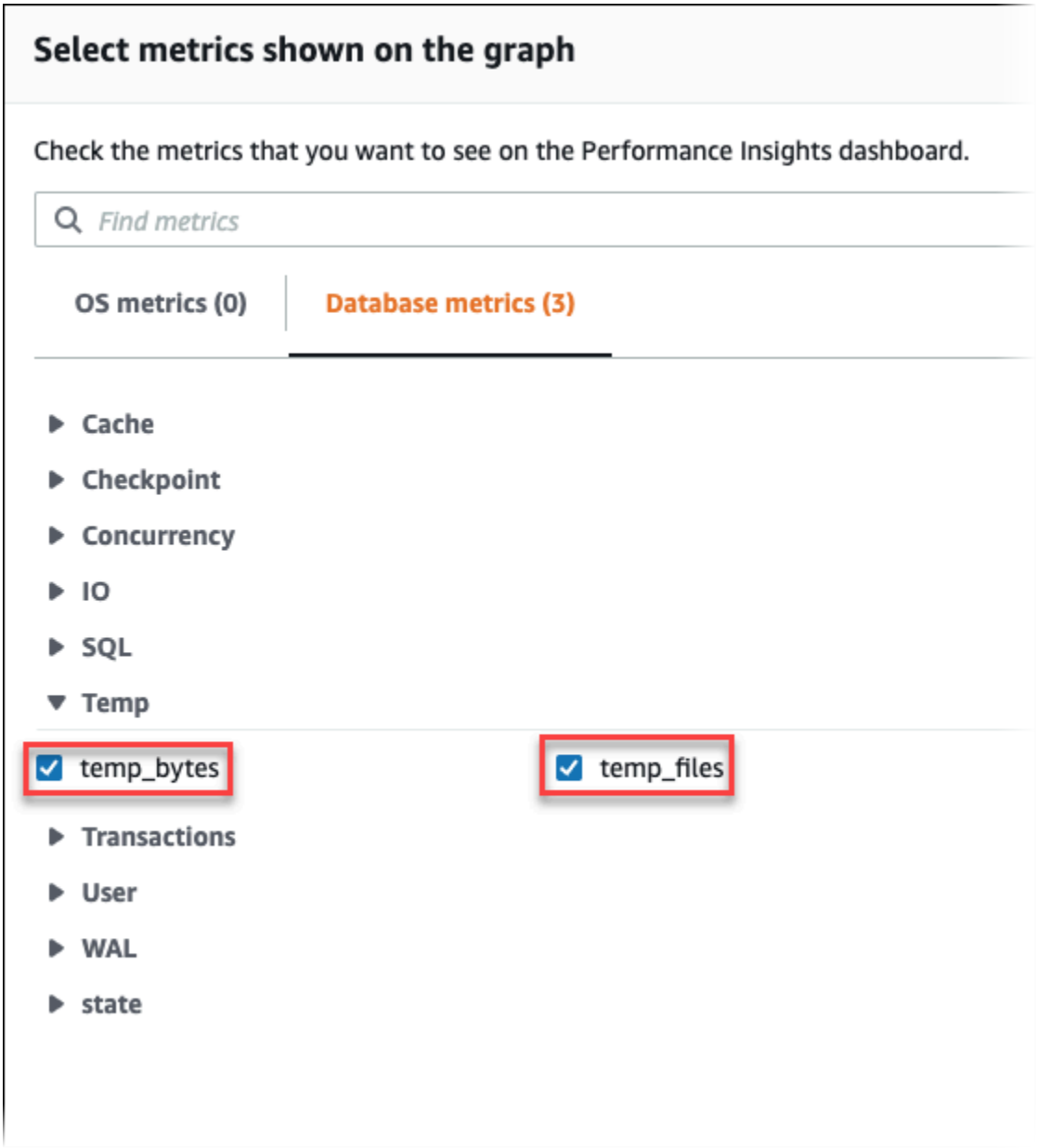
```
----------------------+---------------------------+-------+------------------------
+---------------------------
 -7170349228837045701 | select a.aid from pgbench |    50 |                 239226 |
                      388678
(1 row)
```

- **Performance Insights**: nel pannello di controllo di Approfondimenti sulle prestazioni, puoi visualizzare l'utilizzo dei file temporanei attivando le metriche temp_bytes e temp_files. Puoi quindi vedere la media di entrambe queste metriche e verificare se corrispondono al carico di lavoro delle query. La visualizzazione all'interno di Approfondimenti sulle prestazioni non evidenzia in modo specifico le query che generano file temporanei. Tuttavia, combinando le informazioni di Approfondimenti sulle prestazioni con la query mostrata per il parametro `pg_ls_tmpdir`, è possibile definire, analizzare e risolvere eventuali problemi a livello di modifiche del carico di lavoro delle query.

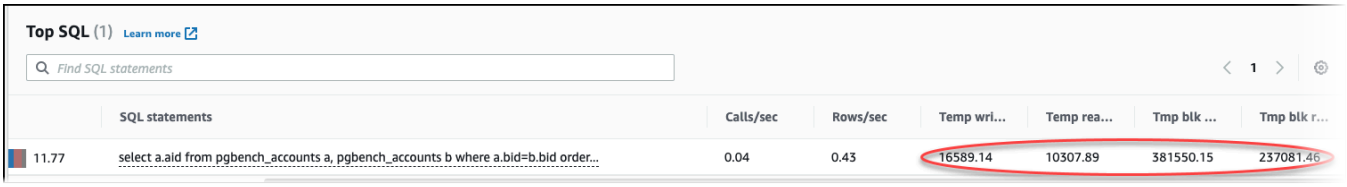
  Per ulteriori informazioni su come analizzare metriche e query con Approfondimenti sulle prestazioni, consulta Per analizzare il parametro utilizzando il pannello di controllo di Performance Insights

  Per visualizzare l'utilizzo dei file temporanei con Approfondimenti sulle prestazioni

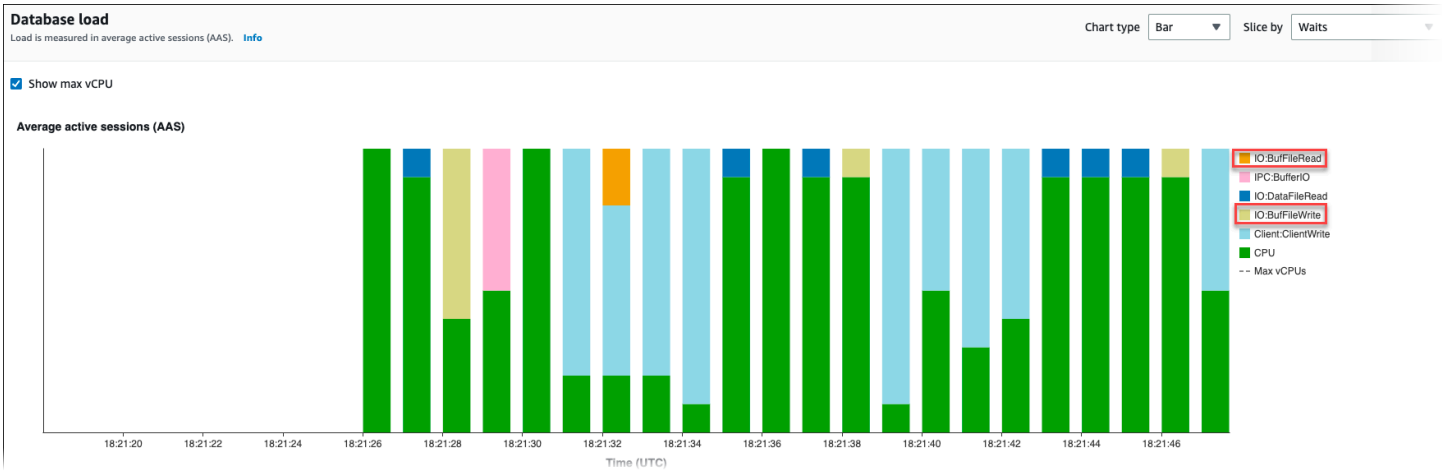
  1. Nel pannello di controllo di Approfondimenti sulle prestazioni, scegli Gestisci parametri.
  2. Seleziona Metriche del database e quindi seleziona le metriche temp_bytes e temp_files come illustrato nell'immagine seguente.



3. Nella scheda SQL principale, scegli l'icona Preferenze.
4. Nella finestra Preferenze, attiva le seguenti statistiche per visualizzarle nella scheda SQL principale e scegli Continua.
   - Scritture temporanee al secondo
   - Letture temporanee al secondo
   - Scritture temporanee in blocco a chiamata
   - Letture temporanee in blocco a chiamata
5. Il file temporaneo viene suddiviso quando viene combinato con la query visualizzata per `pg_ls_tmpdir`, come illustrato nell'esempio seguente.



Gli eventi `IO:BufFileRead` e `IO:BufFileWrite` si verificano quando le query principali del carico di lavoro creano spesso file temporanei. Puoi utilizzare Approfondimenti sulle prestazioni per identificare le query di livello superiore in attesa di `IO:BufFileRead` e `IO:BufFileWrite` esaminando la metrica Sessioni attive medie (AAS) nelle sezioni Caricamento del database e SQL principale.



Per ulteriori informazioni su come analizzare metriche principali ed eventi di attesa con Approfondimenti sulle prestazioni, consulta Panoramica della scheda Prime istruzioni SQL. Devi individuare e ottimizzare le query che causano un aumento dell'utilizzo dei file temporanei e dei relativi eventi di attesa. Per ulteriori informazioni su questi eventi di attesa e sulla loro correzione, consulta IO:BufFileRead e IO:BufFileWrite.

> **Note**
>
> Il parametro `work_mem` controlla quando l'operazione di ordinamento esaurisce la memoria; i risultati vengono scritti in file temporanei. Si consiglia di non modificare l'impostazione di questo parametro specificando un valore superiore al valore predefinito perché ciò causerebbe un maggiore utilizzo della memoria da parte di ciascuna sessione del database. Inoltre, una sessione che esegue unioni e ordinamenti complessi può eseguire operazioni parallele in cui ogni operazione consuma memoria.
> Come best practice, in presenza di un report di grandi dimensioni con più unioni e ordinamenti, imposta questo parametro a livello di sessione utilizzando il comando `SET`

`work_mem`. La modifica verrà quindi applicata solo alla sessione corrente e non comporterà la modifica del valore a livello globale.

Utilizzo di pgBadger per l'analisi del registro con PostgreSQL

È possibile utilizzare un analizzatore di registro come pgBadger per analizzare i registri di PostgreSQL. La documentazione pgBadger afferma che il modello %l (la linea di registro per sessione o processo) dovrebbe essere una parte del prefisso. Tuttavia, se si fornisce l'attuale `log_line_prefix` RDS come parametro per pgBadger, dovrebbe comunque produrre un report.

Ad esempio, il comando seguente formatta correttamente un file di log Amazon RDS for PostgreSQL datato 04-02-2014 usando pgBadger.

```
./pgbadger -f stderr -p '%t:%r:%u@%d:[%p]:' postgresql.log.2014-02-04-00
```

Utilizzo di PGSnapper per il monitoraggio di PostgreSQL

Puoi utilizzare PGSnapper per semplificare la raccolta periodica di statistiche e metriche relative alle prestazioni di Amazon RDS per PostgreSQL. Per ulteriori informazioni, consulta l'argomento relativo al monitoraggio delle prestazioni di Amazon RDS per PostgreSQL con PGSnapper.

Utilizzo dei parametri sull'istanza database RDS for PostgreSQL

In alcuni casi, è possibile creare un'istanza database RDS for PostgreSQL senza specificare un gruppo di parametri personalizzato. In tal caso, l'istanza database viene creata utilizzando il gruppo di parametri di default per la versione di PostgreSQL scelta. Ad esempio, supponiamo di creare un'istanza database RDS for PostgreSQL utilizzando PostgreSQL 13.3. In questo caso, l'istanza database viene creata utilizzando i valori nel gruppo di parametri per le versioni di PostgreSQL 13, `default.postgres13`.

Puoi anche creare i tuoi gruppi di parametri del database personalizzati. È necessario farlo se vuoi modificare qualsiasi impostazione per l'istanza database RDS for PostgreSQL rispetto ai valori predefiniti. Per scoprire come, consulta Utilizzo di gruppi di parametri.

Puoi monitorare le impostazioni sull'istanza database RDS for PostgreSQL in diversi modi. Puoi usare l' AWS Management Console AWS CLI, the o l'API Amazon RDS. Puoi anche eseguire query sui valori dalla tabella `pg_settings` di PostgreSQL della tua istanza, come illustrato di seguito.

```
SELECT name, setting, boot_val, reset_val, unit
 FROM pg_settings
 ORDER BY name;
```

Per ulteriori informazioni sui valori restituiti da questa query, consulta `pg_settings` nella documentazione di PostgreSQL.

Presta particolare attenzione quando modifichi le impostazioni `max_connections` e `shared_buffers` sull'istanza database RDS for PostgreSQL. Supponiamo, ad esempio, di modificare le impostazioni per `max_connections` o `shared_buffers` e di utilizzare valori troppo alti per il carico di lavoro effettivo. In questo caso, l'istanza database RDS for PostgreSQL non verrà avviata. In tal caso, viene visualizzato un errore simile al seguente in `postgres.log`:

```
2018-09-18 21:13:15 UTC::@:[8097]:FATAL:  could not map anonymous shared memory: Cannot
 allocate memory
2018-09-18 21:13:15 UTC::@:[8097]:HINT:  This error usually means that PostgreSQL's
 request for a shared memory segment
exceeded available memory or swap space. To reduce the request size (currently
 3514134274048 bytes), reduce
PostgreSQL's shared memory usage, perhaps by reducing shared_buffers or
 max_connections.
```

Tuttavia, non puoi modificare alcun valore delle impostazioni contenute nei gruppi parametri del database RDS for PostgreSQL di default. Per modificare le impostazioni per qualsiasi parametro, crea innanzitutto un gruppo parametri del database personalizzato. Quindi modifica le impostazioni di tale gruppo personalizzato e applica il gruppo di parametri personalizzato all'istanza database RDS for PostgreSQL. Per ulteriori informazioni, consulta Utilizzo di gruppi di parametri.

Esistono due tipi di parametri in RDS per PostgreSQL.

- Parametri statici: i parametri statici richiedono che l'istanza database RDS for PostgreSQL venga riavviata dopo una modifica in modo che il nuovo valore possa avere effetto.
- Parametri dinamici: i parametri dinamici non richiedono un riavvio dopo aver modificato le impostazioni.

> **Note**
>
> Se l'istanza database di RDS for PostgreSQL utilizza il gruppo parametri del database personalizzato, puoi modificare i valori dei parametri dinamici sull'istanza database in

esecuzione. Puoi farlo usando la AWS Management Console, la AWS CLI o l'API Amazon RDS.

Se disponi dei privilegi per farlo, puoi anche modificare i valori dei parametri utilizzando i comandi `ALTER DATABASE`, `ALTER ROLE` e `SET`.

Elenco dei parametri dell'istanza database di RDS for PostgreSQL

Nella tabella seguente sono elencati alcuni dei parametri disponibili in un'istanza database di RDS for PostgreSQL. Per visualizzare tutti i parametri disponibili, si utilizza il comando. describe-db-parameters AWS CLI Ad esempio, per ottenere l'elenco di tutti i parametri disponibili nel gruppo di parametri predefinito per RDS per PostgreSQL versione 13, esegui il seguente comando.

```
aws rds describe-db-parameters --db-parameter-group-name default.postgres13
```

Puoi anche utilizzare la console. Seleziona Parameter groups (Gruppi di parametri) nel menu di Amazon RDS, quindi scegli il gruppo di parametri tra quelli disponibili nella tua Regione AWS.

Nome del parametro	Apply_Type	Descrizione
application_name	Dinamico	Imposta il nome dell'applicazione da riportare nelle statistiche e nei registri.
archive_command	Dinamico	Imposta il comando shell che verrà chiamato per archiviare un file WAL.
array_nulls	Dinamico	Abilita l'inserimento di elementi NULL negli array.
authentication_timeout	Dinamico	Imposta il tempo massimo concesso per completare l'autenticazione del client.
autovacuum	Dinamico	Avvia il sottoprocesso autovacuum.
autovacuum_analyze_scale_factor	Dinamico	Numero di inserti, aggiornamenti o eliminazioni di tupla prima di analizzare come una frazione di reltuple.
autovacuum_analyze_threshold	Dinamico	Numero minimo di inserti, aggiornamenti o eliminazioni di tupla prima di analizzare.
autovacuum_freeze_max_age	Statico	Età nella quale eseguire l'autovacuum in una tabella per impedire il wraparound ID della transazione.

Nome del parametro	Apply_Type	Descrizione
autovacuum_naptime	Dinamico	Periodo di inattività tra le esecuzioni di autovacuum.
autovacuum_max_workers	Statico	Imposta il numero massimo di processi dipendenti di autovacuum in esecuzione simultanea
autovacuum_vacuum_cost_delay	Dinamico	Ritardo del costo del vacuum, in millisecondi, per l'autovacuum.
autovacuum_vacuum_cost_limit	Dinamico	Quantità del costo del vacuum disponibile prima del napping, per l'autovacuum.
autovacuum_vacuum_scale_factor	Dinamico	Numero di aggiornamenti o eliminazioni di tupla prima del vacuum come una frazione di reltuple.
autovacuum_vacuum_threshold	Dinamico	Numero minimo di aggiornamenti o eliminazioni di tupla prima del vacuum.
backslash_quote	Dinamico	Imposta se una barra rovesciata (\) è consentita nelle stringhe letterali.
bgwriter_delay	Dinamico	Tempo di inattività della scrittura di sfondo tra i round.
bgwriter_lru_maxpages	Dinamico	Numero massimo di scrittura di sfondo delle pagine LRU da ripulire per round.
bgwriter_lru_multiplier	Dinamico	Multiplo dell'uso medio del buffer da liberare per round.
bytea_output	Dinamico	Imposta il formato di output per byte.
check_function_bodies	Dinamico	Controlla i corpi delle funzioni durante CREATE FUNCTION.

Nome del parametro	Apply_Type	Descrizione
checkpoint_completion_target	Dinamico	Tempo trascorso a ripulire i buffer sporchi durante il checkpoint, come una frazione dell'intervallo di checkpoint.
checkpoint_segments	Dinamico	Imposta la distanza massima nei segmenti di registro tra i checkpoint WAL (write-ahead log) automatici.
checkpoint_timeout	Dinamico	Imposta il tempo massimo tra i checkpoint WAL automatici.
checkpoint_warning	Dinamico	Abilita gli avvisi se i segmenti del checkpoint sono riempiti più frequentemente di così.
client_connection_check_interval	Dinamico	Imposta l'intervallo di tempo tra i controlli di disconnessione durante l'esecuzione di query.
client_encoding	Dinamico	Imposta la codifica dell'impostazione del carattere del client.
client_min_messages	Dinamico	Imposta i livelli dei messaggi che vengono inviati al client.
commit_delay	Dinamico	Imposta il ritardo in microsecondi tra il commit della transazione e la pulizia del WAL su disco.
commit_siblings	Dinamico	Imposta le transazioni aperte simultanee minime prima di eseguire commit_delay.
constraint_exclusion	Dinamico	Consente al pianificatore di utilizzare i vincoli per ottimizzare le query.
cpu_index_tuple_cost	Dinamico	Imposta la stima del pianificatore del costo di elaborazione di ciascuna voce di indice durante una scansione dell'indice.

Nome del parametro	Apply_Type	Descrizione
cpu_operator_cost	Dinamico	Imposta la stima del pianificatore del costo di elaborazione di ciascuna chiamata dell'operatore o della funzione.
cpu_tuple_cost	Dinamico	Imposta la stima del pianificatore del costo di elaborazione di ciascuna tupla (riga).
cursor_tuple_fraction	Dinamico	Imposta la stima del pianificatore della frazione delle righe del cursore che verranno recuperate.
datestyle	Dinamico	Imposta il formato del display per i valori di data e ora.
deadlock_timeout	Dinamico	Imposta il tempo di attesa su un lock prima di verificare il deadlock.
debug_pretty_print	Dinamico	I trattini analizzano e visualizzano le visualizzazioni dell'albero.
debug_print_parse	Dinamico	Registra ogni albero di analisi della query.
debug_print_plan	Dinamico	Registra ogni programma di esecuzione della query.
debug_print_rewritten	Dinamico	Registra ogni albero di analisi riscritto della query.
default_statistics_target	Dinamico	Imposta la destinazione della statistica predefinita.
default_tablespace	Dinamico	Imposta il tablespace predefinito per la creazione di tabelle e indici.
default_transaction_deferrable	Dinamico	Imposta lo stato differibile predefinito delle nuove transazioni.

Nome del parametro	Apply_Type	Descrizione
default_transaction_isolation	Dinamico	Imposta il livello di isolamento della transazione di ogni nuova transazione.
default_transaction_read_only	Dinamico	Imposta lo stato di sola lettura predefinito delle nuove transazioni.
default_with_oids	Dinamico	Crea nuove tabelle con ID oggetto (OID) per impostazione predefinita.
effective_cache_size	Dinamico	Imposta l'ipotesi del pianificatore sulla dimensione della cache del disco.
effective_io_concurrency	Dinamico	Numero di richieste simultanee che possono essere gestite in modo efficace dal sottosistema del disco.
enable_bitmapscan	Dinamico	Abilita l'utilizzo da parte del pianificatore di piani di scansione bitmap.
enable_hashagg	Dinamico	Abilita l'utilizzo da parte del pianificatore di piani di aggregazione hash.
enable_hashjoin	Dinamico	Abilita l'utilizzo da parte del pianificatore di piani di unione hash.
enable_indexscan	Dinamico	Abilita l'utilizzo da parte del pianificatore di piani di scansione dell'indice.
enable_material	Dinamico	Abilita l'utilizzo da parte del pianificatore della materializzazione.
enable_mergejoin	Dinamico	Abilita l'utilizzo da parte del pianificatore di piani di unione.
enable_nestloop	Dinamico	Abilita l'utilizzo da parte del pianificatore di piani di unione a ciclo nested.

Nome del parametro	Apply_Type	Descrizione
enable_seqscan	Dinamico	Abilita l'utilizzo da parte del pianificatore di piani di scansione sequenziali.
enable_sort	Dinamico	Abilita l'utilizzo da parte del pianificatore di passaggi di ordinamento espliciti.
enable_tidscan	Dinamico	Abilita l'utilizzo da parte del pianificatore di piani di scansione TID.
escape_string_warning	Dinamico	Avvisa circa la perdita di barre rovesciate (\) nelle stringhe letterali ordinarie.
extra_float_digits	Dinamico	Imposta il numero di cifre visualizzate per i valori del punto variabile.
from_collapse_limit	Dinamico	Imposta la dimensione dell'elenco FROM oltre la quale le sottoquery non vengono compresse.
fsync	Dinamico	Forza la sincronizzazione degli aggiornamenti sul disco.
full_page_writes	Dinamico	Scrive pagine intere su WAL quando viene modificato per la prima volta dopo un checkpoint.
geqo	Dinamico	Abilita l'ottimizzazione genetica delle query.
geqo_effort	Dinamico	GEQO: lo sforzo viene utilizzato per impostare il valore predefinito per altri parametri GEQO.
geqo_generations	Dinamico	GEQO: numero di iterazioni dell'algoritmo.
geqo_pool_size	Dinamico	GEQO: numero di individui nella popolazione.
geqo_seed	Dinamico	GEQO: seme per la selezione casuale del percorso.

Nome del parametro	Apply_Type	Descrizione
geqo_selection_bias	Dinamico	GEQO: pressione selettiva all'interno della popolazione.
geqo_threshold	Dinamico	Imposta la soglia degli elementi FROM oltre i quali viene utilizzato GEQO.
gin_fuzzy_search_limit	Dinamico	Imposta il risultato massimo consentito per la ricerca esatta da GIN.
hot_standby_feedback	Dinamico	Determina se un hot standby invia messaggi di feedback allo standby principale o upstream.
intervalstyle	Dinamico	Imposta il formato del display per i valori dell'intervallo.
join_collapse_limit	Dinamico	Imposta la dimensione dell'elenco FROM oltre la quale i costrutti JOIN non vengono appiattiti.
lc_messages	Dinamico	Imposta la lingua nella quale vengono visualizzati i messaggi.
lc_monetary	Dinamico	Imposta l'ambientazione per la formattazione degli importi monetari.
lc_numeric	Dinamico	Imposta l'ambientazione per la formattazione degli numeri.
lc_time	Dinamico	Imposta l'ambientazione per la formattazione dei valori di data e ora.
log_autovacuum_min_duration	Dinamico	Imposta il tempo minimo di esecuzione al di sopra del quale verranno registrate le azioni di autovacuum.
log_checkpoints	Dinamico	Registra ogni checkpoint.
log_connections	Dinamico	Registra ogni connessione riuscita.

Nome del parametro	Apply_Type	Descrizione
log_disconnections	Dinamico	Registra la fine di una sessione, compresa la durata.
log_duration	Dinamico	Registra la durata di ogni istruzione SQL completata.
log_error_verbosity	Dinamico	Imposta la verbosità dei messaggi registrati.
log_executor_stats	Dinamico	Scrive le statistiche sulla prestazione degli esecutori nel registro del server.
log_filename	Dinamico	Imposta il modello del nome del file per i file di registro.
log_file_mode	Dinamico	Imposta le autorizzazioni file per i file di registro. Il valore predefinito è 0644.
log_hostname	Dinamico	Registra il nome host nei registri delle connessioni. A partire da PostgreSQL 12 e versioni successive, questo parametro è "disattivato" per impostazione predefinita. Quando è attivato, la connessione utilizza la ricerca inversa del DNS per ottenere il nome host che viene acquisito nei log delle connessioni. Se attivi questo parametro, è necessario monitorare l'impatto che ha sul tempo necessario per stabilire le connessioni.
log_line_prefix	Dinamico	Controlla le informazioni con prefisso su ciascuna riga di registro.
log_lock_waits	Dinamico	Registra lunghe attese di lock.
log_min_duration_statement	Dinamico	Imposta il tempo minimo di esecuzione al di sopra del quale verranno registrate le istruzioni.

Nome del parametro	Apply_Type	Descrizione
log_min_error_statement	Dinamico	Fa sì che tutte le istruzioni che generano un errore pari o superiore a questo livello vengano registrate.
log_min_messages	Dinamico	Imposta i livelli dei messaggi che vengono registrati.
log_parser_stats	Dinamico	Scrive le statistiche sulla prestazione del decodificatore nel registro del server.
log_planner_stats	Dinamico	Scrive le statistiche sulla prestazione del programmatore nel registro del server.
log_rotation_age	Dinamico	La rotazione del file di registro automatico avverrà dopo N minuti.
log_rotation_size	Dinamico	La rotazione del file di registro automatico avverrà dopo N kilobyte.
log_statement	Dinamico	Imposta il tipo di istruzioni registrate.
log_statement_stats	Dinamico	Scrive le statistiche cumulative sulla prestazione nel registro del server.
log_temp_files	Dinamico	Registra l'uso di file temporanei più grandi di questo numero di kilobyte.
log_timezone	Dinamico	Imposta il fuso orario da utilizzare nei messaggi di registro.
log_truncate_on_rotation	Dinamico	Tronca i file di log esistenti con lo stesso nome durante la rotazione del registro.
logging_collector	Statico	Avvia un processo secondario per acquisire l'output stderr e/o csvlogs nei file di log.
maintenance_work_mem	Dinamico	Imposta la memoria massima da utilizzare per le operazioni di manutenzione.

Nome del parametro	Apply_Type	Descrizione
max_connections	Statico	Imposta il numero massimo di connessioni simultanee.
max_files_per_process	Statico	Imposta il numero massimo di file aperti in modo simultaneo per ogni processo del server.
max_locks_per_transaction	Statico	Imposta il numero massimo di lock per transazione.
max_pred_locks_per_transaction	Statico	Imposta il numero massimo di lock del predicato per transazione.
max_prepared_transactions	Statico	Imposta il numero massimo di transazioni preparati in modo simultaneo.
max_stack_depth	Dinamico	Imposta la profondità di stack massima in kilobyte.
max_standby_archive_delay	Dinamico	Imposta il ritardo massimo prima di annullare le query quando un server hot standby elabora i dati WAL archiviati.
max_standby_streaming_delay	Dinamico	Imposta il ritardo massimo prima di annullare le query quando un server hot standby elabora i dati WAL in streaming.
max_wal_size	Dinamico	Imposta la dimensione WAL (MB) che attiva un punto di controllo. Per tutte le versioni successive a RDS per PostgreSQL 10, l'impostazione predefinita è almeno 1 GB (1024 MB). Ad esempio, l'impostazione `max_wal_size` per RDS per PostgreSQL 14 è 2 GB (2048 MB). Utilizza il comando `SHOW max_wal_size;` sull'istanza database RDS per PostgreSQL per visualizzare il valore corrente.

Nome del parametro	Apply_Type	Descrizione
min_wal_size	Dinamico	Imposta la dimensione minima di contrazione di WAL. Per PostgreSQL versione 9.6 e precedenti, min_wal_size è nelle unità di 16 MB. Per PostgreSQL versione 10 e successive, min_wal_size è nelle unità di 1 MB.
quote_all_identifiers	Dinamico	Aggiunge virgolette (") a tutti gli identificatori quando si generano i frammenti SQL.
random_page_cost	Dinamico	Imposta la stima del pianificatore del costo di una pagina del disco recuperata in modo non sequenziale. Questo parametro non ha valore a meno che non sia attivata la gestione del piano di query. Quando la gestione del piano di query è attiva, il valore di default per questo parametro è 4.
rds.adaptive_autovacuum	Dinamico	Regola in modo automatico i parametri di autovacuum ogni qualvolta la soglia dell'ID di transazione venga superata.
rds.force_ssl	Dinamico	Richiede l'uso di connessioni SSL. Il valore predefinito è impostato su 1 (on) per RDS per PostgreSQL versione 15. Tutti gli altri database RDS per PostgreSQL versione principale 14 e precedenti hanno il valore predefinito impostato su 0 (off).
rds.local_volume_spill_enabled	Statico	Consente la scrittura di file di spill logici nel volume locale.
rds.log_retention_period	Dinamico	Imposta la conservazione dei registri in modo che Amazon RDS elimini i registri PostgreSQL più vecchi di n minuti.

Nome del parametro	Apply_Type	Descrizione
rds.rds_superuser_reserved_connections	Statico	Imposta il numero di slot di connessione riservati a rds_superusers. Questo parametro è disponibile solo nelle versioni 15 e precedenti. Per ulteriori informazioni, consulta la documentazione PostgreSQL reserved_connections.
rds.restrict_password_commands	Statico	Limita chi può gestire le password per gli utenti con il ruolo rds_password . Impostare questo parametro a 1 per abilitare la limitazione per la password. Il valore predefinito è 0.
search_path	Dinamico	Imposta l'ordine di ricerca dello schema per i nomi che non sono qualificati come schema.
seq_page_cost	Dinamico	Imposta la stima del pianificatore del costo di una pagina del disco recuperata in modo sequenziale.
session_replication_role	Dinamico	Imposta il comportamento delle sessioni per i trigger e le regole di riscrittura.
shared_buffers	Statico	Imposta il numero di buffer di memoria condivisa utilizzati dal server.
shared_preload_libraries	Statico	Elenca le librerie condivise da precaricare nell'istanza database di RDS per PostgreSQL. I valori supportati includono auto_explain, orafce, pgaudit, pglogical, pg_bigm, pg_cron, pg_hint_plan, pg_prewarm, pg_similarity, pg_stat_statements, pg_tle, pg_transport, plprofiler e plrust.
ssl	Dinamico	Abilita le connessioni SSL.
sql_inheritance	Dinamico	Fa sì che le sottotabelle vengano incluse per impostazione predefinita in vari comandi.

Nome del parametro	Apply_Type	Descrizione
ssl_renegotiation_limit	Dinamico	Imposta la quantità di traffico da inviare e ricevere prima di rinegoziare le chiavi di crittografia.
standard_conforming_strings	Dinamico	Fa sì che le stringhe ... trattino letteralmente le barre rovesciate.
statement_timeout	Dinamico	Imposta la durata massima concessa di ogni istruzione.
synchronize_seqscans	Dinamico	Abilita le scansioni sequenziali sincronizzate.
synchronous_commit	Dinamico	Imposta il livello di sincronizzazione delle transazioni correnti.
tcp_keepalives_count	Dinamico	Numero massimo di ritrasmissioni keepalive TCP.
tcp_keepalives_idle	Dinamico	Tempo tra l'emissione di keepalive TCP.
tcp_keepalives_interval	Dinamico	Tempo tra la ritrasmissione di keepalive TCP.
temp_buffers	Dinamico	Imposta il numero massimo di buffer temporanei utilizzati da ogni sessione.
temp_file_limit	Dinamico	Imposta la dimensione massima in KB fino a cui i file temporanei possono crescere.
temp_tablespaces	Dinamico	Imposta i tablespace da usare per le tabelle temporanee e i file di ordinamento.

Nome del parametro	Apply_Type	Descrizione
timezone	Dinamico	Imposta il fuso orario per la visualizzazione e l'interpretazione dei timestamp. Internet Assigned Numbers Authority (IANA) pubblica nuovi fusi orari all'indirizzo https://www.iana.org/time-zones più volte all'anno. Ogni volta che RDS rilascia una nuova versione di manutenzione secondaria di PostgreSQL, la versione viene fornita con i dati sul fuso orario più recenti al momento del rilascio. Quando utilizzi le versioni più recenti di RDS per PostgreSQL, hai a disposizione i dati recenti relativi ai fusi orari di RDS. Per assicurarti che l'istanza database disponga dei dati più aggiornati relativi ai fusi orari, ti consigliamo di eseguire l'aggiornamento a una versione successiva del motore di database. Non è possibile modificare manualmente le tabelle dei fusi orari nelle istanze database di PostgreSQL. RDS non modifica né ripristina i dati dei fusi orari delle istanze database in esecuzione. I nuovi dati dei fusi orari vengono installati solo quando si esegue un aggiornamento della versione del motore di database.
track_activities	Dinamico	Raccoglie informazioni sull'esecuzione dei comandi.
track_activity_query_size	Statico	Imposta la dimensione riservata per pg_stat_activity.current_query, in byte.
track_counts	Dinamico	Raccoglie statistiche sull'attività del database.
track_functions	Dinamico	Raccoglie statistiche a livello di funzione sull'attività del database.

Nome del parametro	Apply_Type	Descrizione
track_io_timing	Dinamico	Raccoglie statistiche di temporizzazione sull'attività di I/O del database.
transaction_deferrable	Dinamico	Indica se rinviare una transazione serializzabile di sola lettura fino a quando non può essere eseguita senza possibili errori di serializzazione.
transaction_isolation	Dinamico	Imposta il livello di isolamento delle transazioni attuali.
transaction_read_only	Dinamico	Imposta lo stato di sola lettura delle transazioni attuali.
transform_null_equals	Dinamico	Tratta expr=NULL come expr È NULL.
update_process_title	Dinamico	Aggiorna il titolo del processo per mostrare il comando SQL attivo.
vacuum_cost_delay	Dinamico	Ritardo del costo del vacuum, in millisecondi.
vacuum_cost_limit	Dinamico	Quantità del costo del vacuum disponibile prima del napping.
vacuum_cost_page_dirty	Dinamico	Costo del vacuum per una pagina sporcata dal vacuum.
vacuum_cost_page_hit	Dinamico	Costo del vacuum per una pagina trovata nella cache del buffer.
vacuum_cost_page_miss	Dinamico	Costo del vacuum per una non pagina trovata nella cache del buffer.
vacuum_defer_cleanup_age	Dinamico	Numero di transazioni in base alle quali il vacuum e la pulizia a caldo devono essere posticipati, se presenti.

Nome del parametro	Apply_Type	Descrizione
vacuum_freeze_min_age	Dinamico	Età minima in cui il vacuum dovrebbe congelare una riga della tabella.
vacuum_freeze_table_age	Dinamico	Età in cui il vacuum dovrebbe eseguire la scansione di un'intera tabella per congelare le tuple.
wal_buffers	Statico	Imposta il numero di buffer della pagina del disco nella memoria condivisa per WAL.
wal_writer_delay	Dinamico	Tempo di inattività della scrittura WAL tra i flush di WAL.
work_mem	Dinamico	Imposta la memoria massima da utilizzare per gli spazi di lavoro delle query.
xmlbinary	Dinamico	Imposta come i valori binari devono essere codificati in XML.
xmloption	Dinamico	Imposta se i dati XML nelle operazioni di analisi e serializzazione implicite devono essere considerati come documenti o frammenti di contenuto.

Amazon RDS usa le unità PostgreSQL predefinite per tutti i parametri. La tabella seguente mostra l'unità predefinita di PostgreSQL per ogni parametro.

Nome del parametro	Unità
archive_timeout	s
authentication_timeout	s
autovacuum_naptime	s
autovacuum_vacuum_cost_delay	ms

Nome del parametro	Unità
bgwriter_delay	ms
checkpoint_timeout	s
checkpoint_warning	s
deadlock_timeout	ms
effective_cache_size	8 KB
lock_timeout	ms
log_autovacuum_min_duration	ms
log_min_duration_statement	ms
log_rotation_age	minuti
log_rotation_size	KB
log_temp_files	KB
maintenance_work_mem	KB
max_stack_depth	KB
max_standby_archive_delay	ms
max_standby_streaming_delay	ms
post_auth_delay	s
pre_auth_delay	s
segment_size	8 KB
shared_buffers	8 KB
statement_timeout	ms

Nome del parametro	Unità
ssl_renegotiation_limit	KB
tcp_keepalives_idle	s
tcp_keepalives_interval	s
temp_file_limit	KB
work_mem	KB
temp_buffers	8 KB
vacuum_cost_delay	ms
wal_buffers	8 KB
wal_receiver_timeout	ms
wal_segment_size	B
wal_sender_timeout	ms
wal_writer_delay	ms
wal_receiver_status_interval	s

Ottimizzazione degli eventi di attesa per RDS per PostgreSQL

Gli eventi di attesa sono un importante strumento di ottimizzazione per RDS per PostgreSQL. Quando si scopre perché le sessioni sono in attesa di risorse e l'azione svolta, è possibile risolvere i colli di bottiglia in maniera più efficiente. È possibile utilizzare le informazioni contenute in questa sezione per trovare possibili cause e azioni correttive. Questa sezione illustra anche i concetti di base sull'ottimizzazione di PostgreSQL.

Gli eventi di attesa in questa sezione sono specifici di RDS per PostgreSQL.

Argomenti

Concetti essenziali per l'ottimizzazione di RDS per PostgreSQL

Prima di ottimizzare il database RDS per PostgreSQL, assicurati di sapere cosa sono gli eventi di attesa e perché si verificano. Esamina anche l'architettura di base della memoria e del disco di RDS per PostgreSQL. Per un utile diagramma architettonico, vedere il wikibook PostgreSQL.

Argomenti

Eventi di attesa di RDS per PostgreSQL

Un evento di attesa indica che una sessione è in attesa di una risorsa. Ad esempio, l'evento di attesa `Client:ClientRead` si verifica quando RDS per PostgreSQL è in attesa di ricevere dati dal client. Le sessioni in genere attendono risorse come le seguenti.

- Accesso a thread singolo a un buffer, ad esempio, quando una sessione sta tentando di modificare un buffer
- Una riga attualmente bloccata da un'altra sessione
- Un file di dati letto
- Scrittura di un file log

Ad esempio, per soddisfare una query, la sessione potrebbe eseguire una scansione completa della tabella. Se i dati non sono già in memoria, la sessione attende il completamento dell'I/O del disco. Quando i buffer vengono letti in memoria, potrebbe essere necessario attendere perché altre sessioni accedono agli stessi buffer. Il database registra le attese utilizzando un evento di attesa predefinito. Questi eventi sono raggruppati in categorie.

Di per sé, un singolo evento di attesa non indica un problema di prestazioni. Ad esempio, se i dati richiesti non sono in memoria, è necessaria la lettura dei dati dal disco. Se una sessione blocca una riga per un aggiornamento, un'altra sessione attende che la riga venga sbloccata in modo che possa aggiornarla. Un commit richiede di attendere il completamento della scrittura su un file di registro. Le attese sono parte integrante del normale funzionamento di un database.

Un gran numero di eventi di attesa in genere mostrano un problema di prestazioni. In questi casi, è possibile utilizzare i dati degli eventi di attesa per determinare dove stanno trascorrendo il tempo delle

sessioni. Ad esempio, se un report che in genere viene eseguito in minuti ora viene eseguito per ore, è possibile identificare gli eventi di attesa che contribuiscono maggiormente al tempo di attesa totale. Se è possibile determinare le cause degli eventi di attesa principali, a volte è possibile apportare modifiche che migliorano le prestazioni. Ad esempio, se la sessione è in attesa su una riga bloccata da un'altra sessione, è possibile terminare la sessione di blocco.

Memoria RDS per PostgreSQL

La memoria RDS per PostgreSQL è divisa in condivisa e locale.

Argomenti

- Memoria condivisa in RDS per PostgreSQL
- Memoria locale in RDS per PostgreSQL

Memoria condivisa in RDS per PostgreSQL

RDS per PostgreSQL assegna memoria condivisa all'avvio dell'istanza. La memoria condivisa è divisa in più sottoaree. Di seguito è possibile trovare una descrizione dei più importanti.

Argomenti

- Buffer condivisi
- Buffer Write ahead log (WAL)

Buffer condivisi

Il pool buffer condiviso è un'area di memoria RDS per PostgreSQL che contiene tutte le pagine che sono o sono state utilizzate dalle connessioni delle applicazioni. Una pagina è la versione di memoria di un blocco disco. Il buffer pool condiviso memorizza nella cache i blocchi di dati letti dal disco. Il pool riduce la necessità di rileggere i dati dal disco, rendendo il database più efficiente.

Ogni tabella e indice vengono memorizzati come una matrice di pagine di dimensioni fisse. Ogni blocco contiene più tuple, che corrispondono alle righe. Una tupla può essere memorizzata in qualsiasi pagina.

Il buffer pool condiviso ha memoria finita. Se una nuova richiesta richiede una pagina che non è in memoria, e non esiste più memoria, RDS per PostgreSQL espelle una pagina utilizzata meno frequentemente per soddisfare la richiesta. La politica di sfratto è implementata da un algoritmo di sweep dell'orologio.

Il parametro `shared_buffers` determina la quantità di memoria che il server dedica alla memorizzazione nella cache dei dati.

Buffer Write ahead log (WAL)

Un Buffer write-ahead log (WAL) conserva i dati delle transazioni che RDS per PostgreSQL successivamente scrive sull'archiviazione persistente. Utilizzando il meccanismo WAL, RDS per PostgreSQL può effettuare le seguenti operazioni:

- Recuperare i dati dopo un errore
- Ridurre l'I/O del disco evitando scritture frequenti su disco

Quando un client modifica i dati, RDS per PostgreSQL scrive le modifiche nel buffer WAL. Quando il client emette un `COMMIT`, il processo di scrittura WAL scrive i dati delle transazioni nel file WAL.

Il parametro `wal_level` determina la quantità di informazioni scritte sul WAL.

Memoria locale in RDS per PostgreSQL

Ogni processo di back-end assegna memoria locale per l'elaborazione delle query.

Argomenti

- Area di memoria di lavoro
- Area memoria di lavoro di manutenzione
- Area buffer temporanea

Area di memoria di lavoro

L'area di memoria di lavoro contiene dati temporanei per query che eseguono ordinamenti e hash. Ad esempio, una query con clausola `ORDER BY` esegue un ordinamento. Le query utilizzano tabelle hash nei join e nelle aggregazioni hash.

Il parametro `work_mem` indica la quantità di memoria da utilizzare dalle operazioni di ordinamento interno e dalle tabelle hash prima di scrivere su file di disco temporanei. Il valore predefinito è 4 MB. È possibile eseguire più sessioni contemporaneamente e ogni sessione può eseguire operazioni di manutenzione in parallelo. Per questo motivo, la memoria di lavoro totale utilizzata può essere costituita da multipli dell'impostazione `work_mem`.

Area memoria di lavoro di manutenzione

L'area di memoria di lavoro di manutenzione memorizza nella cache i dati per le operazioni di manutenzione. Queste operazioni includono l'aspirazione, la creazione di un indice e l'aggiunta di chiavi esterne.

Il parametro `maintenance_work_mem` specifica la quantità massima di memoria da utilizzare nelle operazioni di manutenzione. Il valore predefinito è 64 MB. Una sessione di database può eseguire solo un'operazione di manutenzione alla volta.

Area buffer temporanea

L'area buffer temporanea memorizza nella cache le tabelle temporanee per ciascuna sessione del database.

Ogni sessione assegna buffer temporanei secondo necessità fino al limite specificato. Quando la sessione scade, il server cancella i buffer.

Il parametro `temp_buffers` imposta il numero massimo di buffer temporanei utilizzati da ogni sessione. Prima del primo utilizzo di tabelle temporanee all'interno di una sessione, è possibile modificare il valore `temp_buffers`.

Processi di RDS per PostgreSQL

RDS per PostgreSQL utilizza più processi.

Argomenti

Processo postmaster

Il processo postmaster è il primo processo eseguito quando si avvia RDS per PostgreSQL. Il processo postmaster ha le seguenti responsabilità principali:

- Forcella e monitoraggio dei processi in background
- Ricevere le richieste di autenticazione dai processi client e autenticarle prima di consentire al database di servire le richieste

Processi di back-end

Se il postmaster autentica una richiesta del cliente, il postmaster forcherà un nuovo processo di back-end, chiamato anche processo postgres. Un processo client si connette esattamente a un processo back-end. Il processo client e il processo di backend comunicano direttamente senza intervento da parte del processo postmaster.

Processi in background

Il processo postmaster forca diversi processi che eseguono diverse attività di back-end. Alcuni dei più importanti includono quanto segue:

- Scrittore WAL

  RDS per PostgreSQL scrive i dati nel buffer WAL (write ahead logging) nei file di log. Il principio della registrazione write ahead è che il database non può scrivere modifiche ai file di dati fino a quando il database ha scritto i record di log che descrivono tali modifiche su disco. Il meccanismo WAL riduce l'I/O del disco e consente a RDS per PostgreSQL di utilizzare i log per recuperare il database dopo un errore.
- Background writer

  Questo processo scrive periodicamente pagine sporche (modificate) dai buffer di memoria ai file di dati. Una pagina diventa sporca quando un processo di back-end la modifica in memoria.
- Il daemon dell'Autovacuum

  Il daemon include i seguenti elementi:

  - Il lanciatore di autovacuum
  - I processi di autovacuum worker

  Se abilitata, verifica la presenza di tabelle con un numero elevato di tuple inserite, aggiornate o eliminate. Il daemon ha le seguenti responsabilità:

  - Recuperare o riutilizzare lo spazio su disco occupato da righe aggiornate o eliminate
  - Aggiornare le statistiche utilizzate dal planner
  - Proteggere contro la perdita di dati precedenti a causa di un involucro dell'ID transazione

  La funzione autovacuum automatizza l'esecuzione dei comandi `VACUUM` e `ANALYZE`. `VACUUM` ha le seguenti varianti: standard e full. Il vuoto standard funziona in parallelo con altre operazioni di database. `VACUUM FULL` richiede un blocco esclusivo sulla tabella su cui sta lavorando. Pertanto,

non può essere eseguito in parallelo con le operazioni che accedono alla stessa tabella. `VACUUM` crea una notevole quantità di traffico I/O, che può causare prestazioni scadenti per altre sessioni attive.

Eventi di attesa di RDS per PostgreSQL

La seguente tabella elenca gli eventi di attesa per RDS per PostgreSQL che indicano i problemi di prestazioni con le cause più comuni e le azioni correttive.

Evento di attesa	Definizione
Client:ClientRead	Ad esempio, l'evento di attesa si verifica quando RDS per PostgreSQL è in attesa di ricevere dati dal client.
Client:ClientWrite	Ad esempio, l'evento di attesa si verifica quando RDS per PostgreSQL è in attesa di ricevere dati dal client.
CPU	Questo evento si verifica quando un thread è attivo nella CPU o è in attesa della CPU.
IO:buffileRead e IO:buffileWrite	Questi eventi si verificano quando RDS per PostgreSQL crea file temporanei.
IO: DataFileRead	Questo evento si verifica quando una connessione attende un processo di back-end per leggere una pagina richiesta dalla memoria perché la pagina non è disponibile nella memoria condivisa.
IO:WALWrite	Questo evento si verifica quando RDS per PostgreSQL è in attesa che i buffer WAL (write-ah ead log) vengano scritti in un file WAL.
Lock:advisory	Questo evento si verifica quando un'applicazione PostgreSQL utilizza un blocco per coordinare l'attività su più sessioni.

Evento di attesa	Definizione
Lock:extend	Questo evento si verifica quando un processo di back-end è in attesa di bloccare una relazione per estenderla mentre un altro processo ha un blocco su tale relazione per lo stesso scopo.
Lock:Relation	Questo evento si verifica quando una query è in attesa di acquisire un blocco su una tabella o vista attualmente bloccata da un'altra transazione.
Lock:transactionid	Questo evento si verifica quando una transazione è in attesa di un blocco a livello di riga.
Lock:tuple	Questo evento si verifica quando un processo di backend aspetta di acquisire un blocco su una tupla.
LWLock:BufferMapping (LWLock:buffer_mapping)	Questo evento si verifica quando un processo di backend è in attesa di associare un blocco di dati a un buffer nel pool di buffer condiviso.
LWLock:BufferIO (IPC:BufferIO)	Questo evento si verifica quando RDS per PostgreSQL è in attesa che altri processi finiscano le operazioni di input/output (I/O) quando si tenta contemporaneamente di accedere a una pagina.
LWLock:buffer_content (BufferContent)	Questo evento si verifica quando una sessione è in attesa di accedere in lettura o scrittura a una pagina dati in memoria mentre un'altra sessione ha bloccato la pagina in scrittura.
LWLock:lock_manager (LWLock:lockmanager)	Questo evento si verifica quando il motore RDS per PostgreSQL mantiene l'area di memoria del blocco condiviso per allocare, controllare e deallocare un blocco quando non è possibile un blocco rapido del percorso.

Evento di attesa	Definizione
Timeout: PG Sleep	Questo evento si verifica quando un processo server ha chiamato la funzione `pg_sleep` e sta aspettando la scadenza del timeout del sonno.
Timeout:VacuumDelay	Questo evento indica che il processo vacuum è inattivo perché è stato raggiunto il limite di costo stimato.

Client:ClientRead

L'evento `Client:ClientRead` si verifica quando RDS per PostgreSQL è in attesa di ricevere dati dal client.

Argomenti

- Versioni del motore supportate
- Context
- Probabili cause di aumento delle attese
- Operazioni

Versioni del motore supportate

Queste informazioni relative all'evento di attesa sono supportate per RDS per PostgreSQL versione 10 e successive.

Context

Un'istanza database RDS per PostgreSQL è in attesa di ricevere dati dal client. L'istanza database RDS per PostgreSQL deve ricevere i dati dal client prima di poter inviare più dati al client. Il tempo di attesa dell'istanza prima di ricevere i dati dal client è un evento `Client:ClientRead`.

Probabili cause di aumento delle attese

Le cause comuni della comparsa dell'evento `Client:ClientRead` che appare nelle prime attese includono:

Maggiore latenza di rete

Potrebbe esserci una maggiore latenza di rete tra l'istanza database RDS per PostgreSQL e il client. Una maggiore latenza di rete aumenta il tempo necessario per la ricezione dei dati dal client dell'istanza database.

Aumento del carico sul client

Potrebbe esserci una pressione della CPU o una saturazione della rete sul client. Un aumento del carico sul client può ritardare la trasmissione dei dati dal client all'istanza RDS per PostgreSQL.

Eccesso di viaggi di andata e ritorno in rete

Un numero elevato di round trip di rete tra l'istanza database RDS per PostgreSQL e il client può ritardare la trasmissione dei dati dal client all'istanza database RDS per PostgreSQL.

Operazione copia di grandi dimensioni

Durante un'operazione di copia, i dati vengono trasferiti dal file system del client all'istanza database RDS per PostgreSQL. L'invio di una grande quantità di dati all'istanza database può ritardare la trasmissione dei dati dal client all'istanza database.

Connessione client inattiva

Quando un client si connette all'istanza database RDS per PostgreSQL nello stato `idle in transaction`, l'istanza database potrebbe attendere che il client invii più dati o emetta un comando. Una connessione in questo stato può portare ad un aumento degli eventi `Client:ClientRead`.

PGBouncer utilizzato per il connection pooling

PGBouncer ha un'impostazione di configurazione di rete di basso livello denominata `pkt_buf`, che è impostata su 4.096 per impostazione predefinita. Se il carico di lavoro invia pacchetti di query superiori a 4.096 byte tramite PGBouncer, consigliamo di aumentare l'impostazione `pkt_buf` a 8.192. Se la nuova impostazione non diminuisce il numero di eventi `Client:ClientRead`, consigliamo di aumentare l'impostazione `pkt_buf` su valori più grandi, come 16.384 o 32.768. Se il testo della query è grande, l'impostazione più grande può essere particolarmente utile.

Operazioni

Consigliamo azioni diverse a seconda delle cause dell'evento di attesa.

Argomenti

Posizionamento dei client nella stessa zona di disponibilità e sottorete VPC dell'istanza

Per ridurre la latenza di rete e aumentare la velocità di trasmissione effettiva di rete, posiziona i client nella stessa sottorete della zona di disponibilità e cloud privato virtuale (VPC) dell'istanza database RDS per PostgreSQL. Assicurati che i client siano geograficamente il più vicini possibile all'istanza database.

Ridimensionare il client

Utilizzando Amazon CloudWatch o altri parametri host, stabilire se il proprio client è attualmente vincolato dalla CPU o dalla larghezza di banda di rete o entrambi. Se il client è vincolato, ridimensionare il client di conseguenza.

Utilizza istanze di generazione corrente

In alcuni casi, potresti non utilizzare una classe di istanza DB che supporta i frame jumbo. Se stai eseguendo l'applicazione su Amazon EC2, considera l'utilizzo di un'istanza di generazione corrente per il client. Inoltre, configura l'unità di trasmissione massima (MTU) sul sistema operativo client. Questa tecnica potrebbe ridurre il numero di round trip di rete e aumentare il throughput di rete. Per ulteriori informazioni, consulta Jumbo frames (9001 MTU) nella Guida per l'utente di Amazon EC2 per le istanze Linux.

Per informazioni sulle classi di istanza database, consulta Classi di istanze database. Per determinare la classe di istanza DB equivalente a un tipo di istanza Amazon EC2, posizionare `db.` prima del nome del tipo di istanza Amazon EC2. Ad esempio, l'istanza Amazon EC2 `r5.8xlarge` è equivalente alla classe di istanza DB `db.r5.8xlarge`.

Aumentare la larghezza di banda di rete

Utilizza i parametri Amazon CloudWatch `NetworkReceiveThroughput` e `NetworkTransmitThroughput` per monitorare il traffico di rete in entrata e in uscita sull'istanza

database. Questi parametri possono aiutarti a determinare se la larghezza di banda della rete è sufficiente per il tuo carico di lavoro.

Se la larghezza di banda della rete non è sufficiente, aumentala. Se il file client AWS o l'istanza DB sta raggiungendo i limiti di larghezza di banda di rete, l'unico modo per aumentare la larghezza di banda è aumentare la dimensione dell'istanza DB. Per ulteriori informazioni, consulta Tipi di classi di istanza database.

Per ulteriori informazioni sui parametri di CloudWatch, consultare CloudWatch Parametri Amazon per Amazon RDS.

Monitora il massimo delle prestazioni di rete

Se utilizzi client Amazon EC2, Amazon EC2 fornisce il massimo per i parametri delle prestazioni di rete, inclusa la larghezza di banda aggregata in entrata e in uscita. Fornisce inoltre il monitoraggio della connessione per garantire che i pacchetti vengano restituiti come previsto e l'accesso ai servizi locali per servizi come il DNS (Domain Name System). Per monitorare questi massimi, utilizza un driver di rete avanzato e monitora le prestazioni di rete per il client.

Per ulteriori informazioni, consulta Monitoraggio delle prestazioni di rete per l'istanza Amazon EC2 nella Guida per l'utente di Amazon EC2 User Guide per le istanze Linux e Monitoraggio delle prestazioni di rete per l'istanza Amazon EC2 nella Guida per l'utente di Amazon EC2 per le istanze Windows.

Monitorare le transazioni nello stato «inattivo nella transazione»

Controlla se hai un numero crescente di connessioni `idle in transaction`. Per fare ciò, monitora la colonna `state` nella tabella `pg_stat_activity`. Potrebbe essere possibile identificare l'origine della connessione eseguendo una query simile alla seguente.

```
select client_addr, state, count(1) from pg_stat_activity
where state like 'idle in transaction%'
group by 1,2
order by 3 desc
```

Client:ClientWrite

L'evento `Client:ClientWrite` si verifica quando RDS per PostgreSQL è in attesa di scrivere dati sul client.

Argomenti

Versioni del motore supportate

Queste informazioni relative all'evento di attesa sono supportate per RDS per PostgreSQL versione 10 e successive.

Context

Il processo client deve ricevere i dati da un cluster database RDS per PostgreSQL prima di poter inviare più dati. Il tempo in cui il cluster attende prima inviare altri dati al client è un evento `Client:ClientWrite`.

La velocità di trasmissione effettiva di rete ridotto tra l'istanza database RDS per PostgreSQL e il client può causare questo evento. Anche la pressione della CPU e la saturazione della rete sul client possono causare questo evento. Pressione CPU è quando la CPU è completamente utilizzata e ci sono attività in attesa del tempo della CPU. Saturazione rete è quando la rete tra il database e il client trasporta più dati di quelli che è in grado di gestire.

Probabili cause di aumento delle attese

Le cause comuni della comparsa dell'evento `Client:ClientWrite` che appare nelle prime attese includono:

Maggiore latenza di rete

Potrebbe esserci una maggiore latenza di rete tra l'istanza database RDS per PostgreSQL e il client. Una maggiore latenza di rete aumenta il tempo necessario per la ricezione dei dati dal client.

Aumento del carico sul client

Potrebbe esserci una pressione della CPU o una saturazione della rete sul client. Un aumento del carico sul client ritarda la ricezione dei dati dall'istanza database RDS per PostgreSQL.

Ampio volume di dati inviati al client

L'istanza database RDS per PostgreSQL DB potrebbe inviare una grande quantità di dati al client. Un client potrebbe non essere in grado di ricevere i dati con la stessa rapidità dell'invio del cluster. Attività come una copia di una tabella di grandi dimensioni possono comportare un aumento degli eventi `Client:ClientWrite`.

Operazioni

Consigliamo azioni diverse a seconda delle cause dell'evento di attesa.

Argomenti

- Posizionare i client nella stessa area di disponibilità e subnet VPC del cluster
- Utilizza istanze di generazione corrente
- Ridurre la quantità di dati inviati al client
- Ridimensionare il client

Posizionare i client nella stessa area di disponibilità e subnet VPC del cluster

Per ridurre la latenza di rete e aumentare la velocità di trasmissione effettiva di rete, posiziona i client nella stessa sottorete della zona di disponibilità e cloud privato virtuale (VPC) dell'istanza database RDS per PostgreSQL.

Utilizza istanze di generazione corrente

In alcuni casi, potresti non utilizzare una classe di istanza DB che supporta i frame jumbo. Se stai eseguendo l'applicazione su Amazon EC2, considera l'utilizzo di un'istanza di generazione corrente per il client. Inoltre, configura l'unità di trasmissione massima (MTU) sul sistema operativo client. Questa tecnica potrebbe ridurre il numero di round trip di rete e aumentare il throughput di rete. Per ulteriori informazioni, consulta Jumbo frames (9001 MTU) nella Guida per l'utente di Amazon EC2 per le istanze Linux.

Per informazioni sulle classi di istanza database, consulta Classi di istanze database. Per determinare la classe di istanza DB equivalente a un tipo di istanza Amazon EC2, posizionare `db.` prima del nome del tipo di istanza Amazon EC2. Ad esempio, l'istanza Amazon EC2 `r5.8xlarge` è equivalente alla classe di istanza DB `db.r5.8xlarge`.

Ridurre la quantità di dati inviati al client

Quando possibile, regola l'applicazione per ridurre la quantità di dati che l'istanza database RDS per PostgreSQL invia al client. Effettuare tali regolazioni allevia la contesa della CPU e della rete sul client.

Ridimensionare il client

Utilizzando Amazon CloudWatch o altri parametri host, stabilire se il proprio client è attualmente vincolato dalla CPU o dalla larghezza di banda di rete o entrambi. Se il client è vincolato, ridimensionare il client di conseguenza.

CPU

Questo evento si verifica quando un thread è attivo nella CPU o è in attesa della CPU.

Argomenti

Versioni del motore supportate

Queste informazioni relative all'evento di attesa sono pertinenti per tutte le versioni di RDS per PostgreSQL.

Context

L’unità di elaborazione centrale (CPU) è il componente di un computer che esegue le istruzioni. Ad esempio, le istruzioni della CPU eseguono operazioni aritmetiche e scambiano dati in memoria. Se una query aumenta il numero di istruzioni eseguite tramite il motore di database, il tempo impiegato per eseguire la query aumenta. Pianificazione CPU sta dando tempo alla CPU a un processo. La pianificazione viene orchestrata dal kernel del sistema operativo.

Argomenti

- Parametri di utilizzo di OS.CPU
- Probabile causa della pianificazione della CPU

Come sapere quando si verifica questa attesa

Questo evento di attesa CPU indica che un processo di backend è attivo nella CPU o è in attesa della CPU. Si verifica quando una query mostra le seguenti informazioni:

- La colonna `pg_stat_activity.state` ha il valore `active`.
- Le colonne `wait_event_type` e `wait_event` in `pg_stat_activity` sono entrambe `null`.

Per visualizzare i processi di backend in uso o in attesa sulla CPU, eseguire la seguente query.

```
SELECT *
FROM   pg_stat_activity
WHERE  state = 'active'
AND    wait_event_type IS NULL
AND    wait_event IS NULL;
```

Parametro DBLOADCPU

Il parametro Performance Insights per la CPU è `DBLoadCPU`. Il valore di `DBLoadCPU` può differire dal valore del parametro Amazon CloudWatch `CPUUtilization`. Quest'ultimo parametro viene raccolto dall'HyperVisor per un'istanza di database.

Parametri di utilizzo di OS.CPU

I parametri del sistema operativo Performance Insights forniscono informazioni dettagliate sull'utilizzo della CPU. Ad esempio, è possibile visualizzare i seguenti parametri:

- `os.cpuUtilization.nice.avg`
- `os.cpuUtilization.total.avg`
- `os.cpuUtilization.wait.avg`
- `os.cpuUtilization.idle.avg`

Performance Insights segnala l'utilizzo della CPU da parte del motore di database come `os.cpuUtilization.nice.avg`.

Probabile causa della pianificazione della CPU

Il kernel del sistema operativo (OS) gestisce la pianificazione per la CPU. Quando la CPU è attiva, potrebbe essere necessario attendere la pianificazione di un processo. La CPU è attiva durante l'esecuzione dei calcoli. È attiva anche se ha un thread inattivo che non è in esecuzione, ovvero un thread inattivo che attende l'I/O della memoria. Questo tipo di I/O domina il tipico carico di lavoro del database.

È probabile che i processi attendano di essere pianificati su una CPU quando vengono soddisfatte le seguenti condizioni:

- Il parametro CloudWatch `CPUUtilization` è vicino al 100 percento.
- Il carico medio è superiore al numero di vCPU, indicando un carico pesante. Puoi trovare il parametro `loadAverageMinute` nella sezione parametri del sistema operativo in Performance Insights.

Probabili cause di aumento delle attese

Quando l'evento di attesa della CPU si verifica più del normale, probabilmente indicando un problema di prestazioni, le cause tipiche includono le seguenti.

Argomenti

Probabili cause di picchi improvvisi

Le cause più probabili di picchi improvvisi sono le seguenti:

- L'applicazione ha aperto troppe connessioni simultanee al database. Questo scenario è noto come «tempesta di connessione».
- Il carico di lavoro dell'applicazione è cambiato in uno dei seguenti modi:
  - Nuove query
  - Aumento delle dimensioni del set di dati
  - Manutenzione o creazione dell'indice
  - Nuove funzioni

  - Nuovi operatori
  - Un aumento dell'esecuzione parallela di query
- I piani di esecuzione delle query sono cambiati. In alcuni casi, un cambiamento può causare un aumento dei buffer. Ad esempio, la query ora utilizza una scansione sequenziale quando in precedenza utilizzava un indice. In questo caso, le query richiedono più CPU per raggiungere lo stesso obiettivo.

Probabili cause di alta frequenza a lungo termine

Le cause più probabili di eventi che si ripetono per un lungo periodo:

- Troppi processi backend sono in esecuzione contemporaneamente sulla CPU. Questi processi possono essere lavoratori paralleli.
- Le query vengono eseguite in modo subottimale perché necessitano di un numero elevato di buffer.

Casi particolari

Se nessuna delle cause probabili risulta essere una causa effettiva, potrebbero verificarsi le seguenti situazioni:

- La CPU sta scambiando i processi in entrata e in uscita.
- La CPU potrebbe gestire le voci della tabella della pagina se la funzionalità delle pagine di grandi dimensioni è stata disattivata. La funzionalità di gestione della memoria è attivata per impostazione predefinita per tutte le classi di istanza database diverse dalle classi di istanza micro, piccole e medie. Per ulteriori informazioni, consulta Pagine di grandi dimensioni per RDS for PostgreSQL .

Operazioni

Se l'evento di attesa CPU domina l'attività del database, non indica necessariamente un problema di prestazioni. Rispondi a questo evento solo quando le prestazioni diminuiscono.

Argomenti

- Indagare se il database sta causando l'aumento della CPU
- Determina se il numero di connessioni è aumentato
- Rispondere alle modifiche del carico di lavoro

Indagare se il database sta causando l'aumento della CPU

Esamina il parametro `os.cpuUtilization.nice.avg` in Performance Insights. Se questo valore è molto inferiore all'utilizzo della CPU, i processi nondatabase sono il principale contributore della CPU.

Determina se il numero di connessioni è aumentato

Esamina il parametro `DatabaseConnections` in Amazon CloudWatch. L'azione dipende dal fatto che il numero sia aumentato o diminuito durante il periodo di aumento degli eventi di attesa della CPU.

Le connessioni sono aumentate

Se il numero di connessioni è aumentato, confrontare il numero di processi back-end che consumano la vCPU con il numero di vCPU. Gli scenari possibili sono i seguenti:

- Il numero di processi backend che consumano vCPU è inferiore al numero di vCPU.

  In questo caso, il numero di connessioni non è un problema. Tuttavia, è comunque possibile provare a ridurre l'utilizzo della CPU.
- Il numero di processi backend che consumano vCPUs è inferiore al numero di vCPU.

  In questo caso, valuta le seguenti opzioni:
  - Riduci il numero di processi back-end collegati al database. Ad esempio, implementa una soluzione di connection pooling come RDS Proxy. Per ulteriori informazioni, consulta Utilizzo di Server proxy per Amazon RDS.
  - Aggiorna le dimensioni dell'istanza per ottenere un numero maggiore di vCPUs.
  - Reindirizza alcuni carichi di lavoro di sola lettura ai nodi del lettore, se applicabile.

Le connessioni sono aumentate

Esamina il parametro `blks_hit` in Performance Insights. Cerca una correlazione tra un aumento `blks_hit` e utilizzo CPU. Gli scenari possibili sono i seguenti:

- Utilizzo CPU e `blks_hit` sono correlati.

  In questo caso, trova le istruzioni SQL principali collegate all'utilizzo della CPU e cerca le modifiche al piano. Puoi utilizzare una delle seguenti tecniche:
  - Spiegare i piani manualmente e confrontarli con il piano di esecuzione previsto.

- Cercare un aumento degli hit di blocco al secondo e dei blocchi locali al secondo. Nella sezione Top SQL della dashboard Performance Insights, scegli Preferenze.

- Utilizzo CPU e `blks_hit` non sono correlati.

  In questo caso, determinare se si verifica una delle seguenti condizioni:

  - L'applicazione si sta rapidamente connettendo e disconnettendo dal database.

    Diagnosticare questo comportamento attivando `log_connections` e `log_disconnections`, quindi analizzando i registri PostgreSQL. Considerare l'utilizzo dell'analizzatore di log `pgbadger`. Per ulteriori informazioni, consulta https://github.com/darold/pgbadger.

  - Il sistema operativo è sovraccarico.

    In questo caso, Performance Insights mostra che i processi back-end consumano la CPU per un tempo più lungo del solito. Cerca prove nei parametri `os.cpuUtilization` di Performance Insights o nel parametro CloudWatch `CPUUtilization`. Se il sistema operativo è sovraccarico, esamina i parametri di Enhanced Monitoring per effettuare ulteriori diagnosi. In particolare, guarda l'elenco dei processi e la percentuale di CPU consumata da ciascun processo.

  - Le istruzioni SQL principali consumano troppa CPU.

    Esaminare le istruzioni collegate all'utilizzo della CPU per verificare se possono utilizzare meno CPU. Esegui il comando `EXPLAIN` e concentrati sui nodi del piano che hanno il maggior impatto. Prendi in considerazione l'utilizzo di un visualizzatore del piano di esecuzione PostgreSQL. Per provare questo strumento, vedi http://explain.dalibo.com/.

Rispondere alle modifiche del carico di lavoro

Se il carico di lavoro è cambiato, cerca i seguenti tipi di modifiche:

Nuove query

Verifica se sono previste le nuove query. In tal caso, assicurarsi che siano previsti i piani di esecuzione e il numero di esecuzioni al secondo.

Aumento delle dimensioni del set di dati

Determina se il partizionamento, se non è già implementato, potrebbe essere d'aiuto. Questa strategia potrebbe ridurre il numero di pagine che una query deve recuperare.

Manutenzione o creazione dell'indice

Verificare se è previsto il programma per la manutenzione. Una best practice è pianificare le attività di manutenzione al di fuori delle attività di picco.

Nuove funzioni

Verifica se queste funzioni funzionano come previsto durante il test. In particolare, controlla se è previsto il numero di esecuzioni al secondo.

Nuovi operatori

Verifica se queste funzioni funzionano come previsto durante il test.

Un aumento dell'esecuzione di query parallele

Determina se si è verificata una delle seguenti situazioni:

- Le relazioni o gli indici coinvolti sono improvvisamente cresciuti di dimensioni in modo che differiscano in modo significativo da `min_parallel_table_scan_size` o `min_parallel_index_scan_size`.
- Cambiamenti recenti sono stati apportati a `parallel_setup_cost` o `parallel_tuple_cost`.
- Cambiamenti recenti sono stati apportati a `max_parallel_workers` o `max_parallel_workers_per_gather`.

IO:buffileRead e IO:buffileWrite

Gli eventi `IO:BufFileRead` e `IO:BufFileWrite` si verificano quando RDS per PostgreSQL crea file temporanei. Quando le operazioni richiedono più memoria rispetto ai parametri della memoria di lavoro attualmente definiti, scrivono dati temporanei sullo storage persistente. Questa operazione è talvolta chiamata «versamento su disco».

Argomenti

- Versioni del motore supportate
- Context
- Probabili cause di aumento delle attese
- Operazioni

Versioni del motore supportate

Queste informazioni relative all'evento di attesa sono supportate per tutte le versioni di RDS per PostgreSQL.

Context

`IO:BufFileRead` e `IO:BufFileWrite` riguardano l'area di memoria di lavoro e l'area di memoria di lavoro di manutenzione. Per ulteriori informazioni su queste aree di memoria locale, consulta Resource Consumption (Consumo di risorse) nella documentazione di PostgreSQL.

Il valore predefinito per `work_mem` è 4 MB. Se una sessione esegue operazioni in parallelo, ogni lavoratore che gestisce il parallelismo utilizza 4 MB di memoria. Per questo motivo, imposta `work_mem` con attenzione. Se si aumenta troppo il valore, un database che esegue molte sessioni potrebbe consumare troppa memoria. Se si imposta il valore troppo basso, RDS per PostgreSQL crea file temporanei nella memoria locale. L'I/O del disco per questi file temporanei può ridurre le prestazioni.

Se si osserva la seguente sequenza di eventi, il database potrebbe generare file temporanei:

1. Riduzione improvvisa e brusca della disponibilità
2. Ripristino rapido per lo spazio libero

Potresti vedere anche un motivo a «motosega». Questo modello può indicare che il database sta creando costantemente file di piccole dimensioni.

Probabili cause di aumento delle attese

In generale, questi eventi di attesa sono causati da operazioni che consumano più memoria rispetto a quella allocata dai parametri `work_mem` o `maintenance_work_mem`. Per compensare, le operazioni scrivono su file temporanei. Cause comuni degli eventi `IO:BufFileRead` e `IO:BufFileWrite` includono quanto segue:

Query che richiedono più memoria di quella esistente nell'area della memoria di lavoro

Le query con le seguenti caratteristiche utilizzano l'area di memoria di lavoro:

- Hash join
- `ORDER BY` Clausola

- `GROUP BY`Clausola
- `DISTINCT`
- Funzioni finestra
- `CREATE TABLE AS SELECT`
- Aggiornamento vista materializzata

Istruzioni che richiedono più memoria di quella esistente nell'area della memoria di lavoro di manutenzione

Le istruzioni seguenti utilizzano l'area di memoria di lavoro di manutenzione:

- `CREATE INDEX`
- `CLUSTER`

Operazioni

Consigliamo azioni diverse a seconda delle cause dell'evento di attesa.

Argomenti

- Identificare il problema
- Esamina le tue query di join
- Esamina le query ORDER BY e GROUP BY
- Evitare di utilizzare l'operazione DISTINCT
- Considera l'utilizzo di funzioni finestra anziché le funzioni GROUP BY
- Indagare sulle viste materializzate e le istruzioni CTAS
- Uso di pg_repack per ricostruire gli indici
- Aumenta maintenance_work_mem quando esegui cluster
- Sintonizza la memoria per impedire IO:BuffileRead e IO:BuffileWrite

Identificare il problema

Supponiamo una situazione in cui Performance Insights non è attivato e sospetti che `IO:BufFileRead` e `IO:BufFileWrite` si verificano più frequentemente del normale. Per identificare l'origine del problema, è possibile impostare il parametro `log_temp_files` per registrare

tutte le query che generano file temporanei superiori alla soglia di KB specificata. Per impostazione predefinita, `log_temp_files` è impostato su `-1`, il che disattiva la funzionalità di registrazione. Se imposti questo parametro su `0`, RDS for PostgreSQL registra tutti i file temporanei. Se lo imposti su `1024`, RDS per PostgreSQL registra tutte le query che producono file temporanei di dimensioni superiori a 1 MB. Per ulteriori informazioni su `log_temp_files`, consulta Creazione di log e report di errore nella documentazione di PostgreSQL.

Esamina le tue query di join

È probabile che la tua query utilizzi i join. Ad esempio, la seguente query unisce quattro tabelle.

```
SELECT *
       FROM "order"
 INNER JOIN order_item
       ON (order.id = order_item.order_id)
 INNER JOIN customer
       ON (customer.id = order.customer_id)
 INNER JOIN customer_address
       ON (customer_address.customer_id = customer.id AND
           order.customer_address_id = customer_address.id)
 WHERE customer.id = 1234567890;
```

Una possibile causa di picchi nell'utilizzo temporaneo dei file è un problema nella query stessa. Ad esempio, una clausola interrotta potrebbe non filtrare correttamente i join. Considera il secondo inner join nell'esempio seguente.

```
SELECT *
       FROM "order"
 INNER JOIN order_item
       ON (order.id = order_item.order_id)
 INNER JOIN customer
       ON (customer.id = customer.id)
 INNER JOIN customer_address
       ON (customer_address.customer_id = customer.id AND
           order.customer_address_id = customer_address.id)
 WHERE customer.id = 1234567890;
```

La query precedente unisce erroneamente `customer.id` a `customer.id`, generando un prodotto cartesiano tra ogni cliente e ogni ordine. Questo tipo di join accidentale genera file temporanei di grandi dimensioni. A seconda delle dimensioni delle tabelle, una query cartesiana può anche riempire

lo spazio di archiviazione. La domanda potrebbe avere join cartesiani quando vengono soddisfatte le seguenti condizioni:

- Si notano forti e nitide riduzioni della disponibilità dello storage, seguite da un rapido ripristino.
- Non sono in fase di creazione di indici.
- Non vengono rilasciate istruzioni `CREATE TABLE FROM SELECT`.
- Nessuna vista materializzata viene aggiornata.

Per verificare se le tabelle vengono unite utilizzando le chiavi appropriate, ispezionare le direttive di mappatura relazionale di query e oggetti. Tieni presente che alcune query della tua applicazione non vengono sempre chiamate e alcune query vengono generate dinamicamente.

Esamina le query ORDER BY e GROUP BY

In alcuni casi, una clausola `ORDER BY` può comportare file temporanei eccessivi. Considera le linee guida seguenti:

- Includi solo colonne in una clausola `ORDER BY` quando devono essere ordinate. Questa linea guida è particolarmente importante per le query che restituiscono migliaia di righe e specificano molte colonne nella clausola `ORDER BY`.
- Considerando la creazione di indici per accelerare clausole `ORDER BY` quando corrispondono a colonne che hanno lo stesso ordine crescente o decrescente. Gli indici parziali sono preferibili perché sono più piccoli. Gli indici più piccoli vengono letti e attraversati più rapidamente.
- Se si creano indici per colonne che possono accettare valori nulli, considerare se si desidera che i valori nulli siano memorizzati alla fine o all'inizio degli indici.

  Se possibile, ridurre il numero di righe che devono essere ordinate filtrando il set di risultati. Se utilizzi istruzioni clausole o sottoquery `WITH`, ricorda che una query interna genera un set di risultati e lo passa alla query esterna. Maggiore è il numero di righe che una query può filtrare, minore è la necessità di ordinare la query.
- Se non è necessario ottenere il set completo di risultati, utilizzare la clausola `LIMIT`. Ad esempio, se si desidera solo le prime cinque righe, una query che utilizza la clausola `LIMIT` non continua a generare risultati. In questo modo, la query richiede meno memoria e file temporanei.

Una query che utilizza una clausola `GROUP BY` può richiedere anche file temporanei. Le query `GROUP BY` riepilogano i valori utilizzando funzioni come le seguenti:

- COUNT
- AVG
- MIN
- MAX
- SUM
- STDDEV

Per sintonizzare le query GROUP BY, segui i consigli per le query ORDER BY.

Evitare di utilizzare l'operazione DISTINCT

Se possibile, evitare di utilizzare l'operazione DISTINCT per rimuovere righe duplicate. Più righe non necessarie e duplicate restituite dalla tua query, più costosa diventa l'operazione DISTINCT. Se possibile, aggiungi filtri nella clausola WHERE anche se utilizzi gli stessi filtri per tabelle diverse. Filtrare la query e unirsi correttamente migliora le prestazioni e riduce l'utilizzo delle risorse. Previene inoltre report e risultati errati.

Se devi usare DISTINCT per più righe di una stessa tabella, prendi in considerazione la possibilità di creare un indice composito. Il raggruppamento di più colonne in un indice può migliorare il tempo necessario per valutare righe distinte. Inoltre, se utilizzi RDS per PostgreSQL versione 10 o successiva, puoi correlare le statistiche tra più colonne utilizzando il comando CREATE STATISTICS.

Considera l'utilizzo di funzioni finestra anziché le funzioni GROUP BY

Utilizzando GROUP BY si modifica il set di risultati e quindi recupera il risultato aggregato. Utilizzando le funzioni della finestra, si aggregano i dati senza modificare il set di risultati. Una funzione finestra utilizza la clausola OVER per eseguire calcoli tra i set definiti dalla query, correlando una riga con un'altra. È possibile utilizzare tutte le funzioni GROUP BY nelle funzioni di finestra, ma anche funzioni come le seguenti:

- RANK
- ARRAY_AGG
- ROW_NUMBER
- LAG
- LEAD

Per ridurre al minimo il numero di file temporanei generati da una funzione di finestra, rimuovere le duplicazioni per lo stesso set di risultati quando sono necessarie due aggregazioni distinte. Considera la query seguente.

```
SELECT sum(salary) OVER (PARTITION BY dept ORDER BY salary DESC) as sum_salary
     , avg(salary) OVER (PARTITION BY dept ORDER BY salary ASC) as avg_salary
  FROM empsalary;
```

È possibile riscrivere la query con la clausola `WINDOW` come segue.

```
SELECT sum(salary) OVER w as sum_salary
         , avg(salary) OVER w as_avg_salary
    FROM empsalary
  WINDOW w AS (PARTITION BY dept ORDER BY salary DESC);
```

Per impostazione predefinita, il planner di esecuzione RDS per PostgreSQL consolida nodi simili in modo da non duplicare le operazioni. Tuttavia, utilizzando una dichiarazione esplicita per il blocco finestra, è possibile mantenere la query più facilmente. È inoltre possibile migliorare le prestazioni impedendo la duplicazione.

Indagare sulle viste materializzate e le istruzioni CTAS

Quando una vista materializzata si aggiorna, esegue una query. Questa query può contenere un'operazione come `GROUP BY`, `ORDER BY` oppure`DISTINCT`. Durante un aggiornamento, è possibile osservare un numero elevato di file temporanei e gli eventi di attesa `IO:BufFileWrite` e `IO:BufFileRead`. Allo stesso modo, quando crei una tabella basata su una dichiarazione `SELECT`, l'istruzione `CREATE TABLE` esegue una query. Per ridurre i file temporanei necessari, ottimizza la query.

Uso di pg_repack per ricostruire gli indici

Quando crei un indice, il motore ordina il set di risultati. Man mano che le tabelle aumentano di dimensioni e man mano che i valori nella colonna indicizzata diventano più diversi, i file temporanei richiedono più spazio. Nella maggior parte dei casi, non è possibile impedire la creazione di file temporanei per tabelle di grandi dimensioni senza modificare l'area della memoria di lavoro di manutenzione. Per ulteriori informazioni su `maintenance_work_mem`, consulta https://www.postgresql.org/docs/current/runtime-config-resource.html nella documentazione di PostgreSQL.

Una possibile soluzione alternativa quando si ricrea un indice di grandi dimensioni consiste nell'utilizzare l'estensione pg_repack. Per ulteriori informazioni, consulta Riorganizzare le tabelle

nei database PostgreSQL con blocchi minimi nella documentazione di pg_repack. Per informazioni sull'impostazione dell'estensione nell'istanza database RDS per PostgreSQL, consulta Riduzione della dimensione nelle tabelle e negli indici con l'estensione pg_repack.

Aumenta maintenance_work_mem quando esegui cluster

Il comando `CLUSTER` raggruppa la tabella specificata da table_name in base a un indice esistente specificato da index_name. RDS per PostgreSQL ricrea fisicamente la tabella in modo che corrisponda all'ordine di un determinato indice.

Quando lo storage magnetico era prevalente, il clustering era comune perché il throughput di storage era limitato. Ora che lo storage basato su SSD è comune, il clustering è meno popolare. Tuttavia, se si raggruppano le tabelle, è comunque possibile aumentare leggermente le prestazioni a seconda delle dimensioni della tabella, dell'indice, della query e così via.

Se esegui il comando `CLUSTER` e osservi gli eventi di attesa `IO:BufFileWrite` e `IO:BufFileRead`, sintonizza `maintenance_work_mem`. Aumenta la dimensione della memoria a una quantità abbastanza grande. Un valore elevato significa che il motore può utilizzare più memoria per l'operazione di clustering.

Sintonizza la memoria per impedire IO:BuffileRead e IO:BuffileWrite

In alcune situazioni, è necessario ottimizzare la memoria. L'obiettivo è bilanciare la memoria tra le seguenti aree di consumo utilizzando i parametri appropriati, come segue.

- Il valore `work_mem`
- La memoria rimanente dopo aver scontato il valore `shared_buffers`
- Le connessioni massime aperte e in uso, limitate da `max_connections`

Per ulteriori informazioni sull'ottimizzazione della memoria, consulta Resource Consumption (Consumo delle risorse) nella documentazione di PostgreSQL.

Aumentare le dimensioni dell'area di memoria di lavoro

In alcune situazioni, l'unica opzione è aumentare la memoria utilizzata dalla sessione. Se le tue query sono scritte correttamente e utilizzano i tasti corretti per i join, prendi in considerazione la possibilità di aumentare il valore `work_mem`.

Per scoprire quanti file temporanei genera una query, imposta `log_temp_files` su `0`. Aumentando il valore `work_mem` al valore massimo identificato nei log, si impedisce alla query di generare

file temporanei. Tuttavia, `work_mem` imposta il massimo per nodo piano per ogni connessione o operatore parallelo. Se il database ha 5.000 connessioni e se ciascuna utilizza una memoria di 256 MiB, il motore necessita di 1,2 TiB di RAM. Pertanto, la tua istanza potrebbe esaurirsi dalla memoria.

Riserva una memoria sufficiente per il buffer pool condiviso

Il database utilizza aree di memoria come il buffer pool condiviso, non solo l'area di memoria di lavoro. Considerare i requisiti di queste aree di memoria aggiuntive prima di aumentare `work_mem`.

Ad esempio, supponi che la classe di istanza RDS per PostgreSQL sia db.r5.2xlarge. Questa classe ha 64 GiB di memoria. Per impostazione predefinita, il 25% della memoria è riservato al buffer pool condiviso. Dopo aver sottratto la quantità allocata all'area di memoria condivisa, rimangono 16.384 MB. Non allocare la memoria rimanente esclusivamente all'area della memoria di lavoro perché anche il sistema operativo e il motore richiedono memoria.

La memoria a cui è possibile allocare per `work_mem` dipende dalla classe di istanza. Se si utilizza una classe di istanza più grande, è disponibile più memoria. Tuttavia, nell'esempio precedente, non è possibile utilizzare più di 16 GiB. In caso contrario, la tua istanza diventa non disponibile quando esaurisce la memoria. Per ripristinare l'istanza dallo stato non disponibile, i servizi di automazione RDS per PostgreSQL si riavviano automaticamente.

Gestisci il numero di connessioni

Supponiamo che l'istanza del database disponga di 5.000 connessioni simultanee. Ogni connessione utilizza almeno 4 MiB di `work_mem`. L'elevato consumo di memoria delle connessioni rischia di peggiorare le prestazioni. Sono disponibili le seguenti opzioni:

- Eseguire l'aggiornamento a una classe di istanza più grande
- Diminuire il numero di connessioni simultanee al database utilizzando un proxy o un pool di connessioni.

Per i proxy, considera Amazon RDS Proxy, PGBouncer o un connection pooler basato sulla tua applicazione. Questa soluzione riduce il carico della CPU. Riduce inoltre il rischio quando tutte le connessioni richiedono l'area di memoria di lavoro. Quando esistono meno connessioni al database, è possibile aumentare il valore di `work_mem`. In questo modo, si riduce il verificarsi degli eventi di attesa `IO:BufFileRead` e `IO:BufFileWrite`. Inoltre, le query in attesa dell'area di memoria di lavoro accelerano in modo significativo.

IO: DataFileRead

L'evento `IO:DataFileRead` si verifica quando una connessione attende un processo di back-end per leggere una pagina richiesta dalla memoria perché la pagina non è disponibile nella memoria condivisa.

Argomenti

Versioni del motore supportate

Queste informazioni relative all'evento di attesa sono supportate per tutte le versioni di RDS per PostgreSQL.

Context

Tutte le query e le operazioni di manipolazione dei dati (DML) accedono alle pagine del buffer pool. Le dichiarazioni che possono indurre letture includono `SELECT`, `UPDATE` e `DELETE`. Ad esempio, un `UPDATE` può leggere pagine da tabelle o indici. Se la pagina richiesta o aggiornata non si trova nel buffer pool condiviso, questa lettura può portare all'evento `IO:DataFileRead`.

Poiché il buffer pool condiviso è finito, può essere riempito. In questo caso, le richieste di pagine che non sono in memoria impongono al database di leggere i blocchi dal disco. Se l'evento `IO:DataFileRead` si verifica frequentemente, il buffer pool condiviso potrebbe essere troppo piccolo per adattarsi al carico di lavoro. Questo problema è particolarmente grave per le query `SELECT` che leggono un numero elevato di righe che non rientrano nel buffer pool. Per ulteriori informazioni sul pool di buffer, consulta Resource Consumption (Consumo delle risorse) nella documentazione di PostgreSQL.

Probabili cause di aumento delle attese

Cause comuni dell'evento `IO:DataFileRead` includono quanto segue:

Picchi di connessione

Potresti trovare più connessioni che generano lo stesso numero di eventi IO: DataFileRead wait. In questo caso, può verificarsi un picco (aumento improvviso e grande) negli eventi `IO:DataFileRead`.

Le istruzioni SELECT e DML eseguono scansioni sequenziali

L'applicazione potrebbe aver eseguito una nuova operazione. Oppure un'operazione esistente potrebbe cambiare a causa di un nuovo piano di esecuzione. In questi casi, cerca tabelle (in particolare tabelle di grandi dimensioni) che abbiano un valore `seq_scan` maggiore. Puoi trovarli interrogando `pg_stat_user_tables`. Per tenere traccia delle query che generano più operazioni di lettura, utilizzare l'estensione `pg_stat_statements`.

CTAS e CREATE INDEX per set di dati di grandi dimensioni

Un CTAS è una `CREATE TABLE AS SELECT`dichiarazione. Se si esegue un CTAS utilizzando un set di dati di grandi dimensioni come origine o si crea un indice su una tabella di grandi dimensioni, l'evento `IO:DataFileRead` può verificarsi. Quando si crea un indice, il database potrebbe dover leggere l'intero oggetto utilizzando una scansione sequenziale. Un CTAS genera letture `IO:DataFile` quando le pagine non sono in memoria.

Diversi lavoratori sottovuoto in esecuzione contemporaneamente

Gli operatori del vuoto possono essere attivati manualmente o automaticamente. Raccomandiamo di adottare una strategia aggressiva per il vuoto. Tuttavia, quando una tabella contiene molte righe aggiornate o cancellate, l'attesa `IO:DataFileRead` aumenta. Dopo aver recuperato lo spazio, il tempo dedicato al vuoto su `IO:DataFileRead` diminuisce.

Ingresso di grandi quantità di dati

Quando l'applicazione acquisisce quantità di dati elevate, le operazioni `ANALYZE` potrebbero verificarsi più spesso. Il processo `ANALYZE` può essere attivato da un launcher automatico o richiamato manualmente.

L'operazione `ANALYZE` legge un sottoinsieme della tabella. Il numero di pagine che devono essere scansionate viene calcolato moltiplicando 30 per il valore `default_statistics_target`. Per ulteriori informazioni, consultare la documentazione di PostgreSQL. Il parametro `default_statistics_target` accetta valori compresi tra 1 e 10.000, dove il valore predefinito è 100.

Fame di risorse

Se si consuma la larghezza di banda di rete dell'istanza o la CPU, l'evento `IO:DataFileRead` potrebbe verificarsi più frequentemente.

Azioni

Consigliamo azioni diverse a seconda delle cause dell'evento di attesa.

Argomenti

- Controlla i filtri predicati per le query che generano attese
- Riduci al minimo l'effetto delle operazioni di manutenzione
- Rispondere a un numero elevato di connessioni

Controlla i filtri predicati per le query che generano attese

Supponiamo di identificare query specifiche che stanno generando eventi di attesa `IO:DataFileRead`. È possibile identificarli utilizzando le seguenti tecniche:

- Approfondimenti sulle prestazioni
- Viste catalogo come quella fornita dall'estensione `pg_stat_statements`
- La vista catalogo `pg_stat_all_tables`, se mostra periodicamente un numero maggiore di letture fisiche
- La vista `pg_statio_all_tables`, se lo mostra che i contatori `_read` sono in aumento

Si consiglia di determinare quali filtri vengono utilizzati nel predicato (clausola `WHERE`) di queste query. Seguire queste linee guida:

- Esegui il comando `EXPLAIN`. Nell'output, identificare quali tipi di scansioni vengono utilizzati. Una scansione sequenziale non indica necessariamente che ci sia un problema. Le query che utilizzano scansioni sequenziali producono naturalmente più eventi `IO:DataFileRead` rispetto alle query che utilizzano filtri.

  Scopri se la colonna elencata nella clausola `WHERE` è indicizzata. In caso contrario, prendi in considerazione la possibilità di creare un indice per questa colonna. Questo approccio evita le scansioni sequenziali e riduce gli eventi `IO:DataFileRead`. Se una query dispone di filtri restrittivi e continua a produrre scansioni sequenziali, valutare se vengono utilizzati gli indici appropriati.

- Scopri se la query sta accedendo a una tabella molto ampia. In alcuni casi, il partizionamento di una tabella può migliorare le prestazioni, consentendo alla query di leggere solo le partizioni necessarie.
- Esamina la cardinalità (numero totale di righe) dalle operazioni di join. Nota quanto sono restrittivi i valori che stai passando nei filtri per la tua clausola `WHERE`. Se possibile, sintonizza la query per ridurre il numero di righe passate in ogni fase del piano.

Riduci al minimo l'effetto delle operazioni di manutenzione

Operazioni di manutenzione come `VACUUM` e `ANALYZE` sono importanti. Si consiglia di non spegnerli qualora vengano trovati eventi di attesa `IO:DataFileRead` relativi a queste operazioni di manutenzione. I seguenti approcci possono ridurre al minimo l'effetto di queste operazioni:

- Eseguire manualmente le operazioni di manutenzione durante le ore non di punta. Questa tecnica impedisce al database di raggiungere la soglia per le operazioni automatiche.
- Per tabelle molto grandi, prendi in considerazione il partizionamento. Questa tecnica riduce il sovraccarico delle operazioni di manutenzione. Il database accede solo alle partizioni che richiedono manutenzione.
- Quando si acquisiscono grandi quantità di dati, prendere in considerazione la possibilità di disabilitare la funzione di analisi automatica.

La funzione autovacuum viene attivata automaticamente per una tabella quando la formula seguente è vera.

```
pg_stat_user_tables.n_dead_tup > (pg_class.reltuples x autovacuum_vacuum_scale_factor)
 + autovacuum_vacuum_threshold
```

La vista `pg_stat_user_tables` e il catalogo `pg_class` hanno più righe. Una riga può corrispondere a una riga della tabella. Questa formula presuppone che i `reltuples` sono per una tabella specifica. I parametri `autovacuum_vacuum_scale_factor` (0,20 per impostazione predefinita) e `autovacuum_vacuum_threshold` (50 tuple per impostazione predefinita) sono generalmente impostate globalmente per l'intera istanza. Tuttavia, è possibile impostare valori diversi per una tabella specifica.

Argomenti

- Ricerca delle tabelle che consumano spazio inutile

- Ricerca degli indici che consumano spazio inutile
- Trova tabelle idonee per l'autovacuum

Ricerca delle tabelle che consumano spazio inutile

Per trovare le tabelle che consumano spazio inutilmente, puoi utilizzare le funzioni dell'estensione di PostgreSQL`pgstattuple`. Questa estensione (modulo) è disponibile per impostazione predefinita su tutte le istanze database RDS per PostgreSQL e ne può essere creata un'istanza con il seguente comando.

```
CREATE EXTENSION pgstattuple;
```

Per ulteriori informazioni su questa estensione, consulta pgstattuple nella documentazione di PostgreSQL.

Puoi verificare l'aumento delle dimensioni della tabella e dell'indice nell'applicazione. Per ulteriori informazioni, consulta Diagnosi delle dimensioni della tabella e dell'indice.

Ricerca degli indici che consumano spazio inutile

Per trovare indici cresciuti e stimare la quantità di spazio consumato inutilmente sulle tabelle per le quali si dispone dei privilegi di lettura, è possibile eseguire la seguente query.

```
-- WARNING: rows with is_na = 't' are known to have bad statistics ("name" type is not
 supported).
-- This query is compatible with PostgreSQL 8.2 and later.

SELECT current_database(), nspname AS schemaname, tblname, idxname,
 bs*(relpages)::bigint AS real_size,
  bs*(relpages-est_pages)::bigint AS extra_size,
  100 * (relpages-est_pages)::float / relpages AS extra_ratio,
  fillfactor, bs*(relpages-est_pages_ff) AS bloat_size,
  100 * (relpages-est_pages_ff)::float / relpages AS bloat_ratio,
  is_na
  -- , 100-(sub.pst).avg_leaf_density, est_pages, index_tuple_hdr_bm,
  -- maxalign, pagehdr, nulldatawidth, nulldatahdrwidth, sub.reltuples, sub.relpages
  -- (DEBUG INFO)
FROM (
  SELECT coalesce(1 +
       ceil(reltuples/floor((bs-pageopqdata-pagehdr)/(4+nulldatahdrwidth)::float)), 0
       -- ItemIdData size + computed avg size of a tuple (nulldatahdrwidth)
```

```
    ) AS est_pages,
    coalesce(1 +
       ceil(reltuples/floor((bs-pageopqdata-pagehdr)*fillfactor/
(100*(4+nulldatahdrwidth)::float))), 0
    ) AS est_pages_ff,
    bs, nspname, table_oid, tblname, idxname, relpages, fillfactor, is_na
    -- , stattuple.pgstatindex(quote_ident(nspname)||'.'||quote_ident(idxname)) AS
 pst,
    -- index_tuple_hdr_bm, maxalign, pagehdr, nulldatawidth, nulldatahdrwidth,
 reltuples
    -- (DEBUG INFO)
  FROM (
    SELECT maxalign, bs, nspname, tblname, idxname, reltuples, relpages, relam,
 table_oid, fillfactor,
      ( index_tuple_hdr_bm +
          maxalign - CASE -- Add padding to the index tuple header to align on MAXALIGN
            WHEN index_tuple_hdr_bm%maxalign = 0 THEN maxalign
            ELSE index_tuple_hdr_bm%maxalign
          END
        + nulldatawidth + maxalign - CASE -- Add padding to the data to align on
 MAXALIGN
            WHEN nulldatawidth = 0 THEN 0
            WHEN nulldatawidth::integer%maxalign = 0 THEN maxalign
            ELSE nulldatawidth::integer%maxalign
          END
      )::numeric AS nulldatahdrwidth, pagehdr, pageopqdata, is_na
      -- , index_tuple_hdr_bm, nulldatawidth -- (DEBUG INFO)
    FROM (
      SELECT
        i.nspname, i.tblname, i.idxname, i.reltuples, i.relpages, i.relam, a.attrelid
 AS table_oid,
        current_setting('block_size')::numeric AS bs, fillfactor,
        CASE -- MAXALIGN: 4 on 32bits, 8 on 64bits (and mingw32 ?)
          WHEN version() ~ 'mingw32' OR version() ~ '64-bit|x86_64|ppc64|ia64|amd64'
 THEN 8
          ELSE 4
        END AS maxalign,
        /* per page header, fixed size: 20 for 7.X, 24 for others */
        24 AS pagehdr,
        /* per page btree opaque data */
        16 AS pageopqdata,
        /* per tuple header: add IndexAttributeBitMapData if some cols are null-able */
        CASE WHEN max(coalesce(s.null_frac,0)) = 0
          THEN 2 -- IndexTupleData size
```

```
         ELSE 2 + (( 32 + 8 - 1 ) / 8)
         -- IndexTupleData size + IndexAttributeBitMapData size ( max num filed per
 index + 8 - 1 /8)
       END AS index_tuple_hdr_bm,
       /* data len: we remove null values save space using it fractionnal part from
 stats */
       sum( (1-coalesce(s.null_frac, 0)) * coalesce(s.avg_width, 1024)) AS
 nulldatawidth,
       max( CASE WHEN a.atttypid = 'pg_catalog.name'::regtype THEN 1 ELSE 0 END ) > 0
 AS is_na
     FROM pg_attribute AS a
       JOIN (
         SELECT nspname, tbl.relname AS tblname, idx.relname AS idxname,
           idx.reltuples, idx.relpages, idx.relam,
           indrelid, indexrelid, indkey::smallint[] AS attnum,
           coalesce(substring(
             array_to_string(idx.reloptions, ' ')
              from 'fillfactor=([0-9]+)')::smallint, 90) AS fillfactor
         FROM pg_index
           JOIN pg_class idx ON idx.oid=pg_index.indexrelid
           JOIN pg_class tbl ON tbl.oid=pg_index.indrelid
           JOIN pg_namespace ON pg_namespace.oid = idx.relnamespace
         WHERE pg_index.indisvalid AND tbl.relkind = 'r' AND idx.relpages > 0
       ) AS i ON a.attrelid = i.indexrelid
       JOIN pg_stats AS s ON s.schemaname = i.nspname
         AND ((s.tablename = i.tblname AND s.attname =
 pg_catalog.pg_get_indexdef(a.attrelid, a.attnum, TRUE))
         -- stats from tbl
         OR  (s.tablename = i.idxname AND s.attname = a.attname))
         -- stats from functional cols
       JOIN pg_type AS t ON a.atttypid = t.oid
     WHERE a.attnum > 0
     GROUP BY 1, 2, 3, 4, 5, 6, 7, 8, 9
   ) AS s1
  ) AS s2
    JOIN pg_am am ON s2.relam = am.oid WHERE am.amname = 'btree'
) AS sub
-- WHERE NOT is_na
ORDER BY 2,3,4;
```

Trova tabelle idonee per l'autovacuum

Per trovare tabelle idonee per l'autovacuum, esegui la query riportata.

```
--This query shows tables that need vacuuming and are eligible candidates.
--The following query lists all tables that are due to be processed by autovacuum.
-- During normal operation, this query should return very little.
WITH  vbt AS (SELECT setting AS autovacuum_vacuum_threshold
              FROM pg_settings WHERE name = 'autovacuum_vacuum_threshold')
    , vsf AS (SELECT setting AS autovacuum_vacuum_scale_factor
              FROM pg_settings WHERE name = 'autovacuum_vacuum_scale_factor')
    , fma AS (SELECT setting AS autovacuum_freeze_max_age
              FROM pg_settings WHERE name = 'autovacuum_freeze_max_age')
    , sto AS (SELECT opt_oid, split_part(setting, '=', 1) as param,
                split_part(setting, '=', 2) as value
              FROM (SELECT oid opt_oid, unnest(reloptions) setting FROM pg_class) opt)
SELECT
    '"'||ns.nspname||'"."'||c.relname||'"' as relation
    , pg_size_pretty(pg_table_size(c.oid)) as table_size
    , age(relfrozenxid) as xid_age
    , coalesce(cfma.value::float, autovacuum_freeze_max_age::float)
 autovacuum_freeze_max_age
    , (coalesce(cvbt.value::float, autovacuum_vacuum_threshold::float) +
         coalesce(cvsf.value::float,autovacuum_vacuum_scale_factor::float) *
 c.reltuples)
         as autovacuum_vacuum_tuples
    , n_dead_tup as dead_tuples
FROM pg_class c
JOIN pg_namespace ns ON ns.oid = c.relnamespace
JOIN pg_stat_all_tables stat ON stat.relid = c.oid
JOIN vbt on (1=1)
JOIN vsf ON (1=1)
JOIN fma on (1=1)
LEFT JOIN sto cvbt ON cvbt.param = 'autovacuum_vacuum_threshold' AND c.oid =
 cvbt.opt_oid
LEFT JOIN sto cvsf ON cvsf.param = 'autovacuum_vacuum_scale_factor' AND c.oid =
 cvsf.opt_oid
LEFT JOIN sto cfma ON cfma.param = 'autovacuum_freeze_max_age' AND c.oid = cfma.opt_oid
WHERE c.relkind = 'r'
AND nspname <> 'pg_catalog'
AND (
    age(relfrozenxid) >= coalesce(cfma.value::float, autovacuum_freeze_max_age::float)
    or
    coalesce(cvbt.value::float, autovacuum_vacuum_threshold::float) +
      coalesce(cvsf.value::float,autovacuum_vacuum_scale_factor::float) * c.reltuples
 <= n_dead_tup
    -- or 1 = 1
```

```
)
ORDER BY age(relfrozenxid) DESC;
```

Rispondere a un numero elevato di connessioni

Quando monitori Amazon CloudWatch, potresti scoprire che la `DatabaseConnections` metrica aumenta. Questo aumento indica un numero maggiore di connessioni al database. Consigliamo quanto segue:

- Limita il numero di connessioni che l'applicazione può aprire con ciascuna istanza. Se l'applicazione dispone di una funzione di connection pool incorporata, impostare un numero ragionevole di connessioni. Basa il numero su ciò che le vCPU nell'istanza possono parallelizzare efficacemente.

  Se l'applicazione non utilizza una funzione di connection pool, considera l'utilizzo di Amazon RDS Proxy o un'alternativa. Questo approccio consente all'applicazione di aprire più connessioni con il bilanciamento del carico. Il bilanciatore può quindi aprire un numero limitato di connessioni con il database. Poiché un numero inferiore di connessioni sono in esecuzione in parallelo, l'istanza DB esegue meno commutazione di contesto nel kernel. Le query dovrebbero progredire più velocemente, causando un minor numero di eventi di attesa. Per ulteriori informazioni, consulta Utilizzo di Server proxy per Amazon RDS.
- Quando possibile, approfitta delle repliche di lettura di RDS per PostgreSQL. Quando l'applicazione esegue un'operazione di sola lettura, invia queste richieste alle repliche di lettura. Questa tecnica riduce la pressione I/O sul nodo primario (di scrittura).
- Prendi in considerazione la possibilità di scalare l'istanza database. Una classe di istanza a maggiore capacità fornisce più memoria, il che offre a RDS per PostgreSQL un buffer pool condiviso più ampio per contenere le pagine. Le dimensioni maggiori conferiscono inoltre all'istanza database più vCPU per gestire le connessioni. Più vCPU sono particolarmente utili quando le operazioni che stanno generando gli eventi di attesa `IO:DataFileRead` sono scritture.

IO:WALWrite

Argomenti

- Versioni del motore supportate
- Context
- Probabili cause di aumento delle attese

- Azioni

Versioni del motore supportate

Queste informazioni relative all'evento di attesa sono supportate per tutte le versioni di RDS per PostgreSQL 10 e successive.

Context

L'attività nel database che genera dati WAL riempie prima i buffer WAL e poi scrive su disco, in modo asincrono. L'evento di attesa `IO:WALWrite` viene generato quando la sessione SQL è in attesa del completamento della scrittura dei dati WAL su disco in modo da poter rilasciare la chiamata COMMIT della transazione.

Probabili cause di aumento delle attese

Se questo evento di attesa si verifica spesso, è necessario esaminare il carico di lavoro e il tipo di aggiornamenti che il carico di lavoro esegue e la loro frequenza. In particolare, cerca i seguenti tipi di attività.

Attività DML intensa

La modifica dei dati nelle tabelle del database non avviene istantaneamente. Un inserimento in una tabella potrebbe dover attendere l'inserimento o l'aggiornamento della stessa tabella di un altro client. Le istruzioni DML (Data Manipulation Language) per la modifica dei valori dei dati (INSERT, UPDATE, DELETE, COMMIT, ROLLBACK TRANSACTION) possono causare conflitti che fanno sì che il file WAL sia in attesa dello svuotamento dei buffer. Questa situazione è illustrata nelle seguenti metriche di Approfondimenti sulle prestazioni di Amazon RDS che indicano un'attività DML intensa.

- `tup_inserted`
- `tup_updated`
- `tup_deleted`
- `xcat_rollback`
- `xact_commit`

Per ulteriori informazioni su questi parametri, consulta Contatori Performance Insights per Amazon RDS for PostgreSQL.

Attività di punti di controllo frequenti

I punti di controllo frequenti contribuiscono a una maggiore dimensione del WAL. In RDS per PostgreSQL, le scritture di pagina intera sono sempre "attive". Le scritture di pagina intera aiutano a proteggersi dalla perdita di dati. Tuttavia, quando i punti di controllo si verificano troppo spesso, il sistema può presentare problemi generali di prestazioni. Ciò si verifica in particolare nei sistemi con un'attività DML intensa. In alcuni casi, potresti trovare messaggi di errore nel `postgresql.log` in cui si afferma che i punti di controllo si verificano troppo spesso.

Quando si ottimizzano i punti di controllo, si consiglia di bilanciare attentamente le prestazioni con il tempo previsto necessario per il ripristino in caso di arresto anomalo.

Azioni

Per ridurre il numero degli eventi di attesa, ti consigliamo di eseguire le seguenti azioni.

Argomenti

- Riduzione del numero di commit
- Monitoraggio dei punti di controllo
- Aumento dell'I/O
- Volume di registro dedicato (DLV)

Riduzione del numero di commit

Per ridurre il numero di commit, puoi combinare le istruzioni in blocchi di transazione. Usa Approfondimenti sulle prestazioni di Amazon RDS per esaminare il tipo di query eseguite. È inoltre possibile spostare le operazioni di manutenzione di grandi dimensioni nelle ore non di punta. Ad esempio, crea gli indici o utilizza le operazioni `pg_repack` durante le ore non di produzione.

Monitoraggio dei punti di controllo

È possibile monitorare due parametri per verificare la frequenza con cui l'istanza database RDS per PostgreSQL scrive nel file WAL per i punti di controllo.

- `log_checkpoints` - Questo parametro è impostato su "on" (attivo) per impostazione predefinita. Fa sì che venga inviato un messaggio al log di PostgreSQL per ogni punto di controllo. Questi messaggi di log includono il numero di buffer scritti, il tempo impiegato per scriverli e il numero di file WAL aggiunti, rimossi o riciclati per il punto di controllo specificato.

Per ulteriori informazioni su questo parametro, consulta Error Reporting and Logging (Creazione di report e log degli errori) nella documentazione di PostgreSQL.

- `checkpoint_warning` - Questo parametro imposta un valore di soglia (in secondi) per la frequenza dei punti di controllo al di sopra del quale viene generato un avviso. Per impostazione predefinita, questo parametro non è impostato in RDS per PostgreSQL. È possibile impostare il valore di questo parametro per ricevere un avviso quando le modifiche al database nell'istanza database RDS per PostgreSQL vengono scritte a una velocità per la quale i file WAL non sono di dimensioni idonee per la gestione. Ad esempio, supponi di impostare questo parametro su 30. Se l'istanza RDS per PostgreSQL deve scrivere le modifiche con una frequenza maggiore rispetto a ogni 30 secondi, nel log di PostgreSQL viene inviato un avviso indicante che i checkpoint si verificano con una frequenza eccessiva. Questo può indicare che il valore `max_wal_size` deve essere aumentato.

  Per ulteriori informazioni consulta Write Ahead Log nella documentazione di PostgreSQL.

Aumento dell'I/O

Questo tipo di evento di attesa di input/output (I/O) può essere risolto aumentando le operazioni di input/output al secondo (IOPS) per fornire un I/O più rapido. L'aumento dell'I/O è preferibile a quello della CPU, poiché l'aumento della CPU può comportare ancora più conflitti in termini di I/O in quanto può gestire più lavoro e quindi peggiorare ulteriormente il collo di bottiglia dell'I/O. Come regola generale, ti consigliamo di ottimizzare il carico di lavoro prima di eseguire operazioni di scalabilità.

Volume di registro dedicato (DLV)

Puoi utilizzare un volume di log dedicato (DLV) per un'istanza database che usa l'archiviazione della capacità di IOPS allocata tramite la console Amazon RDS, la AWS CLI o l'API Amazon RDS. Un DLV sposta i log delle transazioni del database PostgreSQL in un volume di archiviazione separato dal volume contenente le tabelle del database. Per ulteriori informazioni, consulta Volume di registro dedicato (DLV).

Lock:advisory

L'evento `Lock:advisory` si verifica quando un'applicazione PostgreSQL utilizza un blocco per coordinare l'attività su più sessioni.

Argomenti

- Versioni di motori pertinenti

Versioni di motori pertinenti

Queste informazioni relative all'evento di attesa sono supportate per RDS per PostgreSQL versione 9.6 e successive.

Context

I blocchi di consulenza PostgreSQL sono blocchi cooperativi a livello di applicazione esplicitamente bloccati e sbloccati dal codice dell'applicazione dell'utente. Un'applicazione PostgreSQL può utilizzare un blocco per coordinare l'attività su più sessioni. A differenza dei normali blocchi a livello di oggetto o riga, l'applicazione ha il pieno controllo sulla durata del blocco. Per ulteriori informazioni consulta Blocchi di consulenza nella documentazione di PostgreSQL.

I blocchi di consulenza possono essere rilasciati prima della fine di una transazione o essere trattenuti da una sessione tra le transazioni. Ciò tuttavia non è vero per i blocchi impliciti e applicati al sistema, come un blocco esclusivo di accesso su una tabella acquisita da una dichiarazione `CREATE INDEX`.

Per una descrizione delle funzioni utilizzate per acquisire (bloccare) e rilasciare (sbloccare) i blocchi di consulenza, vedere Funzioni di Advisory Lock nella documentazione di PostgreSQL.

I blocchi di consulenza sono implementati sopra il normale sistema di blocco PostgreSQL e sono visibili nella visualizzazione di sistema `pg_locks`.

Cause

Questo tipo di blocco è controllato esclusivamente da un'applicazione che lo utilizza esplicitamente. I blocchi di consulenza acquisiti per ogni riga come parte di una query possono causare un picco di blocchi o un accumulo a lungo termine.

Questi effetti si verificano quando la query viene eseguita in un modo che acquisisce blocchi su più righe di quelle restituite dalla query. L'applicazione dovrà comunque rilasciare ogni blocco, ma se i blocchi vengono acquisiti su righe che non vengono restituite, l'applicazione non riesce a trovare tutti i blocchi.

L'esempio seguente è tratto da Blocchi di consulenza nella documentazione di PostgreSQL.

```
SELECT pg_advisory_lock(id) FROM foo WHERE id > 12345 LIMIT 100;
```

In questo esempio, la clausola `LIMIT` può arrestare l'output della query solo dopo che le righe sono già state selezionate internamente e i relativi valori ID bloccati. Ciò può accadere improvvisamente quando un volume di dati crescente fa sì che il pianificatore scelga un piano di esecuzione diverso che non è stato testato durante lo sviluppo. L'accumulo in questo caso avviene perché l'applicazione chiama esplicitamente `pg_advisory_unlock` per ogni valore ID bloccato. Tuttavia, in questo caso non è possibile trovare il set di blocchi acquisiti su righe che non sono state restituite. Poiché i blocchi vengono acquisiti a livello di sessione, non vengono rilasciati automaticamente alla fine della transazione.

Un'altra possibile causa di picchi nei tentativi di blocco bloccati sono i conflitti non intenzionali. In questi conflitti, parti non correlate dell'applicazione condividono per errore lo stesso spazio ID di blocco.

Operazioni

Esaminare l'utilizzo delle applicazioni dei blocchi di consulenza e i dettagli su dove e quando nel flusso dell'applicazione viene acquisito e rilasciato ogni tipo di blocco consultivo.

Determina se una sessione sta acquisendo troppi blocchi o che una sessione di lunga durata non rilascia blocchi abbastanza presto, causando un lento accumulo di blocchi. È possibile correggere un lento accumulo di blocchi a livello di sessione terminando la sessione utilizzando `pg_terminate_backend(pid)`.

Viene visualizzato un client in attesa di un blocco di avviso in `pg_stat_activity` con `wait_event_type=Lock` e `wait_event=advisory`. È possibile ottenere valori di blocco specifici eseguendo una query nella vista di sistema `pg_locks` per lo stesso `pid`, cercando `locktype=advisory` e `granted=f`.

È quindi possibile identificare la sessione di blocco interrogando `pg_locks` per lo stesso blocco consultivo `granted=t`, come mostrato nell'esempio seguente.

```
SELECT blocked_locks.pid AS blocked_pid,
         blocking_locks.pid AS blocking_pid,
         blocked_activity.usename AS blocked_user,
         blocking_activity.usename AS blocking_user,
         now() - blocked_activity.xact_start AS blocked_transaction_duration,
         now() - blocking_activity.xact_start AS blocking_transaction_duration,
```

```
        concat(blocked_activity.wait_event_type,':',blocked_activity.wait_event) AS
blocked_wait_event,
        concat(blocking_activity.wait_event_type,':',blocking_activity.wait_event) AS
blocking_wait_event,
        blocked_activity.state AS blocked_state,
        blocking_activity.state AS blocking_state,
        blocked_locks.locktype AS blocked_locktype,
        blocking_locks.locktype AS blocking_locktype,
        blocked_activity.query AS blocked_statement,
        blocking_activity.query AS blocking_statement
   FROM pg_catalog.pg_locks blocked_locks
   JOIN pg_catalog.pg_stat_activity blocked_activity ON blocked_activity.pid =
blocked_locks.pid
   JOIN pg_catalog.pg_locks blocking_locks
       ON blocking_locks.locktype = blocked_locks.locktype
       AND blocking_locks.DATABASE IS NOT DISTINCT FROM blocked_locks.DATABASE
       AND blocking_locks.relation IS NOT DISTINCT FROM blocked_locks.relation
       AND blocking_locks.page IS NOT DISTINCT FROM blocked_locks.page
       AND blocking_locks.tuple IS NOT DISTINCT FROM blocked_locks.tuple
       AND blocking_locks.virtualxid IS NOT DISTINCT FROM blocked_locks.virtualxid
       AND blocking_locks.transactionid IS NOT DISTINCT FROM
blocked_locks.transactionid
       AND blocking_locks.classid IS NOT DISTINCT FROM blocked_locks.classid
       AND blocking_locks.objid IS NOT DISTINCT FROM blocked_locks.objid
       AND blocking_locks.objsubid IS NOT DISTINCT FROM blocked_locks.objsubid
       AND blocking_locks.pid != blocked_locks.pid
   JOIN pg_catalog.pg_stat_activity blocking_activity ON blocking_activity.pid =
blocking_locks.pid
   WHERE NOT blocked_locks.GRANTED;
```

Tutte le funzioni API di blocco consultivo hanno due serie di argomenti, un argomento `bigint` o due argomenti `integer`:

- Per le funzioni API con un argomento `bigint`, i 32 bit superiori sono in `pg_locks.classid` e i 32 bit inferiori sono in `pg_locks.objid`.
- Per le funzioni API con due argomenti `integer`, il primo argomento è `pg_locks.classid` e il secondo argomento è `pg_locks.objid`.

Il valore `pg_locks.objsubid` indica quale modulo API è stato utilizzato: 1 significa un argomento `bigint`; 2 significa due argomenti `integer`.

Lock:extend

L'evento `Lock:extend` si verifica quando un processo di back-end è in attesa di bloccare una relazione per estenderla mentre un altro processo ha un blocco su tale relazione per lo stesso scopo.

Argomenti

Versioni del motore supportate

Queste informazioni relative all'evento di attesa sono supportate per tutte le versioni di RDS per PostgreSQL.

Context

L'evento `Lock:extend` indica che un processo di back-end è in attesa di estendere una relazione su cui un altro processo di backend mantiene un blocco mentre sta estendendo tale relazione. Poiché solo un processo alla volta può estendere una relazione, il sistema genera un evento di attesa `Lock:extend`. Le operazioni `INSERT`, `COPY`, e `UPDATE` possono generare questo evento.

Probabili cause di aumento delle attese

Quando l'evento `Lock:extend` si verifica più del normale, probabilmente indicando un problema di prestazioni, le cause tipiche includono le seguenti.

Aumento degli inserti simultanei o degli aggiornamenti della stessa tabella

Potrebbe esserci un aumento del numero di sessioni simultanee con query che inseriscono o aggiornano la stessa tabella.

Larghezza di banda di rete insufficiente

La larghezza di banda di rete sull'istanza database potrebbe essere insufficiente per le esigenze di comunicazione di storage del carico di lavoro corrente. Ciò può contribuire alla latenza dello storage che causa un aumento degli eventi `Lock:extend`.

Operazioni

Consigliamo azioni diverse a seconda delle cause dell'evento di attesa.

Argomenti

- Riduci gli inserti e gli aggiornamenti simultanei alla stessa relazione
- Aumentare la larghezza di banda di rete

Riduci gli inserti e gli aggiornamenti simultanei alla stessa relazione

Innanzitutto, determinare se c'è un aumento dei parametri `tup_inserted` e `tup_updated` e un aumento di questi eventi di attesa. In tal caso, verificare quali relazioni sono in forte contesa per le operazioni di inserimento e aggiornamento. Per determinarlo, interrogare la vista `pg_stat_all_tables` per i valori nei campi `n_tup_ins` e `n_tup_upd`. Per ulteriori informazioni sulla vista `pg_stat_all_tables`, consultare pg_stat_all_tables nella documentazione PostgreSQL.

Per ottenere ulteriori informazioni sul blocco e le query bloccate, eseguire una query `pg_stat_activity` come nel seguente esempio:

```
SELECT
    blocked.pid,
    blocked.usename,
    blocked.query,
    blocking.pid AS blocking_id,
    blocking.query AS blocking_query,
    blocking.wait_event AS blocking_wait_event,
    blocking.wait_event_type AS blocking_wait_event_type
FROM pg_stat_activity AS blocked
JOIN pg_stat_activity AS blocking ON blocking.pid = ANY(pg_blocking_pids(blocked.pid))
where
blocked.wait_event = 'extend'
and blocked.wait_event_type = 'Lock';

   pid  | usename  |            query             | blocking_id |
            blocking_query                          | blocking_wait_event |
 blocking_wait_event_type
  ------+----------+------------------------------+-------------
+-------------------------------------------------------------------
+---------------------+--------------------------
   7143 |  myuser  | insert into tab1 values (1); |        4600 | INSERT INTO tab1 (a)
 SELECT s FROM generate_series(1,1000000) s; | DataFileExtend      | IO
```

Dopo aver identificato le relazioni che contribuiscono ad aumentare gli eventi `Lock:extend`, utilizza le seguenti tecniche per ridurre la contesa:

- Scopri se è possibile utilizzare il partizionamento per ridurre le contese per la stessa tabella. La separazione delle tuple inserite o aggiornate in diverse partizioni può ridurre le contese. Per informazioni sulle partizioni, consulta Gestione delle partizioni PostgreSQL con l'estensione pg_partman.
- Se l'evento di attesa è dovuto principalmente all'attività di aggiornamento, considerare di ridurre il valore del fattore di riempimento della relazione. Ciò può ridurre le richieste di nuovi blocchi durante l'aggiornamento. Il fattore di riempimento è un parametro di archiviazione per una tabella che determina la quantità massima di spazio per l'imballaggio di una pagina di tabella. Viene espresso come percentuale dello spazio totale per una pagina. Per ulteriori informazioni sul parametro fillfactor, consulta CREA TABELLA nella documentazione di PostgreSQL.

> **Important**
>
> Si consiglia vivamente di testare il sistema se si modifica il fattore di riempimento perché la modifica di questo valore può influire negativamente sulle prestazioni, a seconda del carico di lavoro.

Aumentare la larghezza di banda di rete

Per vedere se c'è un aumento della latenza di scrittura, controlla il parametro `WriteLatency` in CloudWatch. Se è presente, usa le metriche Amazon CloudWatch `WriteThroughput` e `ReadThroughput` per monitorare il traffico relativo all'archiviazione sull'istanza database. Questi parametri possono aiutarti a determinare se la larghezza di banda della rete è sufficiente per il tuo carico di lavoro.

Se la larghezza di banda della rete non è sufficiente, aumentala. Se il file client o l'istanza DB sta raggiungendo i limiti di larghezza di banda di rete, l'unico modo per aumentare la larghezza di banda è aumentare la dimensione dell'istanza DB.

Per ulteriori informazioni sui parametri di CloudWatch, consultare Parametri a CloudWatch livello di istanza Amazon per Amazon RDS. Per informazioni sulle prestazioni di rete per una classe di istanza database, consulta Parametri a CloudWatch livello di istanza Amazon per Amazon RDS.

Lock:Relation

L'evento `Lock:Relation` si verifica quando una query è in attesa di acquisire un blocco su una tabella o vista (relazione) attualmente bloccata da un'altra transazione.

Argomenti

Versioni del motore supportate

Queste informazioni relative all'evento di attesa sono supportate per tutte le versioni di RDS per PostgreSQL.

Context

La maggior parte dei comandi PostgreSQL utilizza implicitamente i blocchi per controllare l'accesso simultaneo ai dati nelle tabelle. È inoltre possibile utilizzare questi blocchi esplicitamente nel codice dell'applicazione con il comando `LOCK`. Molte modalità di blocco non sono compatibili tra loro e possono bloccare le transazioni quando cercano di accedere allo stesso oggetto. Quando ciò accade, RDS per PostgreSQL genera un evento `Lock:Relation`. Di seguito sono riportati alcuni esempi comuni:

- Blocchi esclusivi come `ACCESS EXCLUSIVE` possono bloccare tutti gli accessi simultanei. Le operazioni DDL (Data Definition Language) come `DROP TABLE`, `TRUNCATE`, `VACUUM FULL`, e `CLUSTER` acquisiscono implicitamente i blocchi `ACCESS EXCLUSIVE`. `ACCESS EXCLUSIVE` è anche la modalità di blocco predefinita per le istruzioni `LOCK TABLE` che non specificano esplicitamente una modalità.
- L'uso di `CREATE INDEX (without CONCURRENT)` su una tabella è in conflitto con le istruzioni DML (Data Manipulation Language) `UPDATE`, `DELETE`, e `INSERT`, che acquisiscono i blocchi `ROW EXCLUSIVE`.

Per ulteriori informazioni sui blocchi a livello di tabella e sulle modalità di blocco in conflitto, vedere Blocco esplicito nella documentazione di PostgreSQL.

Il blocco di query e transazioni in genere si sblocca in uno dei seguenti modi:

- Query di blocco: l'applicazione può annullare la query o l'utente può terminare il processo. Il motore può anche forzare la fine della query a causa del timeout dell'istruzione di una sessione o di un meccanismo di rilevamento del deadlock.
- Blocco della transazione: una transazione smette di bloccarsi quando esegue un'istruzione `ROLLBACK` o `COMMIT`. I rollback si verificano automaticamente anche quando le sessioni vengono disconnesse da un client o da problemi di rete o terminano. Le sessioni possono essere terminate quando il motore di database è spento, quando il sistema è fuori memoria e così via.

Probabili cause di aumento delle attese

Quando l'evento `Lock:Relation` si verifica più frequentemente del normale, può indicare un problema di prestazioni. Le cause tipiche sono:

Sessioni simultanee aumentate con blocchi di tabella in conflitto

Potrebbe esserci un aumento del numero di sessioni simultanee con query che inseriscono o aggiornano la stessa tabella.

Operazioni di manutenzione

Le operazioni di manutenzione Health come `VACUUM` e `ANALYZE` possono aumentare significativamente il numero di blocchi in conflitto. `VACUUM FULL` acquisisce un blocco `ACCESS EXCLUSIVE`, e `ANALYSE` acquisisce un blocco `SHARE UPDATE EXCLUSIVE`. Entrambi i tipi di blocchi possono causare un evento di attesa `Lock:Relation`. Le operazioni di manutenzione dei dati delle applicazioni, come l'aggiornamento di una vista materializzata, possono anche aumentare le query e le transazioni bloccate.

Blocchi sulle istanze del lettore

È possibile che si verifichi un conflitto tra i blocchi di relazione tenuti dallo scrittore e dai lettori. Attualmente solo i blocchi di relazione `ACCESS EXCLUSIVE` vengono replicati nelle istanze del lettore. Tuttavia, il blocco di relazione `ACCESS EXCLUSIVE` sarà in conflitto con qualsiasi blocco di relazione `ACCESS SHARE` tenuto dal lettore. Questo conflitto può causare un aumento degli eventi di attesa della relazione di blocco sul lettore.

Operazioni

Consigliamo azioni diverse a seconda delle cause dell'evento di attesa.

Argomenti

Riduci l'impatto del blocco delle istruzioni SQL

Per ridurre l'impatto del blocco delle istruzioni SQL, modificare il codice dell'applicazione laddove possibile. Di seguito sono riportate due tecniche comuni per ridurre i blocchi:

- Utilizzo dell'opzione `NOWAIT` — Alcuni comandi SQL, come le istruzioni `SELECT` e `LOCK`, supportano questa opzione. La direttiva `NOWAIT` annulla la query richiedente il blocco se il blocco non può essere acquisito immediatamente. Questa tecnica può aiutare a impedire che una sessione di blocco provochi un accumulo di sessioni bloccate dietro di essa.

  Ad esempio: si supponga che la transazione A sia in attesa di un blocco trattenuto dalla transazione B. Ora, se B richiede un blocco su una tabella bloccata dalla transazione C, la transazione A potrebbe essere bloccata fino al completamento della transazione C. Ma se la transazione B utilizza un `NOWAIT` quando richiede il blocco su C, può fallire rapidamente e garantire che la transazione A non debba attendere indefinitamente.
- Utilizza `SET lock_timeout` — Imposta un valore `lock_timeout` per limitare il tempo in cui un'istruzione SQL attende di acquisire un blocco su una relazione. Se il blocco non viene acquisito entro il timeout specificato, la transazione che richiede il blocco viene annullata. Impostare questo valore a livello di sessione.

Riduci al minimo l'effetto delle operazioni di manutenzione

Operazioni di manutenzione come `VACUUM` e `ANALYZE` sono importanti. Si consiglia di non spegnerli qualora vengano trovati eventi di attesa `Lock:Relation` relativi a queste operazioni di manutenzione. I seguenti approcci possono ridurre al minimo l'effetto di queste operazioni:

- Eseguire manualmente le operazioni di manutenzione durante le ore non di punta.
- Per ridurre le attese `Lock:Relation` causate da attività autovacuum, eseguire qualsiasi sintonizzazione automatica necessaria. Per informazioni sulla sintonizzazione autovacuum, fai riferimento a Funzionamento di PostgreSQL Autovacuum in Amazon RDS nella Guida per l'utente di Amazon RDS.

Lock:transactionid

L'evento `Lock:transactionid` si verifica quando una transazione è in attesa di un blocco a livello di riga.

Argomenti

Versioni del motore supportate

Queste informazioni relative all'evento di attesa sono supportate per tutte le versioni di RDS per PostgreSQL.

Context

L'evento `Lock:transactionid` si verifica quando una transazione sta tentando di acquisire un blocco a livello di riga già concesso a una transazione in esecuzione contemporaneamente. La sessione che mostra il l'evento di attesa `Lock:transactionid` è bloccato a causa di questo blocco. Dopo che la transazione di blocco termina in un'istruzione `COMMIT` o `ROLLBACK`, la transazione bloccata può procedere.

La semantica multiversione di controllo della concorrenza di RDS per PostgreSQL garantisce che i lettori non bloccano gli scrittori e gli scrittori non bloccano i lettori. Affinché si verifichino conflitti a livello di riga, le transazioni bloccate e bloccate devono emettere dichiarazioni in conflitto dei seguenti tipi:

- `UPDATE`
- `SELECT … FOR UPDATE`
- `SELECT … FOR KEY SHARE`

L'istruzione `SELECT … FOR KEY SHARE` è un caso speciale. Il database utilizza la clausola `FOR KEY SHARE` per ottimizzare le prestazioni dell'integrità referenziale. Un blocco a livello di riga su una fila può bloccare i comandi `INSERT`, `UPDATE`, e `DELETE` su altre tabelle che fanno riferimento alla riga.

Probabili cause di aumento delle attese

Quando questo evento appare più del normale, la causa è in genere un'istruzione `UPDATE`, `SELECT … FOR UPDATE`, oppure `SELECT … FOR KEY SHARE` combinate con le seguenti condizioni.

Argomenti

Elevata concorrenza

RDS per PostgreSQL può utilizzare la semantica di blocco granulare a livello di riga. La probabilità di conflitti a livello di riga aumenta quando vengono soddisfatte le seguenti condizioni:

- Un carico di lavoro altamente simultaneo è conteso per le stesse righe.
- Aumenta la concorrenza.

Inattivo in transazione

A volte la colonna `pg_stat_activity.state` mostra il valore `idle in transaction`. Questo valore viene visualizzato per le sessioni che hanno avviato una transazione, ma non hanno ancora emesso un `COMMIT` o `ROLLBACK`. Se il valore `pg_stat_activity.state` non è `active`, la query mostrata in `pg_stat_activity` è la versione più recente a terminare l'esecuzione. La sessione di blocco non sta elaborando attivamente una query perché una transazione aperta contiene un blocco.

Se una transazione inattiva ha acquisito un blocco a livello di riga, potrebbe impedire ad altre sessioni di acquisirlo. Questa condizione porta al frequente verificarsi dell'evento di attesa `Lock:transactionid`. Per diagnosticare il problema, esaminare l'output da `pg_stat_activity` e `pg_locks`.

Transazioni di lunga durata

Le transazioni che vengono eseguite a lungo ricevono blocchi per un lungo periodo di tempo. Questi blocchi a tenuta lunga possono impedire l'esecuzione di altre transazioni.

Operazioni

Il blocco delle righe è un conflitto tra le istruzioni `UPDATE`, `SELECT … FOR UPDATE`, oppure `SELECT … FOR KEY SHARE`. Prima di tentare una soluzione, scopri quando queste istruzioni sono in esecuzione sulla stessa riga. Utilizzare queste informazioni per scegliere una strategia descritta nelle sezioni seguenti.

Argomenti

- Rispondere a un'elevata concorrenza
- Rispondere alle transazioni inattive
- Rispondere alle transazioni di lunga durata

Rispondere a un'elevata concorrenza

Se il problema è la concorrenza, prova una delle seguenti tecniche:

- Riduci la concorrenza nell'applicazione. Ad esempio, diminuisci il numero di sessioni attive.
- Implementa un pool di connessioni. Per informazioni su come mettere in pool le connessioni con RDS Proxy, vedere Utilizzo di Server proxy per Amazon RDS.
- Progettare l'applicazione o il modello di dati per evitare di contendere le istruzioni `UPDATE` e `SELECT … FOR UPDATE`. È inoltre possibile ridurre il numero di chiavi esterne a cui accedono le istruzioni `SELECT … FOR KEY SHARE`.

Rispondere alle transazioni inattive

Se `pg_stat_activity.state` mostra `idle in transaction`, utilizza le seguenti strategie:

- Attiva autocommit laddove possibile. Questo approccio impedisce alle transazioni di bloccare altre transazioni durante l'attesa di un `COMMIT` o `ROLLBACK`.
- Cerca percorsi di codice a cui manca `COMMIT`, `ROLLBACK`, oppure `END`.
- Assicurati che la logica di gestione delle eccezioni nell'applicazione abbia sempre un percorso per un `end of transaction` valido.
- Assicurati che l'applicazione elabori i risultati delle query dopo aver terminato la transazione con `COMMIT` o `ROLLBACK`.

Rispondere alle transazioni di lunga durata

Se le transazioni di lunga durata causano il frequente verificarsi di `Lock:transactionid`, prova le seguenti strategie:

- Tieni i blocchi di riga fuori dalle transazioni di lunga durata.
- Limita la durata delle query implementando autocommit quando possibile.

Lock:tuple

L'evento `Lock:tuple` si verifica quando un processo di backend aspetta di acquisire un blocco su una tupla.

Argomenti

- Versioni del motore supportate
- Context
- Probabili cause di aumento delle attese
- Operazioni

Versioni del motore supportate

Queste informazioni relative all'evento di attesa sono supportate per tutte le versioni di RDS per PostgreSQL.

Context

L'evento `Lock:tuple` indica che un backend è in attesa di acquisire un blocco su una tupla mentre un altro backend tiene un blocco in conflitto sulla stessa tupla. Nella tabella seguente viene illustrato uno scenario in cui le sessioni generano l'evento `Lock:tuple`.

Orari	Sessione 1	Sessione 2	Sessione 3
t1	Inizia una transazione.		
t2	Aggiorna la riga 1.		
t3		Aggiorna la riga 1. La sessione acquisisce un	

Orari	Sessione 1	Sessione 2	Sessione 3
		blocco esclusivo sulla tupla e quindi attende che la sessione 1 rilasci il blocco eseguendo il commit o il rollback.	
t4			Aggiorna la riga 1. La sessione attende che la sessione 2 rilasci il blocco esclusivo sulla tupla.

Oppure puoi simulare questo evento di attesa utilizzando lo strumento di benchmarking `pgbench`. Configurare un numero elevato di sessioni simultanee per aggiornare la stessa riga in una tabella con un file SQL personalizzato.

Per ulteriori informazioni sulle modalità di blocco in conflitto, vedere Blocco esplicito nella documentazione di PostgreSQL. Per ulteriori informazioni su `pgbench`, consulta pgbench nella documentazione di PostgreSQL.

Probabili cause di aumento delle attese

Quando l'evento si verifica più del normale, probabilmente indicando un problema di prestazioni, le cause tipiche includono le seguenti:

- Un numero elevato di sessioni simultanee sta cercando di acquisire un blocco in conflitto per la stessa tupla eseguendo istruzioni `UPDATE` o `DELETE`.
- Sessioni altamente simultanee stanno eseguendo un'istruzione `SELECT` usando le modalità di blocco `FOR UPDATE` o `FOR NO KEY UPDATE`.
- Diversi fattori spingono le applicazioni o i connection pool ad aprire più sessioni per eseguire le stesse operazioni. Mentre le nuove sessioni stanno tentando di modificare le stesse righe, il carico del DB può aumentare e `Lock:tuple` può apparire.

Per ulteriori informazioni consulta Blocchi a livello di riga nella documentazione di PostgreSQL.

Operazioni

Consigliamo azioni diverse a seconda delle cause dell'evento di attesa.

Argomenti

- Indagare la logica dell’applicazione
- Trova la sessione responsabile del blocco
- Riduci la concorrenza quando è alta
- Risoluzione dei problemi dei colli di bottiglia

Indagare la logica dell’applicazione

Scopri se una sessione bloccante è rimasta in stato `idle in transaction` per lungo tempo. In tal caso, considera di terminare la sessione di blocco come soluzione a breve termine. È anche possibile usare la funzione `pg_terminate_backend`. Per ulteriori informazioni su questa funzione, consulta Funzioni di segnalazione server nella documentazione di PostgreSQL.

Per una soluzione a lungo termine, fai quanto seguente:

- Regola la logica dell'applicazione.
- Utilizzo del parametro `idle_in_transaction_session_timeout`. Questo parametro termina qualsiasi sessione con una transazione aperta che è rimasta inattiva per un periodo di tempo superiore al periodo di tempo specificato. Per ulteriori informazioni, consulta la pagina Errori connessione client nella documentazione di PostgreSQL.
- Usa autocommit il più possibile. Per ulteriori informazioni, consulta la pagina CONFIGURA AUTOCOMMIT nella documentazione di PostgreSQL.

Trova la sessione responsabile del blocco

Mentre si verifica l’evento di attesa `Lock:tuple`, identifica il blocco e la sessione bloccata scoprendo quali blocchi dipendono l'uno dall'altro. Per ulteriori informazioni, consulta Informazioni sulle dipendenze dei blocchi nel wiki di PostgreSQL.

L'esempio seguente esegue una query su tutte le sessioni, filtrando su `tuple` e ordinando per `wait_time`.

```
SELECT blocked_locks.pid AS blocked_pid,
         blocking_locks.pid AS blocking_pid,
```

```
        blocked_activity.usename AS blocked_user,
        blocking_activity.usename AS blocking_user,
        now() - blocked_activity.xact_start AS blocked_transaction_duration,
        now() - blocking_activity.xact_start AS blocking_transaction_duration,
        concat(blocked_activity.wait_event_type,':',blocked_activity.wait_event) AS
blocked_wait_event,
        concat(blocking_activity.wait_event_type,':',blocking_activity.wait_event) AS
blocking_wait_event,
        blocked_activity.state AS blocked_state,
        blocking_activity.state AS blocking_state,
        blocked_locks.locktype AS blocked_locktype,
        blocking_locks.locktype AS blocking_locktype,
        blocked_activity.query AS blocked_statement,
        blocking_activity.query AS blocking_statement
   FROM pg_catalog.pg_locks blocked_locks
   JOIN pg_catalog.pg_stat_activity blocked_activity ON blocked_activity.pid =
blocked_locks.pid
   JOIN pg_catalog.pg_locks blocking_locks
       ON blocking_locks.locktype = blocked_locks.locktype
       AND blocking_locks.DATABASE IS NOT DISTINCT FROM blocked_locks.DATABASE
       AND blocking_locks.relation IS NOT DISTINCT FROM blocked_locks.relation
       AND blocking_locks.page IS NOT DISTINCT FROM blocked_locks.page
       AND blocking_locks.tuple IS NOT DISTINCT FROM blocked_locks.tuple
       AND blocking_locks.virtualxid IS NOT DISTINCT FROM blocked_locks.virtualxid
       AND blocking_locks.transactionid IS NOT DISTINCT FROM
blocked_locks.transactionid
       AND blocking_locks.classid IS NOT DISTINCT FROM blocked_locks.classid
       AND blocking_locks.objid IS NOT DISTINCT FROM blocked_locks.objid
       AND blocking_locks.objsubid IS NOT DISTINCT FROM blocked_locks.objsubid
       AND blocking_locks.pid != blocked_locks.pid
   JOIN pg_catalog.pg_stat_activity blocking_activity ON blocking_activity.pid =
blocking_locks.pid
   WHERE NOT blocked_locks.GRANTED;
```

Riduci la concorrenza quando è alta

L'evento `Lock:tuple` potrebbe verificarsi costantemente, soprattutto in un tempo di carico di lavoro occupato. In questa situazione, si consideri di ridurre l'elevata concorrenza per le file molto occupate. Spesso, solo poche righe controllano una coda o la logica booleana, il che rende queste righe molto occupate.

È possibile ridurre la concorrenza utilizzando approcci diversi in base ai requisiti aziendali, alla logica dell'applicazione e al tipo di carico di lavoro. Ad esempio, puoi eseguire le operazioni seguenti:

- Riprogetta la tua tabella e la logica dei dati per ridurre la concorrenza elevata.
- Modificare la logica dell'applicazione per ridurre la concorrenza elevata a livello di riga.
- Sfrutta e riprogetta le query con i blocchi a livello di riga.
- Utilizzo della clausola `NOWAIT` con operazioni di riprova.
- Prendi in considerazione l'utilizzo di un controllo della concorrenza logico ottimistico e ibrido.
- Valuta la possibilità di modificare il livello di isolamento del database.

Risoluzione dei problemi dei colli di bottiglia

`Lock:tuple` può verificarsi con colli di bottiglia come la fame di CPU o il massimo utilizzo della larghezza di banda Amazon EBS. Per ridurre i colli di bottiglia, valuta i seguenti approcci:

- Ridimensiona il tipo di classe di istanza.
- Ottimizza le query a uso intensivo di risorse.
- Modificare la logica dell'applicazione.
- Archivia i dati a cui si accede raramente.

LWLock:BufferMapping (LWLock:buffer_mapping)

Questo evento si verifica quando un processo di backend è in attesa di associare un blocco di dati a un buffer nel pool di buffer condiviso.

> **Note**
>
> Questo evento è denominato `LWLock:BufferMapping` per RDS per PostgreSQL versione 13 e successive. Per RDS per PostgreSQL versione 12 e precedenti, questo evento è denominato `LWLock:buffer_mapping`.

Argomenti

Versioni del motore supportate

Queste informazioni relative all'evento di attesa sono supportate per RDS per PostgreSQL versione 9.6 e successive.

Context

Il pool buffer condiviso è un'area di memoria RDS per PostgreSQL che contiene tutte le pagine che sono o sono state utilizzate dai processi. Quando un processo ha bisogno di una pagina, legge la pagina nel buffer pool condiviso. Il parametro `shared_buffers` imposta le dimensioni del buffer condiviso e riserva un'area di memoria per memorizzare la tabella e le pagine indice. Se modifichi questo parametro, assicurati di riavviare il database.

L'evento di attesa `LWLock:buffer_mapping` si verifica nei seguenti scenari:

- Un processo ricerca nella tabella buffer una pagina e acquisisce un blocco di mappatura buffer condiviso.
- Un processo carica una pagina nel buffer pool e acquisisce un esclusivo blocco di mappatura del buffer.
- Un processo rimuove una pagina dal pool e acquisisce un blocco esclusivo di mappatura del buffer.

Cause

Quando questo evento appare più del normale, probabilmente indicando un problema di prestazioni, il database sta eseguendo il paging in entrata e in uscita dal buffer pool condiviso. Le cause tipiche sono:

- Query di grandi dimensioni
- Indici e tabelle gonfie
- Scansioni complete della tabella
- Dimensioni del pool condiviso più piccole del set di lavoro

Operazioni

Consigliamo azioni diverse a seconda delle cause dell'evento di attesa.

Argomenti

- Monitora i parametri relativi al buffer
- Valuta la tua strategia di indicizzazione
- Riduci il numero di buffer che devono essere allocati rapidamente

Monitora i parametri relativi al buffer

Quando `LWLock:buffer_mapping` aspetta il picco, indaga il rapporto di hit del buffer. È possibile utilizzare questi parametri per comprendere meglio cosa sta accadendo nella cache del buffer. Esamina i seguenti parametri:

`blks_hit`

Questo parametro contatore Performance Insights indica il numero di blocchi recuperati dal buffer pool condiviso. Dopo che appare l'evento di attesa `LWLock:buffer_mapping`, potresti osservare un picco in `blks_hit`.

`blks_read`

Questo parametro del contatore Performance Insights indica il numero di blocchi che richiedevano la lettura di I/O nel buffer pool condiviso. Potresti osservare un picco in `blks_read` in vista dell'evento di attesa `LWLock:buffer_mapping`.

Valuta la tua strategia di indicizzazione

Per confermare che la strategia di indicizzazione non sta peggiorando le prestazioni, verifica quanto segue:

Index bloat

Assicurati che il bloat di indice e tabella non porti alla lettura di pagine non necessarie nel buffer condiviso. Se le tabelle contengono righe inutilizzate, considera l'archiviazione dei dati e la rimozione delle righe dalle tabelle. È quindi possibile ricostruire gli indici per le tabelle ridimensionate.

Indici per query utilizzate di frequente

Per determinare se disponi degli indici ottimali, monitora i parametri del motore DB in Performance Insights. Il parametro `tup_returned` mostra il numero di righe lette. Il parametro `tup_fetched` mostra il numero di righe restituite al client. Se `tup_returned` è significativamente più grande di

`tup_fetched`, i dati potrebbero non essere indicizzati correttamente. Inoltre, le statistiche della tabella potrebbero non essere aggiornate.

Riduci il numero di buffer che devono essere allocati rapidamente

Per ridurre gli eventi di attesa `LWLock:buffer_mapping`, cercare di ridurre il numero di buffer che devono essere allocati rapidamente. Una strategia consiste nell'eseguire operazioni di batch di dimensioni ridotte. Potresti essere in grado di ottenere batch più piccoli partizionando le tabelle.

LWLock:BufferIO (IPC:BufferIO)

L'evento `LWLock:BufferIO` si verifica quando RDS per PostgreSQL è in attesa che altri processi finiscano le operazioni di input/output (I/O) quando si tenta contemporaneamente di accedere a una pagina. Il suo scopo è quello di leggere la stessa pagina nel buffer condiviso.

Argomenti

- Versioni di motori pertinenti
- Context
- Cause
- Operazioni

Versioni di motori pertinenti

Queste informazioni relative all'evento di attesa sono pertinenti per tutte le versioni di RDS per PostgreSQL. Per RDS per PostgreSQL 12 e versioni precedenti questo evento di attesa è denominato lwlock:buffer_io mentre in RDS per PostgreSQL versione 13 è denominato lwlock:bufferio. In RDS per PostgreSQL versione 14 l'evento di attesa BufferIO è stato spostato da `LWLock` al tipo di evento di attesa `IPC` (IPC:BufferIO).

Context

Ogni buffer condiviso ha un blocco I/O associato all'evento di attesa `LWLock:BufferIO`, ogni volta che un blocco (o una pagina) deve essere recuperato all'esterno del buffer pool condiviso.

Questo blocco viene utilizzato per gestire più sessioni che richiedono tutte l'accesso allo stesso blocco. Questo blocco deve essere letto dall'esterno del buffer pool condiviso, definito dal parametro `shared_buffers`.

Non appena la pagina viene letta all'interno del buffer pool condiviso, il blocco `LWLock:BufferIO` viene rilasciato.

> **Note**
>
> L'evento di attesa `LWLock:BufferIO` precede l'evento di attesa IO: DataFileRead. L'evento di attesa `IO:DataFileRead` si verifica mentre i dati vengono letti dallo storage.

Per ulteriori informazioni sui blocchi leggeri, consultaPanoramica dei blocchi.

Cause

Le cause comuni della comparsa dell'evento `LWLock:BufferIO` che appare nelle prime attese includono:

- Più backend o connessioni che tentano di accedere alla stessa pagina in attesa di un'operazione di I/O
- Il rapporto tra le dimensioni del buffer pool condiviso (definito dal parametro `shared_buffers`) e il numero di buffer necessari per il carico di lavoro corrente
- La dimensione del buffer pool condiviso non è ben bilanciata con il numero di pagine sfrattate da altre operazioni
- Indici grandi o gonfi che richiedono al motore di leggere più pagine del necessario nel buffer pool condiviso
- Mancanza di indici che costringe il motore DB a leggere più pagine dalle tabelle del necessario
- Checkpoint che si verificano troppo frequentemente o hanno bisogno di scaricare troppe pagine modificate
- Picchi improvvisi per le connessioni al database che tentano di eseguire operazioni sulla stessa pagina

Operazioni

Consigliamo azioni diverse a seconda delle cause dell'evento di attesa:

- Osservare i parametri di Amazon CloudWatch per la correlazione tra forti diminuzioni degli eventi di attesa `BufferCacheHitRatio` e `LWLock:BufferIO`. Questo effetto può significare che hai

una piccola impostazione dei buffer condivisi. Potrebbe essere necessario aumentarlo o scalare la classe di istanza DB. È possibile dividere il carico di lavoro in più nodi di lettore.

- Ottimizza `max_wal_size` e `checkpoint_timeout` in base al tempo di picco del carico di lavoro se vedi `LWLock:BufferIO` in coincidenza con cali dei parametri `BufferCacheHitRatio` Quindi identifica quale query potrebbe causarla.
- Verifica se hai indici inutilizzati, quindi rimuovili.
- Utilizzare tabelle partizionate (che hanno anche indici partizionati). Ciò aiuta a mantenere basso il riordino dell'indice e ne riduce l'impatto.
- Evitare di indicizzare inutilmente le colonne.
- Evita improvvisi picchi di connessione al database utilizzando un connection pool.
- Limitare il numero massimo di connessioni al database come best practice.

LWLock:buffer_content (BufferContent)

L'evento `LWLock:buffer_content` si verifica quando una sessione è in attesa di accedere in lettura o scrittura a una pagina dati in memoria mentre un'altra sessione ha bloccato la pagina in scrittura. In RDS per PostgreSQL 13 e versioni successive, questo evento di attesa viene chiamato `BufferContent`.

Argomenti

Versioni del motore supportate

Queste informazioni relative all'evento di attesa sono supportate per tutte le versioni di RDS per PostgreSQL.

Context

Per leggere o manipolare i dati, PostgreSQL vi accede tramite buffer di memoria condivisa. Per leggere dal buffer, un processo ottiene un blocco leggero (LWLock) sul contenuto del buffer in modalità condivisa. Per scrivere sul buffer, ottiene quel blocco in modalità esclusiva. I blocchi

condivisi consentono ad altri processi di acquisire contemporaneamente blocchi condivisi su quel contenuto. I blocchi esclusivi impediscono ad altri processi di ottenere qualsiasi tipo di blocco.

L'evento `LWLock:buffer_content` (`BufferContent`) indica che più processi stanno tentando di ottenere un blocco sul contenuto di un buffer specifico.

Probabili cause di aumento delle attese

Quando l'evento `LWLock:buffer_content` (`BufferContent`) si verifica più del normale, probabilmente indicando un problema di prestazioni, le cause tipiche includono le seguenti.

Aggiornamenti simultanei aumentati degli stessi dati

Potrebbe esserci un aumento del numero di sessioni simultanee con query che inseriscono o aggiornano la stessa tabella. Questa contesa può essere più marcata sulle tabelle con molti indici.

I dati del carico di lavoro non sono in memoria

Quando i dati elaborati dal carico di lavoro attivo non sono in memoria, questi eventi di attesa possono aumentare. Questo effetto è dovuto al fatto che i processi che contengono blocchi possono mantenerli più a lungo mentre eseguono operazioni di I/O su disco.

Uso eccessivo di vincoli di chiave esterna

I vincoli di chiave esterna possono aumentare la quantità di tempo che un processo mantiene su un blocco del contenuto del buffer. Questo effetto è dovuto al fatto che le operazioni di lettura richiedono un blocco del contenuto del buffer condiviso sulla chiave di riferimento mentre quella chiave viene aggiornata.

Operazioni

Consigliamo azioni diverse a seconda delle cause dell'evento di attesa. Potresti identificare eventi `LWLock:buffer_content` (`BufferContent`) utilizzando Amazon RDS Performance Insights o interrogando la vista `pg_stat_activity`.

Argomenti

- Migliora l'efficienza in memoria
- Riduzione dell'utilizzo di vincoli di chiave esterna
- Rimuovere gli indici inutilizzati
- Aumento della dimensione della cache quando si usano le sequenze

Migliora l'efficienza in memoria

Per aumentare la probabilità che i dati del carico di lavoro attivo si trovino in memoria, partizionare tabelle o scalare la classe di istanza. Per informazioni sulle classi di istanza database, consulta Classi di istanze database.

Riduzione dell'utilizzo di vincoli di chiave esterna

Indagare sui carichi di lavoro con un numero elevato di eventi di attesa `LWLock:buffer_content` (`BufferContent`) per l'utilizzo di vincoli di chiave esterna. Rimuovere i vincoli di chiave esterna non necessari.

Rimuovere gli indici inutilizzati

Per carichi di lavoro con un numero elevato di eventi di attesa `LWLock:buffer_content` (`BufferContent`), identificare gli indici inutilizzati e rimuoverli.

Aumento della dimensione della cache quando si usano le sequenze

Se le tabelle utilizzano sequenze, aumenta la dimensione della cache per rimuovere i conflitti nelle pagine di sequenza e nelle pagine di indice. Ogni sequenza è una singola pagina nella memoria condivisa. La cache predefinita è per ogni connessione. Potrebbe non essere sufficiente per gestire il carico di lavoro quando molte sessioni simultanee ricevono un valore di sequenza.

LWLock:lock_manager (LWLock:lockmanager)

Questo evento si verifica quando il motore RDS per PostgreSQL mantiene l'area di memoria del blocco condiviso per allocare, controllare e deallocare un blocco quando non è possibile un blocco rapido del percorso.

Argomenti

- Versioni del motore supportate
- Context
- Probabili cause di aumento delle attese
- Operazioni

Versioni del motore supportate

Queste informazioni relative all'evento di attesa sono supportate per RDS per PostgreSQL versione 9.6 e successive. Per le versioni di RDS per PostgreSQL precedenti alla 13, il nome di questo evento

di attesa è `LWLock:lock_manager`. Per RDS per PostgreSQL versione 13 e successive, il nome di questo evento di attesa è `LWLock:lockmanager`.

Context

Quando si emette un'istruzione SQL, RDS per PostgreSQL registra i blocchi per proteggere la struttura, i dati e l'integrità del database durante le operazioni simultanee. Il motore può raggiungere questo obiettivo utilizzando un blocco veloce del percorso o un blocco del percorso che non è veloce. Un blocco del percorso che non è veloce è più costoso e crea più sovraccarico di un blocco veloce del percorso.

Blocco rapido del percorso

Per ridurre il sovraccarico dei blocchi che vengono presi e rilasciati frequentemente, ma che raramente sono in conflitto, i processi di backend possono utilizzare il blocco rapido del percorso. Il database utilizza questo meccanismo per i blocchi che soddisfano i seguenti criteri:

- Usano il metodo di blocco DEFAULT.
- Rappresentano un blocco su una relazione di database anziché una relazione condivisa.
- Sono blocchi deboli che difficilmente entrino in conflitto.
- Il motore può verificare rapidamente che non siano presenti blocchi in conflitto.

Il motore non può utilizzare il blocco rapido del percorso quando una di queste condizioni è vera:

- Il blocco non soddisfa i criteri precedenti.
- Non sono disponibili più slot per il processo di backend.

Per ottimizzare le tue query per il blocco rapido del percorso, puoi utilizzare la seguente query.

```
SELECT count(*), pid, mode, fastpath
  FROM pg_locks
 WHERE fastpath IS NOT NULL
 GROUP BY 4,3,2
 ORDER BY pid, mode;
 count | pid  |      mode       | fastpath
-------+------+-----------------+----------
16 | 9185 | AccessShareLock | t
336 | 9185 | AccessShareLock | f
1 | 9185 | ExclusiveLock   | t
```

La seguente query mostra solo il totale per il database.

```
SELECT count(*), mode, fastpath
  FROM pg_locks
 WHERE fastpath IS NOT NULL
 GROUP BY 3,2
 ORDER BY mode,1;
count |      mode       | fastpath
-------+-----------------+----------
16 | AccessShareLock | t
337 | AccessShareLock | f
1 | ExclusiveLock   | t
(3 rows)
```

Per ulteriori informazioni sul blocco rapido del percorso, consulta percorso rapido nel gestore di blocco PostgreSQL README e pg-locks nella documentazione di PostgreSQL.

Esempio di problema di scalabilità per il blocco manager

In questo esempio, una tabella denominata `purchases` memorizza cinque anni di dati, partizionati per giorno. Ogni partizione ha due indici. Si verifica la seguente sequenza di eventi:

1. Si interrogano dati su diversi giorni, il che richiede al database di leggere molte partizioni.
2. Il database crea una voce di blocco per ogni partizione. Se gli indici di partizione fanno parte del percorso di accesso dell'ottimizzatore, il database crea anche una voce di blocco per loro.
3. Quando il numero di voci di blocco richieste per lo stesso processo di back-end è superiore a 16, ovvero il valore di `FP_LOCK_SLOTS_PER_BACKEND`, il gestore di blocco utilizza il metodo di blocco del percorso non veloce.

Le applicazioni moderne potrebbero avere centinaia di sessioni. Se le sessioni simultanee eseguono una query sul genitore senza un'adeguata scrematura delle partizioni, il database potrebbe creare centinaia o addirittura migliaia di blocchi di percorso non veloci. In genere, quando questa concorrenza è superiore al numero di vCPU, appare l'evento di attesa `LWLock:lock_manager`.

> **Note**
>
> L'evento di attesa `LWLock:lock_manager` non è correlato al numero di partizioni o indici in uno schema di database. Al contrario, è correlato al numero di blocchi di percorso non veloci che il database deve controllare.

Probabili cause di aumento delle attese

Quando l'evento di attesa `LWLock:lock_manager` si verifica più del normale, probabilmente indicando un problema di prestazioni, le cause più probabili di picchi improvvisi sono le seguenti:

- Le sessioni attive simultanee eseguono query che non utilizzano blocchi di percorso veloci. Queste sessioni superano anche la vCPU massima.
- Un gran numero di sessioni attive simultanee sta accedendo a una tabella fortemente partizionata. Ogni partizione ha più indici.
- Il database sta vivendo una tempesta di connessione. Per impostazione predefinita, alcune applicazioni e software del connection pool creano più connessioni quando il database è lento. Questa pratica peggiora il problema. Ottimizza il software del pool di connessioni in modo che non si verifichino tempeste di connessione.
- Un numero elevato di sessioni esegue una query su una tabella padre senza potare le partizioni.
- Un DDL (Data Definition Language), DML (Data Manipolation Language) o un comando di manutenzione blocca esclusivamente una relazione occupata o tuple a cui si accede frequentemente o modificate.

Operazioni

Se l'evento di attesa `CPU` si verifica, ciò non indica necessariamente un problema di prestazioni. Rispondi a questo evento solo quando le prestazioni diminuiscono e questo evento di attesa sta dominando il carico DB.

Argomenti

- Usare la potatura delle partizioni
- Rimuovere indici non necessari
- Ottimizza le tue query per bloccare rapidamente i percorsi
- Sintonizzati per altri eventi di attesa
- Riduzione dei colli di bottiglia hardware
- Utilizzare un connection pooler
- Aggiornamento della versione RDS per PostgreSQL

Usare la potatura delle partizioni

Potatura delle partizioni è una strategia di ottimizzazione delle query per le tabelle partizionate in modo dichiarativo che esclude le partizioni non necessarie dalle scansioni di tabelle, migliorando così le prestazioni. La potatura delle partizioni è attivata per impostazione predefinita. Se è spento, accenderlo come segue.

```
SET enable_partition_pruning = on;
```

Le query possono trarre vantaggio dalla potatura delle partizioni quando la clausola `WHERE` contiene la colonna utilizzata per il partizionamento. Per ulteriori informazioni, consulta Potatura delle partizioni nella documentazione di PostgreSQL.

Rimuovere indici non necessari

Il database potrebbe contenere indici inutilizzati o usati raramente. In tal caso, considera la possibilità di eliminarli. Eseguire una delle operazioni seguenti:

- Scopri come trovare indici non necessari consultanto Indici non utilizzati nel wiki di PostgreSQL.
- Esegui PG Collector. Questo script SQL raccoglie le informazioni del database e le presenta in un report HTML consolidato. Controlla la sezione «Indici non utilizzati». Per ulteriori informazioni, consulta pg-collector nel repository AWS Labs GitHub.

Ottimizza le tue query per bloccare rapidamente i percorsi

Per scoprire se le tue query utilizzano il blocco rapido dei percorsi, esegui una query nella colonna `fastpath` della tabella `pg_locks`. Se le query non utilizzano il blocco rapido dei percorsi, provare a ridurre il numero di relazioni per query a meno di 16.

Sintonizzati per altri eventi di attesa

Se `LWLock:lock_manager` è il primo o il secondo nell'elenco delle prime attese, controlla se nell'elenco vengono visualizzati anche i seguenti eventi di attesa:

- `Lock:Relation`
- `Lock:transactionid`
- `Lock:tuple`

Se gli eventi precedenti appaiono in alto nell'elenco, prendi in considerazione prima l'ottimizzazione di questi eventi di attesa. Questi eventi possono essere un driver per `LWLock:lock_manager`.

Riduzione dei colli di bottiglia hardware

Potresti avere un collo di bottiglia hardware, come la fame di CPU o il massimo utilizzo della larghezza di banda Amazon EBS. In questi casi, valuta la riduzione dei colli di bottiglia hardware. Prendi in considerazione le seguenti azioni:

- Ridimensiona la tua classe di istanza.
- Ottimizza le query che consumano grandi quantità di CPU e memoria.
- Cambia la logica dell'applicazione.
- Archivia i tuoi dati.

Per ulteriori informazioni su CPU, memoria e larghezza di banda di rete EBS, vedere Tipi di istanza Amazon RDS.

Utilizzare un connection pooler

Se il numero totale di connessioni attive supera la vCPU massima, più processi del sistema operativo richiedono CPU di quella supportata dal tipo di istanza. In questo caso, valuta l'utilizzo o l'ottimizzazione di un connection pool. Per ulteriori informazioni sul numero di vCPU per ogni tipo di istanza, consulta Tipi di istanza di Amazon RDS.

Per ulteriori informazioni sul connection pool, consulta le risorse seguenti:

- Utilizzo di Server proxy per Amazon RDS
- pgbouncer
- Connection pool e origini dati nella Documentazione di PostgreSQL

Aggiornamento della versione RDS per PostgreSQL

Se la versione attuale di RDS per PostgreSQL è precedente alla 12, esegui l'aggiornamento alla versione 12 o successiva. Le versioni 12 e successive di PostgreSQL hanno un meccanismo di partizione migliorato. Per ulteriori informazioni sui miglioramenti nella versioni 12, consulta PostgreSQL 12 Release Notes. Per ulteriori informazioni sull'aggiornamento di RDS per PostgreSQL, consulta Aggiornamento del motore del database PostgreSQL per Amazon RDS.

Timeout: PG Sleep

L'evento `Timeout:PgSleep` si verifica quando un processo server ha chiamato la funzione `pg_sleep` e sta aspettando la scadenza del timeout del sonno.

Argomenti

- Versioni del motore supportate
- Probabili cause di aumento delle attese
- Operazioni

Versioni del motore supportate

Queste informazioni relative all'evento di attesa sono supportate per tutte le versioni di RDS per PostgreSQL.

Probabili cause di aumento delle attese

Questo evento di attesa si verifica quando un'applicazione, una funzione memorizzata o un utente emette un'istruzione SQL che chiama una delle seguenti funzioni:

- `pg_sleep`
- `pg_sleep_for`
- `pg_sleep_until`

Le funzioni precedenti ritardano l'esecuzione fino a quando non è trascorso il numero di secondi specificato. Ad esempio: `SELECT pg_sleep(1)` si ferma per 1 secondo. Per ulteriori informazioni, consulta Ritardo dell'esecuzione nella documentazione di PostgreSQL.

Operazioni

Identificare la dichiarazione che stava eseguendo la funzione `pg_sleep`. Determina se l'uso della funzione è appropriato.

Timeout:VacuumDelay

L'evento `Timeout:VacuumDelay` indica che il limite dei costi per l'I/O vacuum è stato superato e che il processo di vacuum è stato interrotto. Le operazioni di vacuum si interrompono per la durata

specificata nel rispettivo parametro di ritardo dei costi, quindi riprendono a funzionare. Per il comando manuale di vacuum, il ritardo è specificato nel parametro `vacuum_cost_delay`. Per il daemon autovacuum, il ritardo è specificato nel `autovacuum_vacuum_cost_delay parameter.`

Argomenti

Versioni del motore supportate

Queste informazioni relative all'evento di attesa sono supportate per tutte le versioni di RDS per PostgreSQL.

Context

PostgreSQL ha sia un daemon autovacuum che un comando di vacuum manuale. Il processo di autovacuum è "attivato" per impostazione predefinita per le istanze database RDS per PostgreSQL. Il comando di vacuum manuale viene utilizzato in base alle necessità, ad esempio per eliminare le tabelle dalle tuple inattive o generare nuove statistiche.

Quando il processo di vacuum è in corso, PostgreSQL utilizza un contatore interno per tenere traccia dei costi stimati mentre il sistema esegue varie operazioni di I/O. Quando il contatore raggiunge il valore specificato dal parametro del limite dei costi, il processo che esegue l'operazione rimane inattivo per la breve durata specificata nel parametro del ritardo dei costi. Quindi ripristina il contatore e continua le operazioni.

Il processo di vacuum include dei parametri che possono essere utilizzati per regolare il consumo di risorse. Il vacuum automatico e il comando di vacuum manuale hanno i propri parametri per l'impostazione del valore limite dei costi. Hanno anche i propri parametri per specificare un ritardo dei costi, il tempo necessario per mettere il vacuum in sospensione quando viene raggiunto il limite. In questo modo, il parametro di ritardo dei costi funge da meccanismo di limitazione (della larghezza di banda della rete) del consumo di risorse. La descrizione di questi parametri è disponibile nell'elenco seguente.

Parametri che influiscono sulla limitazione (della larghezza di banda della rete) del daemon autovacuum

- `autovacuum_vacuum_cost_limit`: specifica il valore del limite dei costi da utilizzare nelle operazioni vacuum automatiche. L'aumento dell'impostazione per questo parametro consente al processo di vacuum di utilizzare più risorse e riduce l'evento di attesa `Timeout:VacuumDelay`.
- `autovacuum_vacuum_cost_delay`: specifica il valore di ritardo dei costi da utilizzare nelle operazioni vacuum automatiche. Il valore predefinito è 2 millisecondi. L'impostazione del parametro di ritardo su 0 disattiva il meccanismo di limitazione (della larghezza di banda della rete) e quindi l'evento di attesa `Timeout:VacuumDelay` non viene visualizzato.

Per ulteriori informazioni, consulta la pagina relativa al vacuum automatico nella documentazione di PostgreSQL.

Parametri che influiscono sulla limitazione (della larghezza di banda della rete) del processo di vacuum manuale

- `vacuum_cost_limit`: la soglia di interruzione del processo di vacuum. Il limite predefinito è 200. Questo numero rappresenta le stime dei costi accumulati per le operazioni I/O aggiuntive necessarie a varie risorse. L'aumento di questo valore riduce il numero dell'evento di attesa `Timeout:VacuumDelay`.
- `vacuum_cost_delay`: il periodo di tempo in cui il processo di vacuum rimane inattivo quando viene raggiunto il limite dei costi del vacuum. L'impostazione predefinita è 0, a indicare che la funzionalità è disattivata. Puoi impostare questo parametro su un valore intero per specificare il numero di millisecondi per attivare questa funzionalità, ma ti consigliamo di lasciare l'impostazione predefinita.

Per ulteriori informazioni sul parametro `vacuum_cost_delay`, consulta Resource Consumption (Consumo delle risorse) nella documentazione di PostgreSQL.

Per ulteriori informazioni su come configurare e usare la funzione di funzione vacuum automatica con RDS per PostgreSQL, consulta Utilizzo della funzione di autovacuum di PostgreSQL in Amazon RDS for PostgreSQL.

Probabili cause di aumento delle attese

`Timeout:VacuumDelay` è influenzato dall'equilibrio tra le impostazioni dei parametri del limite dei costi (`vacuum_cost_limit`, `autovacuum_vacuum_cost_limit`) e i parametri di ritardo

dei costi (`vacuum_cost_delay`, `autovacuum_vacuum_cost_delay`) che controllano la durata della sospensione del vacuum. L'aumento del valore del parametro del limite dei costi consente al vacuum di utilizzare più risorse prima di sospenderlo. Ciò si traduce in un minor numero di eventi di attesa `Timeout:VacuumDelay`. L'aumento di uno dei parametri di ritardo fa sì che l'evento di attesa `Timeout:VacuumDelay` si verifichi più frequentemente e per periodi di tempo più lunghi.

L'impostazione del parametro `autovacuum_max_workers` può anche aumentare il numero di `Timeout:VacuumDelay`. Ogni processo aggiuntivo di worker vacuum automatico contribuisce al meccanismo interno del contatore e quindi il limite può essere raggiunto più rapidamente rispetto a un singolo processo di vacuum automatico. Se il limite dei costi viene raggiunto più rapidamente, il ritardo dei costi viene applicato più frequentemente, con conseguente aumento degli eventi di attesa `Timeout:VacuumDelay`. Per ulteriori informazioni, consultare autovacuum_max_worker nella documentazione di PostgreSQL.

Anche oggetti di grandi dimensioni, quelli di almeno 500 GB, generano questo evento di attesa perché il vacuum può impiegare del tempo per completare l'elaborazione di oggetti di grandi dimensioni.

Operazioni

Se le operazioni di vacuum vengono completate come previsto, non è necessaria alcuna correzione. In altre parole, questo evento di attesa non indica necessariamente un problema. Indica che il vacuum viene messo in sospensione per il periodo di tempo specificato nel parametro di ritardo in modo che le risorse possano essere applicate ad altri processi che devono essere completati.

Se si desidera che le operazioni di vacuum vengano completate più rapidamente, è possibile ridurre i parametri di ritardo. In questo modo si riduce il tempo di sospensione del vacuum.

Ottimizzazione di RDS per PostgreSQL con approfondimenti proattivi di Amazon DevOps Guru

Gli approfondimenti proattivi di DevOps Guru rilevano le condizioni problematiche nelle istanze database RDS per PostgreSQL e ti avvisa prima che si verifichino. Con DevOps Guru è possibile:

- Evitare molti problemi comuni relativi al database controllando la configurazione del database rispetto alle impostazioni consigliate comuni.
- Ricevere gli avvisi per le criticità relative al parco istanze che, se non controllate, possono portare a problemi più gravi in seguito.
- Ricevere gli avvisi per i nuovi problemi individuati.

Ogni approfondimento proattivo contiene un'analisi della causa del problema e i suggerimenti per le azioni correttive.

Argomenti

Il database ha una connessione di transazione inattiva da molto tempo

Una connessione al database è nello stato `idle in transaction` da più di 1800 secondi.

Argomenti

Versioni del motore supportate

Queste informazioni approfondite sono supportate per tutte le versioni di RDS per PostgreSQL.

Context

Una transazione nello stato `idle in transaction` può contenere blocchi che impediscono l'esecuzione di altre query. Può anche impedire al `VACUUM` (incluso l'autovacuum) di cancellare le righe inutilizzate, con conseguente aumento delle dimensioni dell'indice o della tabella o del wraparound dell'ID della transazione.

Probabili cause di questo problema

Una transazione avviata in una sessione interattiva con BEGIN o START TRANSACTION non è stata terminata utilizzando un comando COMMIT, ROLLBACK o END. Lo stato della transazione diventa pertanto `idle in transaction`.

Operazioni

Puoi individuare le transazioni inattive eseguendo la query `pg_stat_activity`.

Nel client SQL, esegui la query riportata di seguito per elencare tutte le connessioni nello stato `idle in transaction` e ordinarle in base alla durata:

```
SELECT now() - state_change as idle_in_transaction_duration, now() - xact_start as
 xact_duration,*
FROM  pg_stat_activity
WHERE state  = 'idle in transaction'
AND   xact_start is not null
ORDER BY 1 DESC;
```

Consigliamo azioni diverse a seconda delle cause degli approfondimenti.

Argomenti

- Terminare la transazione
- Interrompere la connessione
- Configurare il parametro idle_in_transaction_session_timeout
- Controllare lo stato di AUTOCOMMIT
- Controllare la logica delle transazioni nel codice dell'applicazione

Terminare la transazione

Quando si avvia una transazione in una sessione interattiva con BEGIN o START TRANSACTION, lo stato della transazione diventa `idle in transaction`. Rimane in questo stato finché non si termina la transazione con un comando COMMIT, ROLLBACK, END o si disconnette completamente la connessione per eseguire il rollback della transazione.

Interrompere la connessione

Interrompi la connessione con una transazione inattiva utilizzando la seguente query:

```
SELECT pg_terminate_backend(pid);
```

pid è l'ID di processo della connessione.

Configurare il parametro idle_in_transaction_session_timeout

Configura il parametro `idle_in_transaction_session_timeout` nel gruppo di parametri. Il vantaggio della configurazione di questo parametro è che non richiede un intervento manuale per terminare la transazione inattiva da tempo. Per ulteriori informazioni, consulta la documentazione di PostgreSQL.

Il seguente messaggio verrà riportato nel file di log di PostgreSQL dopo l'interruzione della connessione, quando lo stato di una transazione è idle_in_transaction per un periodo superiore al tempo specificato.

```
FATAL: terminating connection due to idle in transaction timeout
```

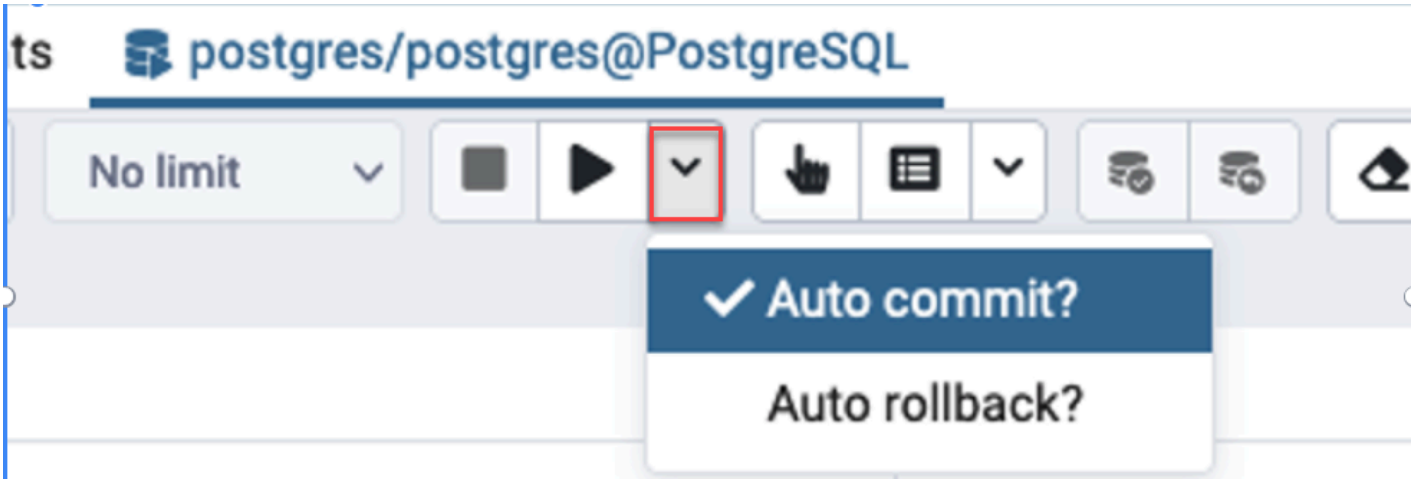
Controllare lo stato di AUTOCOMMIT

AUTOCOMMIT è attivato per impostazione predefinita. Tuttavia accidentalmente se viene disattivato nel client, assicurati di riattivarlo.

- Nel client psql, esegui il comando seguente:

  ```
  postgres=> \set AUTOCOMMIT on
  ```

- In pgadmin, attivalo scegliendo l'opzione AUTOCOMMIT selezionando la freccia rivolta verso il basso.



Controllare la logica delle transazioni nel codice dell'applicazione

Controlla la logica dell'applicazione per individuare eventuali problemi. Prendi in considerazione le seguenti azioni:

- Controlla se il commit automatico JDBC è impostato su true nell'applicazione. Inoltre, considera l'utilizzo di comandi `COMMIT` espliciti nel codice.
- Controlla la logica di gestione degli errori per vedere se chiude una transazione dopo gli errori.
- Controlla se l'applicazione impiega molto tempo per elaborare le righe restituite da una query mentre la transazione è aperta. In tal caso, valuta la possibilità di codificare l'applicazione per chiudere la transazione prima di elaborare le righe.
- Controlla se una transazione contiene molte operazioni di lunga durata. In tal caso, dividi la singola transazione in più transazioni.

Parametri rilevanti

A questo approfondimento sono correlati i seguenti parametri PI:

- idle_in_transaction_count - Numero di sessioni nello stato `idle in transaction`.
- idle_in_transaction_max_time - La durata della transazione in esecuzione più lunga nello stato `idle in transaction`.

Utilizzo delle estensioni PostgreSQL con Amazon RDS for PostgreSQL

È possibile estendere la funzionalità di PostgreSQL installando un'ampia serie di estensioni e moduli. Ad esempio, per lavorare con i dati spaziali è possibile installare e utilizzare l'estensione PostGIS. Per ulteriori informazioni, consulta Gestione dei dati spaziali con estensione PostGIS. Come altro esempio, per migliorare l'immissione dei dati per tabelle molto grandi, è possibile prendere in considerazione il partizionamento dei dati utilizzando l'estensione `pg_partman`. Per ulteriori informazioni, vedi Gestione delle partizioni PostgreSQL con l'estensione pg_partman.

> **Note**
>
> A partire da RDS per PostgreSQL 14.5, RDS per PostgreSQL supporta Trusted Language Extensions per PostgreSQL. Questa funzionalità è implementata come estensione `pg_tle`, che puoi aggiungere all'istanza database RDS per PostgreSQL. Con questa estensione, gli sviluppatori possono creare le proprie estensioni di PostgreSQL in un ambiente sicuro che semplifica i requisiti di impostazione e configurazione. Per ulteriori informazioni, consulta Utilizzo di Trusted Language Extensions per PostgreSQL.

In alcuni casi, anziché installare un'estensione, è possibile aggiungere un modulo specifico all'elenco di `shared_preload_libraries` nel gruppo di parametri database personalizzato dell'istanza database RDS per PostgreSQL. In genere, il gruppo di parametri cluster di database predefinito carica solo `pg_stat_statements`, ma sono disponibili diversi altri moduli da aggiungere all'elenco. Ad esempio, è possibile aggiungere funzionalità di pianificazione aggiungendo il modulo `pg_cron`, come descritto in Pianificazione della manutenzione con l'estensione PostgreSQL pg_cron. Come altro esempio, è possibile registrare i piani di esecuzione delle query caricando il modulo `auto_explain`. Per ulteriori informazioni, consulta Registrazione dei piani di esecuzione delle query nel AWS Knowledge Center.

A seconda della versione di RDS per PostgreSQL, l'installazione di un'estensione potrebbe richiedere autorizzazioni `rds_superuser`, come segue:

- Per RDS per PostgreSQL versione 12 e versioni precedenti, l'installazione delle estensioni richiede i privilegi `rds_superuser`.

- Per RDS per PostgreSQL versione 13 e versioni successive, gli utenti (ruoli) con autorizzazioni di creazione su una determinata istanza database possono installare e utilizzare qualsiasi estensione attendibile. Per un elenco di estensioni attendibili, consulta Estensioni attendibili di PostgreSQL.

È inoltre possibile specificare con precisione le estensioni che possono essere installate sull'istanza database RDS per PostgreSQL, elencandole nel parametro `rds.allowed_extensions`. Per ulteriori informazioni, consulta Limitazione dell'installazione delle estensioni PostgreSQL.

Per ulteriori informazioni sul ruolo `rds_superuser`, consulta Informazioni su ruoli e autorizzazioni di PostgreSQL.

Argomenti

- Utilizzo delle funzioni dall'estenzione orafce
- Gestione delle partizioni PostgreSQL con l'estensione pg_partman
- Utilizzo di pgAudit per registrare l'attività del database
- Pianificazione della manutenzione con l'estensione PostgreSQL pg_cron
- Utilizzo di pglogical per sincronizzare i dati tra le istanze
- Utilizzo di pgactive per supportare la replica active-active
- Riduzione della dimensione nelle tabelle e negli indici con l'estensione pg_repack
- Aggiornamento e utilizzo dell'estensione PLV8
- Utilizzo di PL/Rust per scrivere funzioni PostgreSQL nel linguaggio Rust
- Gestione dei dati spaziali con estensione PostGIS

Utilizzo delle funzioni dall'estenzione orafce

L'estensione orafce fornisce funzioni e operatori che emulano un sottoinsieme di funzioni e pacchetti da un database Oracle. L'estensione orafce consente di portare più facilmente un'applicazione Oracle su PostgreSQL. RDS for PostgreSQL versioni 9.6.6 e successive supportano questa estensione. Per ulteriori informazioni su orafce, vedere orafce on. GitHub

> **Note**
>
> RDS for PostgreSQL non supporta il pacchetto `utl_file`, che fa parte dell'estensione orafce. Ciò avviene perché le funzioni dello schema `utl_file` offrono operazioni di lettura e scrittura sui file di testo del sistema operativo, il che richiede che il

superuser acceda all'host sottostante. Come servizio gestito, RDS for PostgreSQL non fornisce accesso host.

Utilizzare l'estensione orafce

1. Connettiti all'istanza database con il nome utente primario che hai utilizzato per creare l'istanza database.

   Se si desidera attivare orafce per un database diverso nella stessa istanza database, utilizzare il comando psql `/c dbname`. Utilizzando questo comando, si passa dal database primario dopo aver avviato la connessione.
2. Attivare l'estensione orafce con l'istruzione `CREATE EXTENSION`.

   ```
   CREATE EXTENSION orafce;
   ```
3. Trasferire la proprietà dello schema oracle al ruolo rds_superuser con l'istruzione `ALTER SCHEMA`.

   ```
   ALTER SCHEMA oracle OWNER TO rds_superuser;
   ```

   Se si desidera visualizzare l'elenco dei proprietari per lo schema di Oracle, utilizzare il comando \dn psql.

Gestione delle partizioni PostgreSQL con l'estensione pg_partman

Il partizionamento delle tabelle PostgreSQL fornisce un framework per la gestione ad alte prestazioni di input e reporting dei dati. Utilizzare il partizionamento per database che richiedono un input molto veloce di grandi quantità di dati. Il partizionamento fornisce anche query di tabelle di grandi dimensioni più veloci. Il partizionamento consente di conservare i dati senza influire sull'istanza del database perché richiede meno risorse I/O.

Utilizzando il partizionamento, è possibile suddividere i dati in blocchi di dimensioni personalizzate per l'elaborazione. Ad esempio, è possibile partizionare i dati delle serie temporali per intervalli quali orario, giornaliero, settimanale, mensile, trimestrale, annuale, personalizzato o qualsiasi combinazione di questi. Per un esempio di dati di serie temporali, se la tabella è stata partizionata per ora, ogni partizione conterrà un'ora di dati. Se si partiziona la tabella delle serie temporali per giorno, le partizioni conterranno i dati di un giorno e così via. La chiave di partizione controlla le dimensioni di una partizione.

Quando si utilizza un comando SQL `INSERT` o `UPDATE` in una tabella partizionata, il motore database indirizza i dati alla partizione appropriata. Le partizioni di tabella PostgreSQL che memorizzano i dati sono tabelle figlio della tabella principale.

Durante le letture delle query di database, l'ottimizzatore PostgreSQL esamina la clausola `WHERE` della query e, se possibile, indirizza la scansione del database solo alle partizioni pertinenti.

A partire dalla versione 10, PostgreSQL utilizza il partizionamento dichiarativo per implementare il partizionamento delle tabelle. Questo è noto anche come partizionamento PostgreSQL nativo. Prima di PostgreSQL versione 10, per implementare le partizioni venivano utilizzati i trigger.

Il partizionamento delle tabelle PostgreSQL fornisce le seguenti funzionalità:

- Creazione di nuove partizioni in qualsiasi momento.
- Intervalli di partizione variabili.
- Partizioni scollegabili e ricollegabili utilizzando istruzioni DDL (Data Definition Language).

  Ad esempio, le partizioni scollegabili sono utili per rimuovere i dati storici dalla partizione principale, conservando i dati storici per l'analisi.
- Le nuove partizioni ereditano le proprietà della tabella di database padre, tra cui:
  - Indici
  - Chiavi primarie, che devono includere la colonna delle chiavi di partizione

  - Chiavi esterne
  - Vincoli check
  - Riferimenti
- Creazione di indici per la tabella completa o per ogni partizione specifica.

Non è possibile modificare lo schema per una singola partizione. Tuttavia, è possibile modificare la tabella padre (ad esempio, aggiungendo una nuova colonna), che si propaga alle partizioni.

Argomenti

Panoramica dell'estensione PostgreSQL pg_partman

È possibile utilizzare l'estensione `pg_partman` PostgreSQL per automatizzare la creazione e la manutenzione delle partizioni di tabella. Per informazioni più generali, consulta PG Partition Manager nella documentazione di `pg_partman`.

> **Note**
>
> L'estensione `pg_partman` è supportata sul motore Amazon RDS for PostgreSQL versioni 12.5 e successive.

Invece di dover creare manualmente ogni partizione, è possibile configurare `pg_partman` con le seguenti impostazioni:

- Tabella da partizionare
- Tipo di partizione
- Chiave di partizione
- Granularità delle partizioni
- Opzioni di pre-creazione e gestione delle partizioni

Dopo aver creato una tabella con partizioni PostgreSQL, la si registra con pg_partman chiamando la funzione create_parent. In questo modo vengono create le partizioni necessarie in base ai parametri passati alla funzione.

L'estensione pg_partman fornisce anche la funzione run_maintenance_proc, che è possibile chiamare su base pianificata per gestire automaticamente le partizioni. Per pianificare la creazione delle partizioni appropriate in base alle esigenze, puoi pianificare questa funzione in modo che venga eseguita periodicamente (ad esempio, ogni ora). È inoltre possibile assicurarsi che le partizioni vengano eliminate automaticamente.

Abilitazione dell'estensione pg_partman

Se disponi di più database all'interno della stessa istanza database per cui desideri gestire le partizioni, è necessario abilitare l'estensione pg_partman separatamente per ogni database. Per abilitare l'estensione pg_partman per un database specifico, crea lo schema di manutenzione delle partizioni, quindi crea l'estensione pg_partman nel modo seguente:

```
CREATE SCHEMA partman;
CREATE EXTENSION pg_partman WITH SCHEMA partman;
```

> **Note**
>
> Per creare l'estensione pg_partman, assicurati di disporre dei privilegi rds_superuser.

Se viene restituito un errore come il seguente, concedi i privilegi rds_superuser all'account o utilizza l'account utente avanzato.

```
ERROR: permission denied to create extension "pg_partman"
HINT: Must be superuser to create this extension.
```

Per concedere i privilegi rds_superuser, collegati con l'account utente avanzato ed emetti il seguente comando:

```
GRANT rds_superuser TO user-or-role;
```

Per gli esempi che mostrano l'uso dell'estensione pg_partman, si utilizza la tabella di database e la partizione di esempio seguenti. Questo database utilizza una tabella partizionata basata su un

timestamp. Uno schema `data_mart` contiene una tabella denominata `events` con una colonna denominata `created_at`. Nella tabella `events` sono incluse le seguenti impostazioni:

- Chiavi primarie `event_id` e `created_at`, che devono avere la colonna utilizzata per guidare la partizione.
- Un vincolo di controllo `ck_valid_operation` per applicare i valori per una colonna della tabella `operation` .
- Due chiavi esterne, dove una (`fk_orga_membership`)) punta alla tabella esterna `organization` e l'altra (`fk_parent_event_id`) è una chiave esterna autoreferenziata.
- Due indici, dove uno (`idx_org_id`) è per la chiave esterna e l'altro (`idx_event_type`) è per il tipo di evento.

Le seguenti istruzioni DDL creano questi oggetti, che verranno inclusi automaticamente in ogni partizione.

```
CREATE SCHEMA data_mart;
CREATE TABLE data_mart.organization ( org_id BIGSERIAL,
        org_name TEXT,
        CONSTRAINT pk_organization PRIMARY KEY (org_id)
    );

CREATE TABLE data_mart.events(
        event_id        BIGSERIAL,
        operation       CHAR(1),
        value           FLOAT(24),
        parent_event_id BIGINT,
        event_type      VARCHAR(25),
        org_id          BIGSERIAL,
        created_at      timestamp,
        CONSTRAINT pk_data_mart_event PRIMARY KEY (event_id, created_at),
        CONSTRAINT ck_valid_operation CHECK (operation = 'C' OR operation = 'D'),
        CONSTRAINT fk_orga_membership
            FOREIGN KEY(org_id)
            REFERENCES data_mart.organization (org_id),
        CONSTRAINT fk_parent_event_id
            FOREIGN KEY(parent_event_id, created_at)
            REFERENCES data_mart.events (event_id,created_at)
    ) PARTITION BY RANGE (created_at);

CREATE INDEX idx_org_id     ON  data_mart.events(org_id);
```

```
CREATE INDEX idx_event_type ON  data_mart.events(event_type);
```

Configurazione delle partizioni utilizzando la funzione create_parent

Dopo aver abilitato l'estensione `pg_partman`, utilizza la funzione `create_parent` per configurare le partizioni all'interno dello schema di manutenzione delle partizioni. In questo esempio viene utilizzato l'esempio della tabella `events` creato in Abilitazione dell'estensione pg_partman. Richiama la funzione `create_parent` come segue:

```
SELECT partman.create_parent( p_parent_table => 'data_mart.events',
 p_control => 'created_at',
 p_type => 'native',
 p_interval=> 'daily',
 p_premake => 30);
```

I parametri sono i seguenti:

- `p_parent_table` – La tabella partizionata padre. Questa tabella deve già esistere ed essere completa, deve ovvero includere lo schema.
- `p_control` – Colonna su cui basare il partizionamento. Il tipo di dati deve essere intero o basato sul tempo.
- `p_type`: il tipo è `'native'` o `'partman'`. In genere, è consigliabile utilizzare il tipo `native` per migliorare le prestazioni e la flessibilità. Il tipo `partman` si basa sull'ereditarietà.
- `p_interval` – Intervallo di tempo o intervallo intero per ogni partizione. I valori di esempio includono: `daily`, orario e così via.
- `p_premake` – Il numero di partizioni da creare in anticipo per supportare nuovi inserimenti.

Per una descrizione completa della funzione `create_parent`, consulta Funzioni di creazione nella documentazione `pg_partman`.

Configurazione della manutenzione delle partizioni utilizzando la funzione run_maintenance ance_proc

È possibile eseguire operazioni di manutenzione delle partizioni per creare automaticamente nuove partizioni, scollegare partizioni o rimuovere partizioni obsolete. La manutenzione delle partizioni si basa sulla funzione `run_maintenance_proc` dell'estensione `pg_partman` e dell'estensione

pg_cron, che avvia un pianificatore interno. Lo scheduler pg_cron esegue automaticamente le istruzioni SQL, le funzioni e le procedure definite nei database.

Nell'esempio seguente viene utilizzato l'esempio della tabella events creata in Abilitazione dell'estensione pg_partman per impostare l'esecuzione automatica delle operazioni di manutenzione delle partizioni. Come prerequisito, aggiungere pg_cron al parametro shared_preload_libraries nel gruppo di parametri dell'istanza database.

```
CREATE EXTENSION pg_cron;

UPDATE partman.part_config
SET infinite_time_partitions = true,
    retention = '3 months',
    retention_keep_table=true
WHERE parent_table = 'data_mart.events';
SELECT cron.schedule('@hourly', $$CALL partman.run_maintenance_proc()$$);
```

Di seguito, è riportata una spiegazione dettagliata dell'esempio precedente:

1. Modifica il gruppo di parametri associato all'istanza database e aggiungi pg_cron al valore del parametro shared_preload_libraries. Perché questa modifica abbia effetto, è necessario riavviare l'istanza database. Per ulteriori informazioni, consulta Modifica di parametri in un gruppo di parametri del database.
2. Emettere il comando CREATE EXTENSION pg_cron; utilizzando un account con le autorizzazioni rds_superuser. In questo modo, viene abilitata l'estensione pg_cron. Per ulteriori informazioni, consulta Pianificazione della manutenzione con l'estensione PostgreSQL pg_cron.
3. Emettere il comando UPDATE partman.part_config per regolare le impostazioni pg_partman per la tabella data_mart.events.
4. Eseguire il comando SET . . . per configurare la tabella data_mart.events, con le seguenti clausole:
   a. infinite_time_partitions = true, – Configura la tabella in modo da poter creare automaticamente nuove partizioni senza limiti.
   b. retention = '3 months', – Configura la tabella in modo che venga conservata per un massimo di tre mesi.

c. `retention_keep_table=true` – Configura la tabella in modo che quando il periodo di conservazione è scaduto, la tabella non venga eliminata automaticamente. Le partizioni precedenti al periodo di conservazione vengono invece scollegate dalla tabella padre.

5. Eseguire il comando `SELECT cron.schedule . . .` per creare una chiamata di funzione `pg_cron`. Questa chiamata definisce la frequenza con cui lo scheduler esegue la procedura di manutenzione `pg_partman`, `partman.run_maintenance_proc`. Per questo esempio, la procedura viene eseguita ogni ora.

Per una descrizione completa della funzione `run_maintenance_proc`, consulta Funzioni di manutenzione nella documentazione di `pg_partman`.

Utilizzo di pgAudit per registrare l'attività del database

Gli istituti finanziari, gli enti governativi e molti settori devono conservare i log di audit per soddisfare i requisiti normativi. L'utilizzo dell'estensione PostgreSQL Audit (pgAudit) con l'istanza database RDS per PostgreSQL consente di acquisire i record dettagliati richiesti in genere dai revisori o di soddisfare requisiti normativi. Ad esempio, è possibile impostare l'estensione pgAudit per tenere traccia delle modifiche apportate a database e tabelle specifici, per registrare l'utente che ha apportato la modifica e molti altri dettagli.

L'estensione pgAudit si basa sulla funzionalità dell'infrastruttura di registrazione PostgreSQL nativa estendendo i messaggi di registro con maggiori dettagli. In altre parole, l'approccio utilizzato per visualizzare il registro di audit è lo stesso utilizzato per visualizzare i messaggi di registro. Per ulteriori informazioni sulla registrazione PostgreSQL, consulta File di log del database RDS per PostgreSQL.

L'estensione pgAudit consente di oscurare i dati sensibili, come le password in chiaro, dai registri. Se il l'istanza database RDS per PostgreSQL è configurata per registrare le istruzioni DML (Data Manipulation Language) come descritto in Attivazione della registrazione delle query per l'istanza database RDS per PostgreSQL, il problema delle password in chiaro può essere evitato utilizzando l'estensione PostgreSQL Audit.

È possibile configurare l'audit sulle istanze database con un elevato grado di specificità. Puoi eseguire l'audit di tutti i database e di tutti gli utenti. In alternativa, è possibile scegliere di eseguire l'audit solo di determinati database, utenti e altri oggetti. È inoltre possibile escludere esplicitamente determinati utenti e database dall'audit. Per ulteriori informazioni, consulta Esclusione di utenti o database dalla registrazione di audit.

Data la quantità di dettagli che è possibile acquisire, ti consigliamo di monitorare il consumo di archiviazione se utilizzi pgAudit.

L'estensione pgAudit è supportata su tutte le Versioni RDS per PostgreSQL. Per un elenco delle versioni di pgAudit supportate dalle versioni di RDS per PostgreSQL disponibili, consulta Extension versions for Amazon RDS for PostgreSQL (Versioni delle estensioni per Amazon RDS per PostgreSQL) in Amazon RDS for PostgreSQL Release Notes (Note di rilascio di Amazon RDS per PostgreSQL).

Argomenti

- Configurazione dell'estensione pgAudit
- Audit di oggetti di database

- Esclusione di utenti o database dalla registrazione di audit
- Riferimento per l'estensione pgAudit

Configurazione dell'estensione pgAudit

Per configurare l'estensione pgAudit sull'istanza database RDS per PostgreSQL , aggiungi innanzitutto pgAudit alle librerie condivise nel gruppo di parametri database personalizzato per l'istanza database RDS per PostgreSQL Per informazioni sulla creazione di un gruppo di parametri database personalizzato, consulta Utilizzo di gruppi di parametri. Quindi, installa l'estensione pgAudit. Infine, specifica i database e gli oggetti di cui eseguire l'audit. Le procedure in questa sezione mostrano come fare. Puoi utilizzare la AWS Management Console o l'AWS CLI.

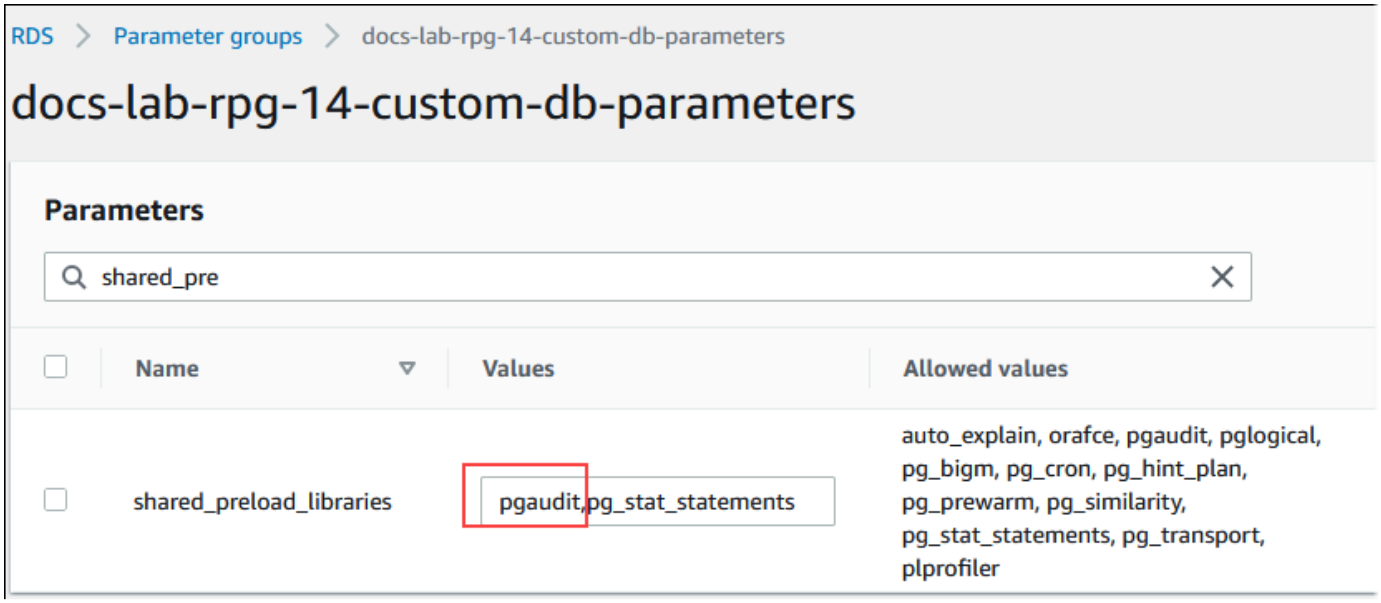
Per eseguire tutte queste attività, sono richieste autorizzazioni come il ruolo `rds_superuser`.

Le fasi seguenti si basano sull'ipotesi che l'istanza database RDS per PostgreSQL sia associata a un gruppo di parametri database personalizzato.

Console

Per impostare l'estensione pgAudit

1. Accedi alla AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, scegli l' istanza database RDS per PostgreSQL.
3. Apri la scheda Configurazione per l' l'istanza database RDS per PostgreSQL. Tra i dettagli dell'istanza, individua il collegamento Parameter group (Gruppo di parametri).
4. Scegli il collegamento per aprire i parametri personalizzati associati l'istanza database RDS per PostgreSQL.
5. Nel campo di ricerca Parametri, digita `shared_pre` per trovare il parametro `shared_preload_libraries`.
6. Scegli Edit parameters (Modifica parametri) per accedere ai valori delle proprietà.
7. Aggiungi `pgaudit` all'elenco nel campo Values (Valori). Utilizza una virgola per separare gli elementi nell'elenco di valori.



8. Riavvia l'istanza database RDS per PostgreSQL in modo che la modifica al parametro `shared_preload_libraries` diventi effettiva.
9. Quando l'istanza è disponibile, verifica che pgAudit sia stato inizializzato. Utilizza `psql` per connetterti all'istanza database RDS per PostgreSQL, quindi esegui il comando seguente.

```
SHOW shared_preload_libraries;
shared_preload_libraries
--------------------------
rdsutils,pgaudit
(1 row)
```

10. Con pgAudit inizializzato, puoi ora creare l'estensione. L'estensione deve essere creata dopo aver inizializzato la libreria perché l'estensione `pgaudit` installa i trigger evento per l'audit delle istruzioni DDL (Data Definition Language).

```
CREATE EXTENSION pgaudit;
```

11. Chiudi la sessione `psql`.

```
labdb=> \q
```

12. Accedi alla AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
13. Trova il parametro `pgaudit.log` nell'elenco e impostalo sul valore appropriato per il caso d'uso. Ad esempio, se si imposta il parametro `pgaudit.log` su `write` come mostrato nell'immagine

seguente, gli inserimenti, gli aggiornamenti, le eliminazioni e alcuni altri tipi di modifiche vengono acquisiti nel registro.

Puoi anche scegliere uno dei seguenti valori per il parametro `pgaudit.log`.

- none: valore predefinito. Non viene registrata alcuna modifica al database.
- all: registra tutto (read, write, function, role, ddl, misc).
- ddl: registra tutte le istruzioni DDL (Data Definition Language) non incluse nella classe `ROLE`.
- function: registra le chiamate di funzione e blocchi `DO`.
- misc: registra vari comandi come `DISCARD`, `FETCH`, `CHECKPOINT`, `VACUUM` e `SET`.
- read: registra `SELECT` e `COPY` quando l'origine è una relazione (ad esempio una tabella) o una query.
- role: registra le istruzioni correlate a ruoli e privilegi, ad esempio `GRANT`, `REVOKE`, `CREATE ROLE`, `ALTER ROLE` e `DROP ROLE`.
- write: registra `INSERT`, `UPDATE`, `DELETE`, `TRUNCATE` e `COPY` quando la destinazione è una relazione (tabella).

14. Seleziona Salvataggio delle modifiche.
15. Apri la console di Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
16. Scegli istanza database RDS per PostgreSQL dall'elenco di database per selezionarla, quindi scegli Reboot (Riavvia) dal menu Actions (Operazioni).

AWS CLI

Per configurare pgAudit

Per configurare pgAudit utilizzandoAWS CLI, si chiama l'modify-db-parameter-groupoperazione per modificare i parametri del registro di controllo nel gruppo di parametri personalizzato, come illustrato nella procedura seguente.

1. Utilizza il seguente comando AWS CLI per aggiungere `pgaudit` al parametro `shared_preload_libraries`.

```
aws rds modify-db-parameter-group \
   --db-parameter-group-name custom-param-group-name \
   --parameters
 "ParameterName=shared_preload_libraries,ParameterValue=pgaudit,ApplyMethod=pending-
reboot" \
   --region aws-region
```

2. Utilizza il comando AWS CLI seguente per riavviare istanza database RDS per PostgreSQL in modo che la libreria pgaudit venga inizializzata.

```
aws rds reboot-db-instance \
    --db-instance-identifier your-instance \
    --region aws-region
```

3. Quando l'istanza è disponibile, verifica che `pgaudit` sia stato inizializzato. Utilizza `psql` per connetterti all'istanza database RDS per PostgreSQL, quindi esegui il comando seguente.

```
SHOW shared_preload_libraries;
shared_preload_libraries
--------------------------
rdsutils,pgaudit
(1 row)
```

   Con pgAudit inizializzato, puoi ora creare l'estensione.

```
CREATE EXTENSION pgaudit;
```

4. Chiudi la sessione `psql` in modo da poter utilizzare AWS CLI.

```
labdb=> \q
```

5. Utilizza il comando AWS CLI seguente per specificare le classi di istruzioni che desideri registrare mediante la registrazione di audit della sessione. L'esempio imposta il parametro `pgaudit.log` su `write`, che acquisisce inserimenti, aggiornamenti ed eliminazioni nel registro.

```
aws rds modify-db-parameter-group \
   --db-parameter-group-name custom-param-group-name \
   --parameters
 "ParameterName=pgaudit.log,ParameterValue=write,ApplyMethod=pending-reboot" \
   --region aws-region
```

   Puoi anche scegliere uno dei seguenti valori per il parametro `pgaudit.log`.

   - none: valore predefinito. Non viene registrata alcuna modifica al database.
   - all: registra tutto (read, write, function, role, ddl, misc).
   - ddl: registra tutte le istruzioni DDL (Data Definition Language) non incluse nella classe `ROLE`.
   - function: registra le chiamate di funzione e blocchi `DO`.
   - misc: registra vari comandi come `DISCARD`, `FETCH`, `CHECKPOINT`, `VACUUM` e `SET`.
   - read: registra `SELECT` e `COPY` quando l'origine è una relazione (ad esempio una tabella) o una query.
   - role: registra le istruzioni correlate a ruoli e privilegi, ad esempio `GRANT`, `REVOKE`, `CREATE ROLE`, `ALTER ROLE` e `DROP ROLE`.
   - write: registra `INSERT`, `UPDATE`, `DELETE`, `TRUNCATE` e `COPY` quando la destinazione è una relazione (tabella).

   Riavvia l'istanza database RDS per PostgreSQL utilizzando il comando AWS CLI seguente.

```
aws rds reboot-db-instance \
    --db-instance-identifier your-instance \
    --region aws-region
```

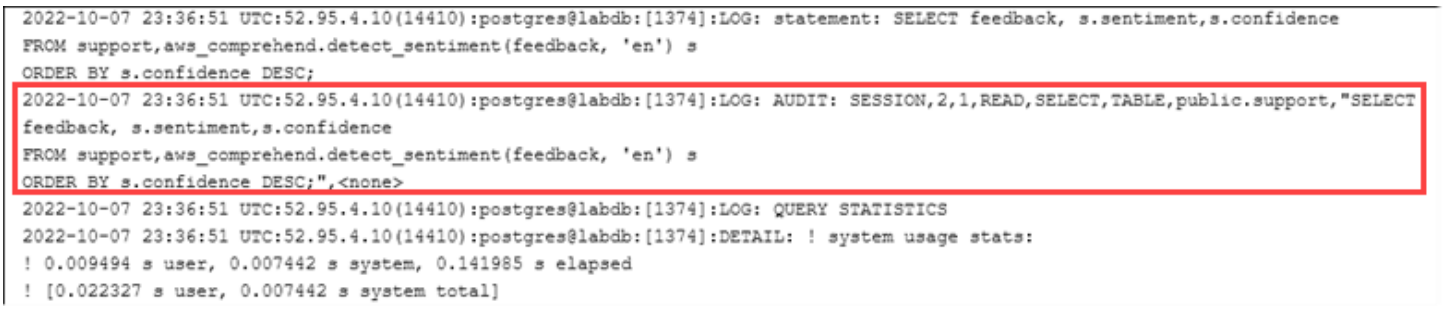
Audit di oggetti di database

Con pgAudit impostata sull'istanza database RDS per PostgreSQL e configurata per i requisiti, informazioni più dettagliate vengono acquisite nel registro PostgreSQL. Ad esempio, sebbene la configurazione della registrazione PostgreSQL predefinita consenta di identificare la data e l'ora della modifica apportata a una tabella di database, con l'estensione pgAudit la voce di registro

può includere lo schema, l'utente che ha apportato la modifica e altri dettagli a seconda della configurazione dei parametri dell'estensione. È possibile configurare l'audit per tenere traccia delle modifiche nei modi seguenti.

- Per ogni sessione, per utente. Per il livello di sessione, è possibile acquisire il testo del comando completo.
- Per ogni oggetto, per utente e per database.

La funzionalità di audit degli oggetti viene attivata quando il ruolo `rds_pgaudit` viene creato nel sistema e quindi aggiunto al parametro `pgaudit.role` nel gruppo di parametri personalizzati. Per impostazione predefinita, l'impostazione del parametro `pgaudit.role` viene annullata e l'unico valore consentito è `rds_pgaudit`. Nelle fasi seguenti si presume che `pgaudit` sia stato inizializzato e che l'estensione `pgaudit` sia stata creata seguendo la procedura in Configurazione dell'estensione pgAudit.



Come mostrato in questo esempio, la riga "LOG: AUDIT: SESSION" fornisce informazioni sulla tabella e il relativo schema, insieme ad altri dettagli.

Per configurare l'audit degli oggetti

1. Utilizza `psql` per connetterti all'istanza database RDS per PostgreSQL.

   ```
   psql --host=your-instance-name.aws-region.rds.amazonaws.com --port=5432 --
   username=postgrespostgres --password --dbname=labdb
   ```

2. Crea un ruolo del database denominato `rds_pgaudit` utilizzando il comando seguente.

   ```
   labdb=> CREATE ROLE rds_pgaudit;
   CREATE ROLE
   labdb=>
   ```

3. Chiudi la sessione `psql`.

```
labdb=> \q
```

Nei passaggi successivi, utilizza AWS CLI per modificare i parametri del registro di audit nel gruppo di parametri personalizzati.

4. Utilizza il seguente comando AWS CLI per impostare il parametro `pgaudit.role` su `rds_pgaudit`. Per impostazione predefinita, questo parametro è vuoto e `rds_pgaudit` è l'unico valore consentito.

```
aws rds modify-db-parameter-group \
   --db-parameter-group-name custom-param-group-name \
   --parameters
 "ParameterName=pgaudit.role,ParameterValue=rds_pgaudit,ApplyMethod=pending-reboot"
 \
   --region aws-region
```

5. Utilizza il seguente comando AWS CLI per riavviare l'istanza database RDS per PostgreSQL in modo da rendere effettive le modifiche apportate ai parametri.

```
aws rds reboot-db-instance \
    --db-instance-identifier your-instance \
    --region aws-region
```

6. Esegui il comando seguente per verificare che `pgaudit.role` sia impostato su `rds_pgaudit`.

```
SHOW pgaudit.role;
pgaudit.role
-----------------
rds_pgaudit
```

Per testare la registrazione di pgAudit, è possibile eseguire diversi comandi di esempio da controllare. Ad esempio, si potrebbero eseguire i comandi seguenti.

```
CREATE TABLE t1 (id int);
GRANT SELECT ON t1 TO rds_pgaudit;
SELECT * FROM t1;
id
----
(0 rows)
```

I registri del database devono contenere una voce simile alla seguente:

```
...
2017-06-12 19:09:49 UTC:...:rds_test@postgres:[11701]:LOG: AUDIT:
OBJECT,1,1,READ,SELECT,TABLE,public.t1,select * from t1;
...
```

Per informazioni sulla visualizzazione dei registri, consulta Monitoraggio dei file di log di Amazon RDS.

Per saperne di più sull'estensione pgAudit, vedi pgAudit su. GitHub

Esclusione di utenti o database dalla registrazione di audit

Come illustrato in File di log del database RDS per PostgreSQL, i registri di PostgreSQL consumano spazio di archiviazione. L'uso dell'estensione pgAudit consente di aumentare il volume di dati raccolti nei registri in misura diversa, a seconda delle modifiche che vengono monitorate. Potrebbe non essere necessario eseguire l'audit di ogni utente o database l'istanza database RDS per PostgreSQL.

Per ridurre al minimo l'impatto sull'archiviazione ed evitare di acquisire inutilmente i record di audit, è possibile escludere utenti e database dall'audit. È anche possibile modificare la registrazione all'interno di una determinata sessione. Negli esempi seguenti viene mostrato come fare.

> **Note**
>
> Le impostazioni dei parametri a livello di sessione hanno la precedenza sulle impostazioni nel gruppo di parametri database personalizzato per l'istanza database RDS per PostgreSQL. Per evitare che gli utenti del database ignorino le impostazioni di configurazione della registrazione di audit, accertati di modificare le loro autorizzazioni.

Supponi che l'istanza database RDS per PostgreSQL sia configurata per eseguire l'audit dello stesso livello di attività per tutti gli utenti e i database. Decidi quindi di non voler eseguire l'audit dell'utente `myuser`. Puoi disattivare l'audit per `myuser` con il comando SQL seguente.

```
ALTER USER myuser SET pgaudit.log TO 'NONE';
```

Quindi, puoi utilizzare la seguente query per controllare la colonna `user_specific_settings` per `pgaudit.log` per verificare che sia impostata su `NONE`.

```
SELECT
    usename AS user_name,
    useconfig AS user_specific_settings
FROM
    pg_user
WHERE
    usename = 'myuser';
```

Viene visualizzato l'output riportato di seguito.

```
 user_name | user_specific_settings
-----------+-----------------------
 myuser    | {pgaudit.log=NONE}
(1 row)
```

Puoi disattivare la registrazione per un determinato utente durante la sessione con il database con il seguente comando.

```
ALTER USER myuser IN DATABASE mydatabase SET pgaudit.log TO 'none';
```

Utilizza la seguente query per controllare la colonna delle impostazioni per pgaudit.log per una combinazione specifica di utente e database.

```
SELECT
    usename AS "user_name",
    datname AS "database_name",
    pg_catalog.array_to_string(setconfig, E'\n') AS "settings"
FROM
    pg_catalog.pg_db_role_setting s
    LEFT JOIN pg_catalog.pg_database d ON d.oid = setdatabase
    LEFT JOIN pg_catalog.pg_user r ON r.usesysid = setrole
WHERE
    usename = 'myuser'
    AND datname = 'mydatabase'
ORDER BY
    1,
    2;
```

L'output visualizzato è simile al seguente.

```
  user_name | database_name |     settings
```

```
-----------+---------------+------------------
 myuser    | mydatabase    | pgaudit.log=none
(1 row)
```

Dopo aver disattivato l'audit per `myuser`, decidi di non voler tenere traccia delle modifiche apportate a `mydatabase`. Puoi disattivare l'audit per il database specifico utilizzando il comando seguente.

```
ALTER DATABASE mydatabase SET pgaudit.log to 'NONE';
```

Quindi, utilizza la seguente query per controllare la colonna database_specific_settings per verificare che pgaudit.log sia impostato su NONE.

```
SELECT
a.datname AS database_name,
b.setconfig AS database_specific_settings
FROM
pg_database a
FULL JOIN pg_db_role_setting b ON a.oid = b.setdatabase
WHERE
a.datname = 'mydatabase';
```

Viene visualizzato l'output riportato di seguito.

```
 database_name | database_specific_settings
---------------+----------------------------
 mydatabase    | {pgaudit.log=NONE}
(1 row)
```

Per ripristinare le impostazioni predefinite di myuser, utilizza il comando seguente:

```
ALTER USER myuser RESET pgaudit.log;
```

Per ripristinare i valori predefiniti delle impostazioni di un database, utilizza il comando seguente.

```
ALTER DATABASE mydatabase RESET pgaudit.log;
```

Per ripristinare l'impostazione predefinita di utente e database, utilizza il comando seguente.

```
ALTER USER myuser IN DATABASE mydatabase RESET pgaudit.log;
```

Puoi anche acquisire eventi specifici nel registro impostando `pgaudit.log` su uno degli altri valori consentiti per il parametro `pgaudit.log`. Per ulteriori informazioni, consulta Elenco delle impostazioni consentite per il parametro `pgaudit.log`.

```
ALTER USER myuser SET pgaudit.log TO 'read';
ALTER DATABASE mydatabase SET pgaudit.log TO 'function';
ALTER USER myuser IN DATABASE mydatabase SET pgaudit.log TO 'read,function'
```

Riferimento per l'estensione pgAudit

Puoi specificare il livello di dettaglio desiderato per il registro di audit modificando uno o più dei parametri elencati in questa sezione.

Controllo del comportamento di pgAudit

Puoi controllare la registrazione di audit modificando uno o più dei parametri elencati nella tabella seguente.

Parametro	Descrizione
`pgaudit.log`	Specifica quali classi di istruzioni verranno registrate per registrazione di audit della sessione. I valori consentiti includono ddl, function, misc, read, role, write, none, all. Per ulteriori informazioni, consulta Elenco delle impostazioni consentite per il parametro `pgaudit.log` .
`pgaudit.log_catalog`	Quando è attivato (impostato su 1), aggiunge istruzioni all'audit trail se tutte le relazioni in un'istruzione si trovano in pg_catalog.
`pgaudit.log_level`	Specifica il livello di registro che verrà utilizzato per le voci di registro. Valori consentiti: debug5, debug4, debug3, debug2, debug1, info, notice, warning, log
`pgaudit.log_parameter`	Quando è attivato (impostato su 1), i parametri passati con l'istruzione vengono acquisiti nel registro di audit.
`pgaudit.log_relation`	Quando è attivato (impostato su 1), il registro di audit della sessione crea una voce di registro separata per ogni relazione

Parametro	Descrizione
	(TABLE, VIEW e così via) a cui si fa riferimento in un'istruzione SELECT o DML.
`pgaudit.log_statement_once`	Specifica se la registrazione includerà il testo dell'istruzione e i parametri con la prima voce di registro per una combinazione istruzione/istruzione secondaria o con ogni voce.
`pgaudit.role`	Specifica il ruolo principale da utilizzare per la registrazione di verifica degli oggetti. L'unica voce consentita è `rds_pgaudit` .

Elenco delle impostazioni consentite per il parametro **`pgaudit.log`**

Valore	Descrizione
nessuno	Questa è l'impostazione predefinita. Non viene registrata alcuna modifica al database.
tutto	Registra tutto (read, write, function, role, ddl, misc).
ddl	Registra tutte le istruzioni DDL (Data Definition Language) non incluse nella classe `ROLE`.
funzione	Registra le chiamate di funzione e i blocchi `DO`.
misc	Registra vari comandi, ad esempio `DISCARD`, `FETCH`, `CHECKPOINT` , `VACUUM` e `SET`.
read	Registra `SELECT` e `COPY` quando l'origine è una relazione (ad esempio una tabella) o una query.
role	Registra le istruzioni relative a ruoli e privilegi, ad esempio `GRANT`, `REVOKE`, `CREATE ROLE`, `ALTER ROLE` e `DROP ROLE`.
write	Registra `INSERT`, `UPDATE`, `DELETE`, `TRUNCATE` e `COPY` quando la destinazione è una relazione (tabella).

Per registrare più tipi di eventi con controllo di sessioni, utilizzare un elenco separato da virgole. Per registrare tutti i tipi di eventi, impostare `pgaudit.log` su `ALL`. Riavviare l'istanza database per applicare le modifiche.

Con il controllo dell'oggetto, è possibile perfezionare la registrazione di controllo per lavorare con relazioni specifiche. Ad esempio, è possibile richiedere la registrazione di audit per le operazioni `READ` su una o più tabelle.

Pianificazione della manutenzione con l'estensione PostgreSQL pg_cron

Puoi utilizzare l'estensione PostgreSQL `pg_cron` per pianificare i comandi di manutenzione all'interno di un database PostgreSQL. Per ulteriori informazioni sull'estensione, consulta Che cos'è pg_cron? nella documentazione di pg_cron.

L'estensione `pg_cron` è supportata sul motore RDS for PostgreSQL versioni 12.5 e successive.

Per ulteriori informazioni sull'uso di `pg_cron`, consulta Schedule jobs with pg_cron on your RDS for PostgreSQL or your Aurora PostgreSQL-Compatible Edition databases (Pianificazione dei processi con pg_cron sui database RDS per PostgreSQL o Aurora edizione compatibile con PostgreSQL).

Argomenti

- Configurazione dell'estensione pg_cron
- Concessione delle autorizzazioni per utilizzare pg_cron agli utenti del database
- Programmazione di processi pg_cron
- Riferimento per l'estensione pg_cron

Configurazione dell'estensione pg_cron

Configura l'estensione `pg_cron` come riportato di seguito:

1. Modifica il gruppo di parametri personalizzati associato all'istanza database PostgreSQL aggiungendo `pg_cron` al valore del parametro `shared_preload_libraries`.
   - Se l'istanza database RDS per PostgreSQL utilizza il parametro `rds.allowed_extensions` per elencare in maniera esplicita le estensioni che possono essere installate, è necessario aggiungere l'estensione `pg_cron` all'elenco. Solo alcune versioni di RDS per PostgreSQL supportano il parametro `rds.allowed_extensions`. Per impostazione predefinita, tutte le estensioni disponibili sono consentite. Per ulteriori informazioni, consulta Limitazione dell'installazione delle estensioni PostgreSQL.

   Per rendere effettive le modifiche apportate al gruppo di parametri, riavvia l'istanza database PostgreSQL. Per ulteriori informazioni sull'utilizzo dei gruppi di parametri, consulta Modifica di parametri in un gruppo di parametri del database.
2. Dopo il riavvio dell'istanza database PostgreSQL, esegui il comando riportato di seguito utilizzando un account che dispone delle autorizzazioni `rds_superuser`. Ad esempio, se hai utilizzato le

impostazioni di default quando hai creato l'istanza database RDS for PostgreSQL, connettiti come utente `postgres` e crea l'estensione.

```
CREATE EXTENSION pg_cron;
```

Lo scheduler `pg_cron` è impostato nel database PostgreSQL predefinito denominato `postgres`. Gli oggetti `pg_cron` vengono creati in questo database `postgres` e tutte le azioni di pianificazione vengono eseguite in questo database.

3. Puoi usare le impostazioni predefinite oppure puoi pianificare i processi da eseguire in altri database all'interno dell'istanza database di PostgreSQL. Per pianificare i processi per altri database all'interno dell'istanza database di PostgreSQL, consulta l'esempio in Pianificazione di un processo cron per un database diverso da quello predefinito.

Concessione delle autorizzazioni per utilizzare pg_cron agli utenti del database

L'installazione dell'estensione `pg_cron` richiede privilegi `rds_superuser`. Tuttavia, le autorizzazioni per utilizzare `pg_cron` possono essere concesse (da un membro del gruppo/ruolo `rds_superuser`) ad altri utenti del database, in modo che possano pianificare i propri lavori. Ti consigliamo di concedere le autorizzazioni allo schema `cron` solo se in base alle esigenze se migliora le operazioni nell'ambiente di produzione.

Per concedere a un database l'autorizzazione utente nello schema `cron`, esegui il seguente comando:

```
postgres=> GRANT USAGE ON SCHEMA cron TO db-user;
```

Questo fornisce l'autorizzazione *`db-user`* per accedere allo schema `cron` per pianificare i processi cron per gli oggetti per i quali si dispone di autorizzazione di accesso. Se l'utente del database non dispone di autorizzazioni, il processo non va a buon fine dopo aver pubblicato il messaggio di errore nel file `postgresql.log`, come mostrato di seguito:

```
2020-12-08 16:41:00 UTC::@:[30647]:ERROR: permission denied for table table-name
2020-12-08 16:41:00 UTC::@:[27071]:LOG: background worker "pg_cron" (PID 30647) exited
 with exit code 1
```

In altre parole, assicurati che gli utenti del database a cui sono concesse le autorizzazioni sullo `cron` schema dispongano anche delle autorizzazioni per gli oggetti (tabelle, schemi e così via) che intendono pianificare.

Nella tabella vengono inoltre registrati i dettagli del cron job e del suo esito positivo o negativo. `cron.job_run_details` Per ulteriori informazioni, consulta Tabelle per la pianificazione dei processi e l'acquisizione dello stato .

Programmazione di processi pg_cron

Nelle sezioni seguenti viene illustrato come è possibile pianificare varie attività di gestione utilizzando le attività `pg_cron`.

> **Note**
>
> Quando crei processi `pg_cron`, controlla che l'impostazione `max_worker_processes` sia maggiore del numero di `cron.max_running_jobs`. Un processo `pg_cron` non riesce se i processi worker in background vengono esauriti. Il numero predefinito di processi `pg_cron` è 5. Per ulteriori informazioni, consulta Parametri per la gestione dell'estensione pg_cron.

Argomenti

- Vacuum di una tabella
- Eliminazione della tabella della cronologia di pg_cron
- Registrazione degli errori solo nel file postgresql.log
- Pianificazione di un processo cron per un database diverso da quello predefinito

Vacuum di una tabella

Autovacuum gestisce la manutenzione del vacuum per la maggior parte dei casi. Tuttavia, potresti voler programmare un vacuum di una tabella specifica in un momento di tua scelta.

Consulta anche, Utilizzo della funzione di autovacuum di PostgreSQL in Amazon RDS for PostgreSQL.

Di seguito è riportato un esempio di utilizzo della funzione `cron.schedule` per impostare un processo in modo da utilizzare `VACUUM FREEZE` su una tabella specifica ogni giorno alle 22:00 (GMT).

```
SELECT cron.schedule('manual vacuum', '0 22 * * *', 'VACUUM FREEZE pgbench_accounts');
 schedule
----------
```

```
1
(1 row)
```

Dopo l'esecuzione dell'esempio precedente, è possibile controllare la cronologia nella tabella `cron.job_run_details` come riportato di seguito.

```
postgres=> SELECT * FROM cron.job_run_details;
jobid  | runid | job_pid | database | username | command                        |
 status    | return_message | start_time                    | end_time
-------+-------+---------+----------+----------+--------------------------------
+-----------+---------------+-------------------------------
+-------------------------------
 1     | 1     | 3395    | postgres | adminuser| vacuum freeze pgbench_accounts
 | succeeded | VACUUM         | 2020-12-04 21:10:00.050386+00 | 2020-12-04
 21:10:00.072028+00
(1 row)
```

Di seguito è riportata un'interrogazione della `cron.job_run_details` tabella per vedere i job falliti.

```
postgres=> SELECT * FROM cron.job_run_details WHERE status = 'failed';
jobid | runid | job_pid | database | username | command                        | status
 | return_message                                | start_time                    |
 end_time
------+-------+---------+----------+----------+--------------------------------+--------
+-----------------------------------------------+-------------------------------
+-------------------------------
 5    | 4     | 30339   | postgres | adminuser| vacuum freeze pgbench_account | failed
 | ERROR: relation "pgbench_account" does not exist | 2020-12-04 21:48:00.015145+00 |
 2020-12-04 21:48:00.029567+00
(1 row)
```

Per ulteriori informazioni, consulta Tabelle per la pianificazione dei processi e l'acquisizione dello stato .

Eliminazione della tabella della cronologia di pg_cron

La tabella `cron.job_run_details` contiene una cronologia di processi cron che può crescere a dismisura nel tempo. Si consiglia di pianificare un processo che elimini questa tabella. Ad esempio, conservare una settimana di voci può essere sufficiente per la risoluzione dei problemi.

Nell'esempio seguente viene utilizzata la funzione cron.schedule per pianificare un processo che viene eseguito ogni giorno a mezzanotte per rimuovere la tabella `cron.job_run_details`. Il

processo conserva solo gli ultimi sette giorni. Utilizza l'account `rds_superuser` per pianificare il processo come riportato di seguito.

```
SELECT cron.schedule('0 0 * * *', $$DELETE
    FROM cron.job_run_details
    WHERE end_time < now() - interval '7 days'$$);
```

Per ulteriori informazioni, consulta Tabelle per la pianificazione dei processi e l'acquisizione dello stato .

Registrazione degli errori solo nel file postgresql.log

Per disattivare la scrittura nella tabella `cron.job_run_details`, modifica il gruppo di parametri associato all'istanza database PostgreSQL e disattiva il parametro `cron.log_run`. L'estensione `pg_cron` non scrive più nella tabella e acquisisce gli errori solo nel file `postgresql.log`. Per ulteriori informazioni, consulta Modifica di parametri in un gruppo di parametri del database.

Utilizza il seguente comando per controllare il valore del parametro `cron.log_run`.

```
postgres=> SHOW cron.log_run;
```

Per ulteriori informazioni, consulta Parametri per la gestione dell'estensione pg_cron.

Pianificazione di un processo cron per un database diverso da quello predefinito

I metadati per `pg_cron` sono tutti contenuti nel database predefinito PostgreSQL denominato `postgres`. Poiché i worker in background vengono utilizzati per l'esecuzione dei processi cron di manutenzione, puoi pianificare un processo in uno qualsiasi dei database all'interno dell'istanza database PostgreSQL.

1. Nel database cron, pianifica il processo come si farebbe normalmente utilizzando cron.schedule.

   ```
   postgres=> SELECT cron.schedule('database1 manual vacuum', '29 03 * * *', 'vacuum
    freeze test_table');
   ```

2. Con il ruolo `rds_superuser`, aggiorna la colonna del database per il processo appena creato in modo che venga eseguito in un altro database all'interno dell'istanza database PostgreSQL.

   ```
   postgres=> UPDATE cron.job SET database = 'database1' WHERE jobid = 106;
   ```

3. Verifica eseguendo una query sulla tabella `cron.job`.

```
postgres=> SELECT * FROM cron.job;
jobid | schedule    | command                          | nodename  | nodeport |
 database | username  | active | jobname
------+-------------+----------------------------------+-----------+----------
+----------+----------+--------+-----------------------
106   | 29 03 * * * | vacuum freeze test_table       | localhost | 8192     |
 database1| adminuser | t      | database1 manual vacuum
  1   | 59 23 * * * | vacuum freeze pgbench_accounts | localhost | 8192     |
 postgres | adminuser | t      | manual vacuum
(2 rows)
```

> **Note**
>
> In alcune situazioni, è possibile aggiungere un processo cron che si intende eseguire in un database diverso. In questi casi, il processo potrebbe essere eseguito nel database predefinito (`postgres`) prima di aggiornare la colonna del database corretta. Se il nome utente dispone delle autorizzazioni, il processo viene eseguito correttamente nel database predefinito.

Riferimento per l'estensione pg_cron

È possibile utilizzare i seguenti parametri, funzioni e tabelle con l'estensione `pg_cron`. Per ulteriori informazioni, consultare Che cos'è pg_cron? nella documentazione di pg_cron.

Argomenti

- Parametri per la gestione dell'estensione pg_cron
- Riferimento alla funzione: cron.schedule
- Riferimento alla funzione: cron.unschedule
- Tabelle per la pianificazione dei processi e l'acquisizione dello stato

Parametri per la gestione dell'estensione pg_cron

Di seguito è riportato l'elenco dei parametri che consentono di controllare il comportamento dell'estensione `pg_cron`.

Parametro	Descrizione
cron.database_name	Il database in cui vengono conservati i metadati pg_cron.
cron.host	Il nome host per connettersi a PostgressQL. Non è possibile modificare questo valore.
cron.log_run	Registra tutti i processi eseguiti nella tabella job_run_details . I valori sono on o off. Per ulteriori informazioni, consulta Tabelle per la pianificazione dei processi e l'acquisizione dello stato .
cron.log_statement	Registra tutte le istruzioni cron prima di eseguirle. I valori sono on o off.
cron.max_running_jobs	Numero massimo di processi che possono essere eseguiti contemporaneamente.
cron.use_background_workers	Utilizza i worker in background anziché le sessioni client. Non è possibile modificare questo valore.

Puoi utilizzare il comando SQL seguente per visualizzare questi parametri e i relativi valori.

```
postgres=> SELECT name, setting, short_desc FROM pg_settings WHERE name LIKE 'cron.%'
 ORDER BY name;
```

Riferimento alla funzione: cron.schedule

Questa funzione pianifica un processo cron. Il processo viene inizialmente pianificato nel database postgres predefinito. La funzione restituisce un valore bigint che rappresenta l'identificativo del processo. Per pianificare l'esecuzione di processi in altri database all'interno dell'istanza database di PostgreSQL, consulta l'esempio in Pianificazione di un processo cron per un database diverso da quello predefinito.

La funzione ha due diverse sintassi.

Sintassi

```
cron.schedule (job_name,
    schedule,
    command
);

cron.schedule (schedule,
    command
);
```

Parametri

Parametro	Descrizione
`job_name`	Il nome del processo cron.
`schedule`	Il testo che indica la pianificazione per il processo cron. Il formato è il formato cron standard.
`command`	Testo del comando da eseguire.

Esempi

```
postgres=> SELECT cron.schedule ('test','0 10 * * *', 'VACUUM pgbench_history');
 schedule
----------
      145
(1 row)

postgres=> SELECT cron.schedule ('0 15 * * *', 'VACUUM pgbench_accounts');
 schedule
----------
      146
(1 row)
```

Riferimento alla funzione: cron.unschedule

Questa funzione elimina un processo cron. Puoi specificare `job_name` o `job_id`. Una policy assicura che l'utente sia il proprietario e possa rimuovere la pianificazione per il processo. La funzione restituisce un valore Booleano che indica la riuscita o l'errore.

La funzione ha la seguente sintassi.

Sintassi

```
cron.unschedule (job_id);

cron.unschedule (job_name);
```

Parametri

Parametro	Descrizione
`job_id`	Un identificativo di processo restituito dalla funzione `cron.schedule` quando è stato pianificato il processo cron.
`job_name`	Il nome di un processo cron pianificato con la funzione `cron.schedule` .

Esempi

```
postgres=> SELECT cron.unschedule(108);
 unschedule
------------
 t
(1 row)

postgres=> SELECT cron.unschedule('test');
 unschedule
------------
 t
(1 row)
```

Tabelle per la pianificazione dei processi e l'acquisizione dello stato

Le seguenti tabelle vengono create e utilizzate per pianificare i processi cron e registrare il modo in cui i processi vengono completati.

Tabella	Descrizione
`cron.job`	Contiene i metadati relativi a ciascun processo pianificato. La maggior parte delle interazioni con questa tabella dovrebbe essere eseguita tramite le funzioni `cron.schedule` e `cron.unschedule` . **Important** Si sconsiglia di concedere privilegi di aggiornamento/inserimento direttamente a questa tabella. In questo modo, si consente all'utente di aggiornare la colonna `username` da eseguire come `rds-superuser` .
`cron.job_run_details`	Contiene informazioni cronologiche sulle esecuzioni precedenti dei processi pianificati. Ciò è utile per analizzare lo stato, i messaggi restituiti e l'ora di inizio e di fine dall'esecuzione del processo. **Note** Per evitare che questa tabella cresca in maniera indefinita, è necessario eliminarla su base regolare. Per un esempio, consulta Eliminazione della tabella della cronologia di pg_cron.

Utilizzo di pglogical per sincronizzare i dati tra le istanze

Tutte le versioni di RDS per PostgreSQL attualmente disponibili supportano l'estensione `pglogical`, che precede la funzionalità di replica logica funzionalmente simile introdotta nella versione 10 di PostgreSQL. Per ulteriori informazioni, consulta Esecuzione della replica logica per Amazon RDS for PostgreSQL.

L'estensione `pglogical` supporta la replica logica tra due o più istanze database RDS per PostgreSQL. Supporta anche la replica tra diverse versioni di PostgreSQL e tra database in esecuzione in istanze database RDS per PostgreSQL e cluster database Aurora PostgreSQL. L'estensione `pglogical` utilizza un modello publish-subscribe per replicare le modifiche apportate alle tabelle e ad altri oggetti, come le sequenze, da un publisher in un subscriber. Si basa su uno slot di replica per garantire la sincronizzazione delle modifiche da un nodo publisher a un nodo subscriber, definiti come indicato di seguito.

- Il nodo publisher è l'istanza database RDS per PostgreSQL che costituisce l'origine dei dati da replicare in altri nodi. Il nodo publisher definisce le tabelle da replicare in un set di pubblicazione.
- Il nodo subscriber è l'istanza database RDS per PostgreSQL che riceve gli aggiornamenti WAL dal publisher. Il subscriber crea una sottoscrizione per connettersi al publisher e ottenere i dati WAL decodificati e contemporaneamente nel nodo publisher viene creato lo slot di replica.

Di seguito sono riportati gli argomenti sull'impostazione dell'estensione `pglogical`.

Argomenti

Requisiti e limitazioni dell'estensione pglogical

Tutte le versioni attualmente disponibili di RDS per PostgreSQL supportano l'estensione `pglogical`.

Sia il nodo publisher che il nodo subscriber devono essere impostati per la replica logica.

Le tabelle che devono essere replicate dal subscriber nel publisher devono avere gli stessi nomi e lo stesso schema. Inoltre devono contenere le stesse colonne e le colonne devono utilizzare gli stessi tipi di dati. Le tabelle del publisher e del subscriber devono avere le stesse chiavi primarie. Si consiglia di utilizzare solo la CHIAVE PRIMARIA come vincolo univoco.

Le tabelle del nodo subscriber possono avere vincoli più permissivi rispetto ai vincoli CHECK e NOT NULL delle tabelle del nodo publisher.

L'estensione `pglogical` fornisce funzionalità, come la replica bidirezionale, che non sono supportate dalla funzionalità di replica logica integrata in PostgreSQL 10 e versioni successive. Per ulteriori informazioni, consulta PostgreSQL bi-directional replication using pglogical (Replica bidirezionale di PostgreSQL utilizzando pglogical).

Impostazione dell'estensione pglogical

Per impostare l'estensione `pglogical` per l'istanza database RDS per PostgreSQL, aggiungi `pglogical` alle librerie condivise nel gruppo di parametri database personalizzato per l'istanza database RDS per PostgreSQL. È inoltre necessario impostare il valore del parametro `rds.logical_replication` su `1` per attivare la decodifica logica. Infine, crei l'estensione nel database. Per queste attività puoi utilizzare la AWS Management Console o AWS CLI.

Per eseguire queste attività sono richieste le autorizzazioni del ruolo `rds_superuser`.

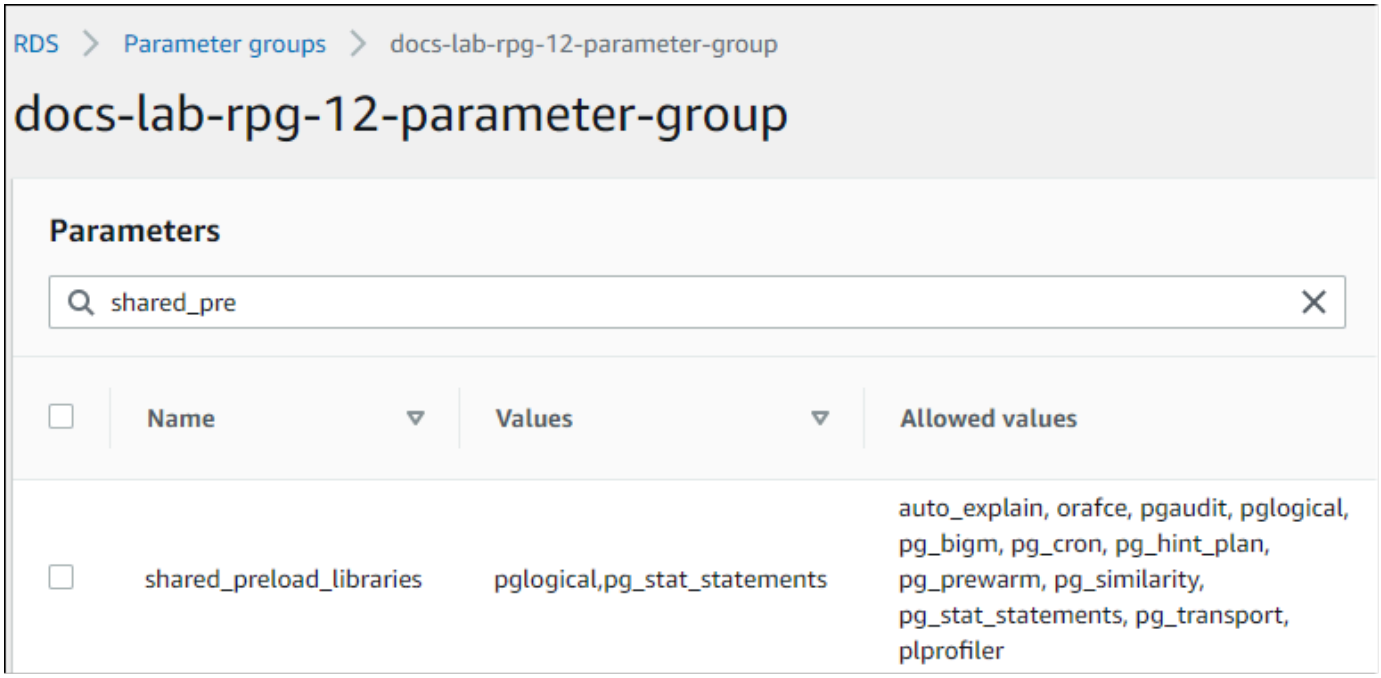
Le fasi seguenti si basano sull'ipotesi che l'istanza database RDS per PostgreSQL sia associata a un gruppo di parametri database personalizzato. Per informazioni sulla creazione di un gruppo di parametri database personalizzato, consulta Utilizzo di gruppi di parametri.

Console

Per impostare l'estensione pglogical

1. Accedi alla AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, scegli l' istanza database RDS per PostgreSQL.
3. Apri la scheda Configurazione per l' l'istanza database RDS per PostgreSQL. Tra i dettagli dell'istanza, individua il collegamento Parameter group (Gruppo di parametri).
4. Scegli il collegamento per aprire i parametri personalizzati associati l'istanza database RDS per PostgreSQL.

5. Nel campo di ricerca Parametri, digita `shared_pre` per trovare il parametro `shared_preload_libraries`.
6. Scegli Edit parameters (Modifica parametri) per accedere ai valori delle proprietà.
7. Aggiungi `pglogical` all'elenco nel campo Values (Valori). Utilizza una virgola per separare gli elementi nell'elenco di valori.



8. Individua il parametro `rds.logical_replication` e impostalo su `1` per attivare la replica logica.
9. Riavvia l'istanza database RDS per PostgreSQL per rendere effettive le modifiche.
10. Quando l'istanza è disponibile, puoi utilizzare `psql` (o pgAdmin) per connetterti all'istanza database RDS per PostgreSQL.

```
psql --host=111122223333.aws-region.rds.amazonaws.com --port=5432 --
username=postgres --password --dbname=labdb
```

11. Per verificare che l'estensione pglogical sia inizializzata, esegui il seguente comando.

```
SHOW shared_preload_libraries;
shared_preload_libraries
--------------------------
rdsutils,pglogical
(1 row)
```

12. Verifica l'impostazione che abilita la decodifica logica, come indicato di seguito.

```
SHOW wal_level;
wal_level
-----------
 logical
(1 row)
```

13. Crea l'estensione, come indicato di seguito.

```
CREATE EXTENSION pglogical;
EXTENSION CREATED
```

14. Seleziona Salvataggio delle modifiche.
15. Apri la console di Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
16. Scegli istanza database RDS per PostgreSQL dall'elenco di database per selezionarla, quindi scegli Reboot (Riavvia) dal menu Actions (Operazioni).

AWS CLI

Per impostare l'estensione pglogical

Per configurare pglogical utilizzandoAWS CLI, si chiama l'modify-db-parameter-groupoperazione per modificare determinati parametri nel gruppo di parametri personalizzato, come illustrato nella procedura seguente.

1. Utilizza il seguente comando AWS CLI per aggiungere `pglogical` al parametro `shared_preload_libraries`.

```
aws rds modify-db-parameter-group \
   --db-parameter-group-name custom-param-group-name \
   --parameters
 "ParameterName=shared_preload_libraries,ParameterValue=pglogical,ApplyMethod=pending-
reboot" \
   --region aws-region
```

2. Utilizza il seguente comando AWS CLI per impostare `rds.logical_replication` su 1 per attivare la funzionalità di decodifica logica per l'istanza database RDS per PostgreSQL.

```
aws rds modify-db-parameter-group \
   --db-parameter-group-name custom-param-group-name \
```

```
   --parameters
 "ParameterName=rds.logical_replication,ParameterValue=1,ApplyMethod=pending-
reboot" \
   --region aws-region
```

3. Utilizza il seguente comando AWS CLI per riavviare l'istanza database RDS per PostgreSQL in modo che la libreria pglogical venga inizializzata.

```
aws rds reboot-db-instance \
    --db-instance-identifier your-instance \
    --region aws-region
```

4. Quando l'istanza è disponibile, utilizza `psql` per connetterti all'istanza database RDS per PostgreSQL.

```
psql --host=111122223333.aws-region.rds.amazonaws.com --port=5432 --
username=postgres --password --dbname=labdb
```

5. Crea l'estensione, come indicato di seguito.

```
CREATE EXTENSION pglogical;
EXTENSION CREATED
```

6. Riavvia l'istanza database RDS per PostgreSQL utilizzando il comando AWS CLI seguente.

```
aws rds reboot-db-instance \
    --db-instance-identifier your-instance \
    --region aws-region
```

Impostazione della replica logica per l'istanza database RDS per PostgreSQL

La seguente procedura mostra come avviare la replica logica tra due istanze database RDS per PostgreSQL. I passaggi presuppongono che sia l'origine (publisher) che la destinazione (subscriber) abbiano l'estensione `pglogical` impostata come descritto dettagliatamente in Impostazione dell'estensione pglogical.

Per creare il nodo publisher e definire le tabelle da replicare

Questi passaggi presuppongono che l'istanza database RDS per PostgreSQL abbia un database contenente una o più tabelle che desideri replicare in un altro nodo. È necessario ricreare la struttura

delle tabelle dal publisher nel subscriber, quindi prima, se occorre, recupera la struttura delle tabelle. Puoi farlo utilizzando il metacomando `psql \d` *`tablename`* e quindi creando la stessa tabella nell'istanza subscriber. Nella procedura seguente viene illustrato come creare una tabella di esempio nel publisher (origine) a scopo dimostrativo.

1. Utilizza `psql` per connetterti all'istanza che include la tabella da usare come origine per i subscriber.

   ```
   psql --host=source-instance.aws-region.rds.amazonaws.com --port=5432 --
   username=postgres --password --dbname=labdb
   ```

   Se non hai una tabella esistente da replicare, puoi creare una tabella di esempio come indicato di seguito.

   a. Crea una tabella di esempio utilizzando la seguente istruzione SQL.

      ```
      CREATE TABLE docs_lab_table (a int PRIMARY KEY);
      ```

   b. Popola la tabella con i dati generati utilizzando la seguente istruzione SQL.

      ```
      INSERT INTO docs_lab_table VALUES (generate_series(1,5000));
      INSERT 0 5000
      ```

   c. Verifica che i dati siano presenti nella tabella utilizzando la seguente istruzione SQL.

      ```
      SELECT count(*) FROM docs_lab_table;
      ```

2. Identifica l'istanza database RDS per PostgreSQL come nodo publisher, come indicato di seguito.

   ```
   SELECT pglogical.create_node(
       node_name := 'docs_lab_provider',
       dsn := 'host=source-instance.aws-region.rds.amazonaws.com port=5432
    dbname=labdb');
    create_node
   -------------
      3410995529
   (1 row)
   ```

3. Aggiungi la tabella da replicare al set di replica predefinito. Per ulteriori informazioni sui set di replica, consulta Replication sets (Set di replica) nella documentazione di pglogical.

```
SELECT pglogical.replication_set_add_table('default', 'docs_lab_table', 'true',
 NULL, NULL);
 replication_set_add_table
  --------------------------
  t
  (1 row)
```

L'impostazione del nodo publisher è completata. Ora puoi impostare il nodo subscriber per ricevere gli aggiornamenti dal publisher.

Per impostare il nodo subscriber e creare una sottoscrizione per ricevere gli aggiornamenti

Questi passaggi presuppongono che sia stata eseguita l'impostazione dell'istanza database RDS per PostgreSQL con l'estensione `pglogical`. Per ulteriori informazioni, consulta Impostazione dell'estensione pglogical.

1. Utilizza `psql` per connetterti all'istanza per cui vuoi ricevere gli aggiornamenti dal publisher.

```
psql --host=target-instance.aws-region.rds.amazonaws.com --port=5432 --
username=postgres --password --dbname=labdb
```

2. Nell'istanza database RDS per PostgreSQL del subscriber crea la stessa tabella presente nel publisher. In questo esempio, la tabella è `docs_lab_table`. È possibile creare la tabella come indicato di seguito.

```
CREATE TABLE docs_lab_table (a int PRIMARY KEY);
```

3. Verifica che questa tabella sia vuota.

```
SELECT count(*) FROM docs_lab_table;
 count
-------
  0
(1 row)
```

4. Identifica l'istanza database RDS per PostgreSQL come nodo subscriber, come indicato di seguito.

```
SELECT pglogical.create_node(
```

```
    node_name := 'docs_lab_target',
    dsn := 'host=target-instance.aws-region.rds.amazonaws.com port=5432
 sslmode=require dbname=labdb user=postgres password=********');
 create_node
-------------
   2182738256
(1 row)
```

5. Crea la sottoscrizione.

```
SELECT pglogical.create_subscription(
   subscription_name := 'docs_lab_subscription',
   provider_dsn := 'host=source-instance.aws-region.rds.amazonaws.com port=5432
 sslmode=require dbname=labdb user=postgres password=*******',
   replication_sets := ARRAY['default'],
   synchronize_data := true,
   forward_origins := '{}' );
 create_subscription
---------------------
1038357190
(1 row)
```

Una volta completato questo passaggio, i dati della tabella del publisher vengono creati nella tabella del subscriber. È possibile verificare questa operazione utilizzando la seguente query SQL.

```
SELECT count(*) FROM docs_lab_table;
 count
-------
  5000
(1 row)
```

Da questo momento in poi, le modifiche apportate alla tabella del publisher vengono replicate nella tabella del subscriber.

Riconnessione della replica logica dopo un aggiornamento principale

Prima di poter eseguire un aggiornamento della versione principale di un'istanza database RDS per PostgreSQL impostata come nodo publisher per la replica logica, è necessario rimuovere tutti gli slot di replica, anche quelli non attivi. Si consiglia di deviare temporaneamente le transazioni del database

dal nodo publisher, rimuovere gli slot di replica, aggiornare l'istanza database RDS per PostgreSQL e quindi riconnettere e riavviare la replica.

Gli slot di replica sono ospitati solo nel nodo publisher. Il nodo subscriber RDS per PostgreSQL in uno scenario di replica logica non ha slot da rimuovere, ma non può essere aggiornato a una versione principale finché è designato come nodo subscriber con una sottoscrizione al publisher. Prima di aggiornare il nodo subscriber RDS per PostgreSQL, rimuovi la sottoscrizione e il nodo. Per ulteriori informazioni, consulta Gestione degli slot di replica logica per RDS per PostgreSQL.

Determinazione della replica logica interrotta

È possibile determinare che il processo di replica è stato interrotto eseguendo una query sul nodo publisher o sul nodo subscriber, come indicato di seguito.

Per controllare il nodo publisher

- Utilizza `psql` per connetterti al nodo publisher e quindi esegui la query sulla funzione `pg_replication_slots`. Osserva il valore nella colonna attiva. Normalmente, la query restituisce `t` (true) per indicare che la replica è attiva. Se restituisce `f` (false), indica che la replica nel subscriber è stata interrotta.

```
SELECT slot_name,plugin,slot_type,active FROM pg_replication_slots;
                    slot_name              |      plugin      | slot_type | active
-------------------------------------------+------------------+-----------+--------
 pgl_labdb_docs_labcb4fa94_docs_lab3de412c | pglogical_output | logical   | f
(1 row)
```

Per controllare il nodo subscriber

Nel nodo subscriber è possibile verificare lo stato della replica in tre modi diversi.

- Esamina i log di PostgreSQL sul nodo subscriber per trovare i messaggi di errore. Il log identifica gli errori nei messaggi che includono il codice di uscita 1, come illustrato di seguito.

```
2022-07-06 16:17:03 UTC::@:[7361]:LOG: background worker "pglogical apply
 16404:2880255011" (PID 14610) exited with exit code 1
2022-07-06 16:19:44 UTC::@:[7361]:LOG: background worker "pglogical apply
 16404:2880255011" (PID 21783) exited with exit code 1
```

- Esegui la query sulla funzione `pg_replication_origin`. Connettiti al database sul nodo subscriber utilizzando `psql` ed esegui la query sulla funzione `pg_replication_origin`, come indicato di seguito.

```
SELECT * FROM pg_replication_origin;
 roident | roname
---------+--------
(0 rows)
```

  Se il set di risultati è vuoto, la replica è stata interrotta. Normalmente, viene restituito l'output riportato di seguito.

```
   roident |                       roname
  ---------+-----------------------------------------------------
         1 | pgl_labdb_docs_labcb4fa94_docs_lab3de412c
  (1 row)
```

- Esegui la query sulla funzione `pglogical.show_subscription_status` come illustrato nell'esempio seguente.

```
SELECT subscription_name,status,slot_name FROM pglogical.show_subscription_status();
     subscription_name | status |              slot_name
---====----------------+--------+-----------------------------------
 docs_lab_subscription | down   | pgl_labdb_docs_labcb4fa94_docs_lab3de412c
(1 row)
```

  Questo output mostra che la replica è stata interrotta. Lo stato è `down` e normalmente l'output mostra lo stato `replicating`.

Se il processo di replica logica è stato interrotto, è possibile riconnettere la replica seguendo questi passaggi.

Per riconnettere la replica logica tra i nodi publisher e subscriber

Per riconnettere la replica, devi prima disconnettere il subscriber dal nodo publisher e quindi riconnettere la sottoscrizione, come descritto in questi passaggi.

1. Connettiti al nodo subscriber utilizzando `psql`, come indicato di seguito.

```
psql --host=222222222222.aws-region.rds.amazonaws.com --port=5432 --
username=postgres --password --dbname=labdb
```

2. Disattiva la sottoscrizione utilizzando la funzione `pglogical.alter_subscription_disable`.

```
SELECT pglogical.alter_subscription_disable('docs_lab_subscription',true);
 alter_subscription_disable
----------------------------
 t
(1 row)
```

3. Ottieni l'identificatore del nodo publisher eseguendo la query su `pg_replication_origin`, come indicato di seguito.

```
SELECT * FROM pg_replication_origin;
 roident |               roname
---------+-------------------------------------
       1 | pgl_labdb_docs_labcb4fa94_docs_lab3de412c
(1 row)
```

4. Utilizza la risposta restituita dal passaggio precedente con il comando `pg_replication_origin_create` per assegnare l'identificatore che può essere usato dalla sottoscrizione una volta riconnessa.

```
SELECT pg_replication_origin_create('pgl_labdb_docs_labcb4fa94_docs_lab3de412c');
  pg_replication_origin_create
------------------------------
                             1
(1 row)
```

5. Attiva la sottoscrizione specificando il nome con lo stato `true`, come illustrato nell'esempio seguente.

```
SELECT pglogical.alter_subscription_enable('docs_lab_subscription',true);
  alter_subscription_enable
---------------------------
 t
(1 row)
```

Controlla lo stato del nodo. Lo stato deve essere `replicating` come mostrato nell'esempio.

```
SELECT subscription_name,status,slot_name
  FROM pglogical.show_subscription_status();
            subscription_name |   status    |              slot_name
------------------------------+-------------+-----------------------------------
 docs_lab_subscription         | replicating |
 pgl_labdb_docs_lab98f517b_docs_lab3de412c
(1 row)
```

Verifica lo stato dello slot di replica del subscriber sul nodo publisher. La colonna `active` dello slot deve restituire `t` (true) per indicare che la replica è stata riconnessa.

```
SELECT slot_name,plugin,slot_type,active
  FROM pg_replication_slots;
                    slot_name              |      plugin      | slot_type | active
-------------------------------------------+------------------+-----------+--------
 pgl_labdb_docs_lab98f517b_docs_lab3de412c | pglogical_output | logical   | t
(1 row)
```

Gestione degli slot di replica logica per RDS per PostgreSQL

Prima di poter eseguire un aggiornamento alla versione principale su un'istanza database RDS per PostgreSQL che funge da nodo publisher in uno scenario di replica logica, è necessario rimuovere gli slot di replica nell'istanza. Il processo di verifica preliminare dell'aggiornamento alla versione principale avvisa che l'aggiornamento non può procedere fino a quando gli slot non vengono rimossi.

Per rimuovere gli slot dall'istanza database RDS per PostgreSQL, è necessario prima rimuovere la sottoscrizione e quindi gli slot.

Per identificare gli slot di replica creati utilizzando l'estensione `pglogical`, accedi a ciascun database e recupera il nome dei nodi. Quando esegui la query sul nodo subscriber, nell'output viene restituito sia il nodo publisher che il nodo subscriber, come mostrato nell'esempio seguente.

```
SELECT * FROM pglogical.node;
node_id   |     node_name
------------+-------------------
 2182738256 | docs_lab_target
 3410995529 | docs_lab_provider
(2 rows)
```

Puoi ottenere i dettagli sulla sottoscrizione con la seguente query.

```
SELECT sub_name,sub_slot_name,sub_target
  FROM pglogical.subscription;
 sub_name |         sub_slot_name          | sub_target
----------+--------------------------------+------------
  docs_lab_subscription     | pgl_labdb_docs_labcb4fa94_docs_lab3de412c | 2182738256
(1 row)
```

A questo punto puoi rimuovere la sottoscrizione, come indicato di seguito.

```
SELECT pglogical.drop_subscription(subscription_name := 'docs_lab_subscription');
 drop_subscription
-------------------
                 1
(1 row)
```

Dopo aver rimosso la sottoscrizione, puoi eliminare il nodo.

```
SELECT pglogical.drop_node(node_name := 'docs-lab-subscriber');
 drop_node
-----------
 t
(1 row)
```

Puoi verificare che il nodo sia stato eliminato, come indicato di seguito.

```
SELECT * FROM pglogical.node;
 node_id | node_name
---------+-----------
(0 rows)
```

Riferimento sui parametri dell'estensione pglogical

Nella tabella sono illustrati i parametri associati all'estensione `pglogical`. Parametri come `pglogical.conflict_log_level` e `pglogical.conflict_resolution` vengono utilizzati per gestire i conflitti di aggiornamento. I conflitti possono emergere quando vengono apportate modifiche localmente alle stesse tabelle che hanno una sottoscrizione con il publisher. I conflitti possono verificarsi anche in altri scenari, ad esempio la replica bidirezionale o quando più subscriber

eseguono la replica dallo stesso publisher. Per ulteriori informazioni, consulta PostgreSQL bi-directional replication using pglogical (Replica bidirezionale di PostgreSQL utilizzando pglogical).

Parametro	Descrizione
pglogical.batch_inserts	Esegue inserimenti batch, se possibile. Non impostato per impostazione predefinita. Imposta "1" per attivarlo, "0" per disattivarlo.
pglogical.conflict_log_level	Imposta il livello di log da utilizzare per la registrazione dei conflitti risolti. I valori di stringa supportati sono debug5, debug4, debug3, debug2, debug1, info, notice, warning, error, log, fatal, panic.
pglogical.conflict_resolution	Imposta il metodo da utilizzare per risolvere i conflitti quando sono risolvibili. I valori di stringa supportati sono error, apply_remote, keep_local, last_update_wins, first_update_wins.
pglogical.extra_connection_options	Specifica le opzioni di connessione da aggiungere a tutte le connessioni dei nodi peer.
pglogical.synchronous_commit	Valore di commit sincrono specifico pglogical
pglogical.use_spi	Utilizza la SPI (Server Programming Interface) invece dell'API di basso livello per applicare le modifiche. Imposta "1" per attivarlo , "0" per disattivarlo. Per ulteriori informazioni sulla SPI, consulta Server Programming Interface nella documentazione PostgreSQL.

Utilizzo di pgactive per supportare la replica active-active

L'estensione `pgactive` utilizza la replica active-active per supportare e coordinare le operazioni di scrittura su più database RDS per PostgreSQL. Amazon RDS for `pgactive` PostgreSQL supporta l'estensione nelle seguenti versioni:

- RDS per PostgreSQL 16.1 e versioni successive 16
- RDS per PostgreSQL 15.4-R2 e versioni successive 15
- RDS per PostgreSQL 14.10 e versioni successive (14)
- RDS per PostgreSQL 13.13 e versioni successive (13)
- RDS per PostgreSQL 12.17 e versioni successive (12)
- RDS per PostgreSQL 11.22

> **Note**
>
> Quando sono presenti operazioni di scrittura su più di un database in una configurazione di replica, sono possibili conflitti. Per ulteriori informazioni, consultare Gestione dei conflitti nella replica active-active

Argomenti

Inizializzazione della funzionalità di estensione pgactive

Per inizializzare la funzionalità dell'estensione `pgactive` sull'istanza database RDS per PostgreSQL, imposta il valore del parametro `rds.enable_pgactive` su `1` e quindi crea l'estensione nel

database. In questo modo si attivano automaticamente i parametri `rds.logical_replication` e `track_commit_timestamp` e il valore `wal_level` viene impostato su `logical`.

Per eseguire queste attività sono richieste le autorizzazioni del ruolo `rds_superuser`.

È possibile utilizzare AWS Management Console o the AWS CLI per creare l'RDS richiesto per le istanze DB PostgreSQL. I passaggi seguenti si basano sull'ipotesi che l'istanza database Amazon RDS per PostgreSQL sia associata a un gruppo di parametri di database personalizzato. Per ulteriori informazioni sulla creazione di un gruppo di parametri personalizzato, consulta Utilizzo di gruppi di parametri.

Console

Inizializzazione della funzionalità di estensione pgactive

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, scegli l'istanza database RDS per PostgreSQL.
3. Apri la scheda Configurazione per l'istanza database RDS per PostgreSQL. Nei dettagli dell'istanza, trova il link Gruppo di parametri dell'istanza database.
4. Scegli il link per aprire i parametri personalizzati associati all'istanza database RDS per PostgreSQL.
5. Trova il parametro `rds.enable_pgactive` e impostalo su 1 per inizializzare la funzionalità `pgactive`.
6. Seleziona Salvataggio delle modifiche.
7. Nel pannello di navigazione della console di Amazon RDS, scegli Database.
8. Seleziona l'istanza database RDS per PostgreSQL, quindi scegli Riavvia dal menu Operazioni.
9. Conferma il riavvio dell'istanza database per applicare le modifiche.
10. Quando l'istanza database è disponibile, puoi usare `psql` o qualsiasi altro client PostgreSQL per connetterti all'istanza database RDS per PostgreSQL.

    L'esempio seguente presuppone che l'istanza database RDS per PostgreSQL abbia un database predefinito denominato *`postgres`*.

    ```
    psql --host=mydb.111122223333.aws-region.rds.amazonaws.com --port=5432 --
    username=master username --password --dbname=postgres
    ```

11. Per verificare che l'estensione pgactive sia inizializzata, esegui il seguente comando.

```
postgres=>SELECT setting ~ 'pgactive'
FROM pg_catalog.pg_settings
WHERE name = 'shared_preload_libraries';
```

Se `pgactive` è presente in `shared_preload_libraries`, il comando precedente restituirà quanto segue:

```
?column?
----------
 t
```

12. Crea l'estensione, come indicato di seguito.

```
postgres=> CREATE EXTENSION pgactive;
```

AWS CLI

Inizializzazione della funzionalità di estensione pgactive

Per inizializzare l'`pgactive`utilizzo di AWS CLI, richiama l'modify-db-parameter-groupoperazione per modificare determinati parametri nel gruppo di parametri personalizzato, come illustrato nella procedura seguente.

1. Utilizzare il AWS CLI comando seguente `rds.enable_pgactive` per 1 impostare l'inizializzazione della `pgactive` funzionalità per l'istanza DB RDS for PostgreSQL.

```
postgres=>aws rds modify-db-parameter-group \
   --db-parameter-group-name custom-param-group-name \
   --parameters
 "ParameterName=rds.enable_pgactive,ParameterValue=1,ApplyMethod=pending-reboot" \
   --region aws-region
```

2. Utilizzare il AWS CLI comando seguente per riavviare l'istanza DB RDS for PostgreSQL in modo che la libreria venga inizializzata. `pgactive`

```
aws rds reboot-db-instance \
    --db-instance-identifier your-instance \
```

```
    --region aws-region
```

3. Quando l'istanza è disponibile, utilizza `psql` per connetterti all'istanza database RDS per PostgreSQL.

```
psql --host=mydb.111122223333.aws-region.rds.amazonaws.com --port=5432 --
username=master user --password --dbname=postgres
```

4. Crea l'estensione, come indicato di seguito.

```
postgres=> CREATE EXTENSION pgactive;
```

Configurazione della replica active-active per istanze database RDS per PostgreSQL

La seguente procedura mostra come avviare la replica active-active tra due istanze di database RDS per PostgreSQL che eseguono PostgreSQL 15.4 o versioni successive nella stessa regione. Per eseguire l'esempio di elevata disponibilità multiregionale, devi distribuire istanze Amazon RDS per PostgreSQL in due regioni diverse e configurare il peering VPC. Per ulteriori informazioni, consulta Peering VPC.

> **Note**
>
> L'invio di traffico tra più regioni può comportare costi aggiuntivi.

Questi passaggi presuppongono che l'istanza database RDS per PostgrSQl sia stata configurata con l'estensione `pgactive`. Per ulteriori informazioni, consulta Inizializzazione della funzionalità di estensione pgactive.

Configurazione della prima istanza database RDS per PostgreSQL con l'estensione **`pgactive`**

L'esempio seguente illustra come viene creato il gruppo `pgactive`, insieme ad altri passaggi necessari per creare l'estensione `pgactive`sull'istanza database RDS per PostgreSQL.

1. Usa `psql` o un altro strumento client per connetterti alla tua prima istanza database RDS per PostgreSQL.

```
psql --host=firstinstance.111122223333.aws-region.rds.amazonaws.com --port=5432 --
username=master username --password --dbname=postgres
```

2. Crea un database sull'istanza RDS per PostgreSQL utilizzando il seguente comando:

```
postgres=> CREATE DATABASE app;
```

3. Passa alla connessione al nuovo database utilizzando il seguente comando:

```
\c app
```

4. Per verificare se il parametro `shared_preload_libraries` contiene `pgactive`, esegui il comando seguente:

```
app=>SELECT setting ~ 'pgactive' FROM pg_catalog.pg_settings WHERE name =
 'shared_preload_libraries';
```

```
 ?column?
----------
 t
```

5. Crea e popola una tabella di esempio utilizzando le seguenti istruzioni SQL:

   a. Crea una tabella di esempio utilizzando la seguente istruzione SQL.

   ```
   app=> CREATE SCHEMA inventory;
   CREATE TABLE inventory.products (
   id int PRIMARY KEY, product_name text NOT NULL,
   created_at timestamptz NOT NULL DEFAULT CURRENT_TIMESTAMP);
   ```

   b. Popola la tabella con alcuni dati di esempio utilizzando la seguente istruzione SQL.

   ```
   app=> INSERT INTO inventory.products (id, product_name)
   VALUES (1, 'soap'), (2, 'shampoo'), (3, 'conditioner');
   ```

   c. Verifica che i dati siano presenti nella tabella utilizzando la seguente istruzione SQL.

   ```
    app=>SELECT count(*) FROM inventory.products;

    count
   -------
    3
   ```

6. Crea l'estensione `pgactive` database esistente.

```
app=> CREATE EXTENSION pgactive;
```

7. Crea e inizializza il gruppo pgactive usando i seguenti comandi:

```
app=> SELECT pgactive.pgactive_create_group(
    node_name := 'node1-app',
    node_dsn := 'dbname=app host=firstinstance.111122223333.aws-
region.rds.amazonaws.com user=master username password=PASSWORD');
```

node1-app è il nome che assegni per identificare in modo univoco un nodo nel gruppo `pgactive`.

> **Note**
>
> Per eseguire correttamente questo passaggio su un'istanza database accessibile pubblicamente, è necessario attivare il parametro `rds.custom_dns_resolution` impostandolo su 1.

8. Per verificare se l'istanza database è pronta, usa il seguente comando:

```
app=> SELECT pgactive.pgactive_wait_for_node_ready();
```

Se il comando viene eseguito correttamente, verrà visualizzato il seguente output:

```
pgactive_wait_for_node_ready
-----------------------------
(1 row)
```

Configurazione della seconda istanza RDS per PostgreSQL e collegamento al gruppo **`pgactive`**

L'esempio seguente illustra come puoi collegare un'istanza database RDS per PostgreSQL al gruppo `pgactive`, insieme ad altri passaggi necessari per creare l'estensione `pgactive` sull'istanza database.

Questi passaggi presuppongono che sia stata eseguita la configurazione di un ulteriore istanza database RDS per PostgreSQL con l'estensione `pgactive`. Per ulteriori informazioni, consulta Inizializzazione della funzionalità di estensione pgactive.

1. Utilizza `psql` per connetterti all'istanza per cui vuoi ricevere gli aggiornamenti dal publisher.

```
psql --host=secondinstance.111122223333.aws-region.rds.amazonaws.com --port=5432 --
username=master username --password --dbname=postgres
```

2. Crea un database sulla seconda istanza database RDS per PostgreSQL utilizzando il seguente comando:

```
postgres=> CREATE DATABASE app;
```

3. Passa alla connessione al nuovo database utilizzando il seguente comando:

```
\c app
```

4. Crea l'estensione `pgactive` database esistente.

```
app=> CREATE EXTENSION pgactive;
```

5. Unisci la seconda istanza database RDS per PostgreSQL al gruppo `pgactive` come segue.

```
app=> SELECT pgactive.pgactive_join_group(
node_name := 'node2-app',
node_dsn := 'dbname=app host=secondinstance.111122223333.aws-
region.rds.amazonaws.com user=master username password=PASSWORD',
join_using_dsn := 'dbname=app host=firstinstance.111122223333.aws-
region.rds.amazonaws.com user=postgres password=PASSWORD');
```

   node2-app è il nome che assegni per identificare in modo univoco un nodo nel gruppo `pgactive`.

6. Per verificare se l'istanza database è pronta, usa il seguente comando:

```
app=> SELECT pgactive.pgactive_wait_for_node_ready();
```

   Se il comando viene eseguito correttamente, verrà visualizzato il seguente output:

```
pgactive_wait_for_node_ready
-----------------------------
(1 row)
```

Se il primo database RDS per PostgreSQL è relativamente grande, puoi vedere l'emissione di `pgactive.pgactive_wait_for_node_ready()` del report sullo stato di avanzamento dell'operazione di ripristino. L'esito si presenta in maniera analoga all'immagine riportata di seguito.

```
NOTICE:  restoring database 'app', 6% of 7483 MB complete
NOTICE:  restoring database 'app', 42% of 7483 MB complete
NOTICE:  restoring database 'app', 77% of 7483 MB complete
NOTICE:  restoring database 'app', 98% of 7483 MB complete
NOTICE:  successfully restored database 'app' from node node1-app in
 00:04:12.274956
 pgactive_wait_for_node_ready
-----------------------------
(1 row)
```

Da questo momento in poi, `pgactive` sincronizza i dati tra le due istanze database.

7. Puoi utilizzare il comando seguente per verificare se il database della seconda istanza database contiene i dati:

```
app=> SELECT count(*) FROM inventory.products;
```

Se i dati vengono sincronizzati correttamente, verrà visualizzato il seguente output:

```
 count
-------
 3
```

8. Esegui il seguente comando per inserire nuovi valori:

```
app=> INSERT INTO inventory.products (id, product_name) VALUES ('lotion');
```

9. Connettiti al database della prima istanza database ed esegui la seguente query:

```
app=> SELECT count(*) FROM inventory.products;
```

Se la replica active-active è inizializzata, l'output è simile al seguente:

```
count
-------
 4
```

Scollegamento e rimozione di un'istanza database dal gruppo **`pgactive`**

Puoi scollegare e rimuovere un'istanza database dal gruppo `pgactive` utilizzando la procedura seguente:

1. Puoi scollegare la seconda istanza database dalla prima istanza database utilizzando il seguente comando:

   ```
   app=> SELECT * FROM pgactive.pgactive_detach_nodes(ARRAY['node2-app']);
   ```

2. Rimuovi l'estensione `pgactive` dalla seconda istanza database utilizzando il seguente comando:

   ```
   app=> SELECT * FROM pgactive.pgactive_remove();
   ```

   Per rimuovere forzatamente l'estensione:

   ```
   app=> SELECT * FROM pgactive.pgactive_remove(true);
   ```

3. Elimina l'estensione usando il seguente comando:

   ```
   app=> DROP EXTENSION pgactive;
   ```

Gestione dei conflitti nella replica active-active

L'estensione `pgactive` funziona per database e non per cluster. Ogni istanza database che utilizza `pgactive` è un'istanza indipendente e può accettare modifiche ai dati da qualsiasi origine. Quando viene inviata una modifica a un'istanza database, PostgreSQL la esegue localmente e quindi utilizza `pgactive` per replicare la modifica in modo asincrono su altre istanze database. Quando due istanze database PostgreSQL aggiornano lo stesso record quasi contemporaneamente, può verificarsi un conflitto.

L'estensione `pgactive` fornisce meccanismi per il rilevamento dei conflitti e la risoluzione automatica. Tiene traccia del timestamp in cui è stata effettuata la transazione su entrambe le istanze database e applica automaticamente la modifica con il timestamp più recente. L'estensione `pgactive` registra anche quando si verifica un conflitto nella tabella `pgactive.pgactive_conflict_history`.

Continueranno a crescere. `pgactive.pgactive_conflict_history` Potresti voler definire una politica di eliminazione. Questo può essere fatto cancellando alcuni record regolarmente o definendo uno schema di partizionamento per questa relazione (e successivamente staccando, eliminando e troncando le partizioni di interesse). Per implementare regolarmente la politica di eliminazione, un'opzione è utilizzare l'estensione. `pg_cron` Vedi le seguenti informazioni su un esempio per la tabella della `pg_cron` cronologia, Pianificazione della manutenzione con l'estensione PostgreSQL pg_cron.

Gestione delle sequenze nella replica active-active

Un'istanza database RDS per PostgreSQL con l'estensione `pgactive` utilizza due diversi meccanismi di sequenza per generare valori univoci.

Sequenze globali

Per utilizzare una sequenza globale, crea una sequenza locale con l'istruzione `CREATE SEQUENCE`. Utilizza `pgactive.pgactive_snowflake_id_nextval(seqname)` invece di `usingnextval(seqname)` per ottenere il valore univoco successivo della sequenza.

L'esempio seguente crea una sequenza globale:

```
postgres=> CREATE TABLE gstest (
      id bigint primary key,
      parrot text
    );
```

```
postgres=>CREATE SEQUENCE gstest_id_seq OWNED BY gstest.id;
```

```
postgres=> ALTER TABLE gstest \
      ALTER COLUMN id SET DEFAULT \
      pgactive.pgactive_snowflake_id_nextval('gstest_id_seq');
```

Sequenze partizionate

Nelle sequenze suddivise o partizionate, viene utilizzata una normale sequenza PostgreSQL su ciascun nodo. Ogni sequenza incrementa della stessa quantità e inizia con offset diversi. Ad esempio, con il passaggio 100, il nodo 1 genera una sequenza come 101, 201, 301 e così via e il nodo 2 genera una sequenza come 102, 202, 302 e così via. Questo schema funziona bene anche se i nodi non possono comunicare per periodi prolungati, ma richiede che il progettista specifichi un numero massimo di nodi al momento di stabilire lo schema e richiede una configurazione per nodo. Gli errori possono facilmente portare alla sovrapposizione di sequenze.

È relativamente semplice configurare questo approccio con `pgactive` creando la sequenza desiderata su un nodo nel modo seguente:

```
CREATE TABLE some_table (generated_value bigint primary key);
```

```
postgres=> CREATE SEQUENCE some_seq INCREMENT 100 OWNED BY some_table.generated_value;
```

```
postgres=> ALTER TABLE some_table ALTER COLUMN generated_value SET DEFAULT
 nextval('some_seq');
```

Quindi richiama `setval` su ogni nodo per assegnare un valore iniziale di offset diverso nel modo seguente.

```
postgres=>
-- On node 1
SELECT setval('some_seq', 1);

-- On node 2
SELECT setval('some_seq', 2);
```

Riferimento ai parametri dell'estensione pgactive

È possibile utilizzare la seguente query per visualizzare tutti i parametri associati all'estensione `pgactive`.

```
postgres=> SELECT * FROM pg_settings WHERE name LIKE 'pgactive.%';
```

Misurazione del ritardo di replica tra i membri pgactive

È possibile utilizzare la seguente query per visualizzare il ritardo di replica tra i membri. `pgactive` Esegui questa query su ogni `pgactive` nodo per ottenere il quadro completo.

```
     postgres=# SELECT *, (last_applied_xact_at - last_applied_xact_committs) AS lag
 FROM pgactive.pgactive_node_slots;
-{ RECORD 1 ]---------------
+-------------------------------------------------------------
node_name                   | node2-app
slot_name                   | pgactive_5_7332551165694385385_0_5__
slot_restart_lsn            | 0/1A898A8
slot_confirmed_lsn          | 0/1A898E0
walsender_active            | t
walsender_pid               | 69022
sent_lsn                    | 0/1A898E0
write_lsn                   | 0/1A898E0
flush_lsn                   | 0/1A898E0
replay_lsn                  | 0/1A898E0
last_sent_xact_id           | 746
last_sent_xact_committs     | 2024-02-06 18:04:22.430376+00
last_sent_xact_at           | 2024-02-06 18:04:22.431359+00
last_applied_xact_id        | 746
last_applied_xact_committs  | 2024-02-06 18:04:22.430376+00
last_applied_xact_at        | 2024-02-06 18:04:52.452465+00
lag                         | 00:00:30.022089
```

Limitazioni per l'estensione pgactive

- Tutte le tabelle richiedono una chiave primaria, altrimenti le opzioni di aggiornamento ed eliminazione non sono consentite. I valori nella colonna Chiave primaria non devono essere aggiornati.
- Le sequenze possono presentare delle lacune e talvolta potrebbero non seguire un ordine. Le sequenze non vengono replicate. Per ulteriori informazioni, consulta Gestione delle sequenze nella replica active-active.
- DDL e oggetti di grandi dimensioni non vengono replicati.
- Gli indici univoci secondari possono causare divergenze tra i dati.
- Le regole di confronto devono essere identiche su tutti i nodi del gruppo.
- Il bilanciamento del carico tra i nodi è un anti-pattern.

- Le transazioni di grandi dimensioni possono causare ritardi nella replica.

Riduzione della dimensione nelle tabelle e negli indici con l'estensione pg_repack

Puoi usare l'`pg_repack`estensione per rimuovere il bloat da tabelle e indici in alternativa a. `VACUUM FULL` Questa estensione è supportata sul motore Amazon RDS for PostgreSQL versioni 9.6.3 e successive. Per ulteriori informazioni sull'`pg_repack`estensione e sul table repack completo, consulta la documentazione del progetto. GitHub

Al contrario`VACUUM FULL`, l'`pg_repack`estensione richiede un lock (AccessExclusiveLock) esclusivo solo per un breve periodo di tempo durante l'operazione di ricostruzione della tabella nei seguenti casi:

- Creazione iniziale della tabella di log: viene creata una tabella di log per registrare le modifiche che si verificano durante la copia iniziale dei dati, come illustrato nell'esempio seguente:

```
postgres=>\dt+ repack.log_*
List of relations
-[ RECORD 1 ]-+----------
Schema        | repack
Name          | log_16490
Type          | table
Owner         | postgres
Persistence   | permanent
Access method | heap
Size          | 65 MB
Description   |
```

- swap-and-drop Fase finale.

Per il resto dell'operazione di ricostruzione, è sufficiente un `ACCESS SHARE` blocco sulla tabella originale per copiare le righe da essa alla nuova tabella. Ciò consente alle operazioni INSERT, UPDATE e DELETE di procedere come al solito.

Raccomandazioni

I seguenti consigli si applicano quando si rimuove bloat dalle tabelle e dagli indici utilizzando l'estensione: `pg_repack`

- Esegui il repack durante le ore non lavorative o durante una finestra di manutenzione per minimizzarne l'impatto sulle prestazioni di altre attività del database.

- Monitora attentamente le sessioni di blocco durante l'attività di ricostruzione e assicurati che sulla tabella originale non vi siano attività potenzialmente bloccabilipg_repack, in particolare durante la swap-and-drop fase finale, quando è necessario un blocco esclusivo sulla tabella originale. Per ulteriori informazioni, vedere Identificazione degli elementi che bloccano una query.

  Quando viene visualizzata una sessione di blocco, è possibile interromperla utilizzando il seguente comando dopo un'attenta valutazione. Questo aiuta a continuare pg_repack a completare la ricostruzione:

  ```
  SELECT pg_terminate_backend(pid);
  ```

- Durante l'applicazione delle modifiche accumulate dalla tabella di pg_repack's registro su sistemi con un tasso di transazione molto elevato, il processo di applicazione potrebbe non essere in grado di tenere il passo con la frequenza delle modifiche. In questi casi, non pg_repack sarebbe in grado di completare la procedura di candidatura. Per ulteriori informazioni, consulta Monitoraggio della nuova tabella durante il repack. Se gli indici sono gravemente gonfiati, una soluzione alternativa consiste nell'eseguire un reimpacchettamento del solo indice. Questo aiuta anche i cicli di pulizia degli indici di VACUUM a terminare più velocemente.

  Puoi saltare la fase di pulizia dell'indice utilizzando VACUUM manuale dalla versione 12 di PostgreSQL e viene saltata automaticamente durante l'autovacuum di emergenza dalla versione 14 di PostgreSQL. Ciò consente a VACUUM di completare più rapidamente il processo senza rimuovere il volume dell'indice ed è pensato solo per situazioni di emergenza, ad esempio per evitare che VACUUM si verifichi. Per ulteriori informazioni, consulta Avoiding Bloat negli indici nella Guida per l'utente di Amazon Aurora.

Prerequisiti

- La tabella deve avere un vincolo PRIMARY KEY o un vincolo UNIQUE non nullo.
- La versione dell'estensione deve essere la stessa sia per il client che per il server.
- Assicurati che l'istanza RDS abbia una dimensione FreeStorageSpace superiore alla dimensione totale della tabella senza il problema. Ad esempio, considera la dimensione totale della tabella, compresi TOAST e indici, pari a 2 TB e il gonfiore totale nella tabella pari a 1 TB. Il valore richiesto FreeStorageSpace deve essere superiore al valore restituito dal seguente calcolo:

  2TB (Table size) - 1TB (Table bloat) = 1TB

È possibile utilizzare la seguente query per verificare la dimensione totale della tabella e utilizzarla `pgstattuple` per ricavare bloat. Per ulteriori informazioni, consulta Diagnosticare il gonfiamento di tabelle e indici nella Guida per l'utente di Amazon Aurora

```
SELECT pg_size_pretty(pg_total_relation_size('table_name')) AS total_table_size;
```

Questo spazio viene recuperato dopo il completamento dell'attività.

- Assicurati che l'istanza RDS abbia una capacità di elaborazione e IO sufficiente per gestire l'operazione di repack. Potresti prendere in considerazione l'idea di aumentare la classe di istanze per un equilibrio ottimale delle prestazioni.

Per utilizzare l'**`pg_repack`**estensione

1. Installa l'estensione `pg_repack` nell'istanza database di Amazon RDS for PostgreSQL eseguendo il comando seguente.

   ```
   CREATE EXTENSION pg_repack;
   ```

2. Esegui i seguenti comandi per concedere l'accesso in scrittura alle tabelle di registro temporanee create da`pg_repack`.

   ```
   ALTER DEFAULT PRIVILEGES IN SCHEMA repack GRANT INSERT ON TABLES TO PUBLIC;
   ALTER DEFAULT PRIVILEGES IN SCHEMA repack GRANT USAGE, SELECT ON SEQUENCES TO
    PUBLIC;
   ```

3. Connect al database utilizzando l'utilità `pg_repack` client. Usa un account con privilegi `rds_superuser`. Ad esempio, presumiamo che il ruolo `rds_test` abbia i privilegi `rds_superuser`. La sintassi seguente funziona `pg_repack` per tabelle complete, inclusi tutti gli indici delle tabelle del database. `postgres`

   ```
   pg_repack -h db-instance-name.111122223333.aws-region.rds.amazonaws.com -U rds_test
    -k postgres
   ```

   > **Note**
   >
   > È necessario connettersi utilizzando l'opzione -k. L'opzione a non è supportata.

La risposta del `pg_repack` client fornisce informazioni sulle tabelle dell'istanza DB che vengono reimballate.

```
INFO: repacking table "pgbench_tellers"
INFO: repacking table "pgbench_accounts"
INFO: repacking table "pgbench_branches"
```

4. La sintassi seguente ripacchetta una singola tabella, `orders` inclusi gli indici nel database. `postgres`

```
pg_repack -h db-instance-name.111122223333.aws-region.rds.amazonaws.com -U rds_test
 --table orders -k postgres
```

La sintassi seguente riimpacchetta solo gli indici per la tabella nel database. `orders` `postgres`

```
pg_repack -h db-instance-name.111122223333.aws-region.rds.amazonaws.com -U rds_test
 --table orders --only-indexes -k postgres
```

Monitoraggio della nuova tabella durante il repack

- La dimensione del database viene aumentata della dimensione totale della tabella meno il bloat, fino alla swap-and-drop fase di repack. È possibile monitorare il tasso di crescita delle dimensioni del database, calcolare la velocità del repack e stimare approssimativamente il tempo necessario per completare il trasferimento iniziale dei dati.

  Ad esempio, considera la dimensione totale della tabella di 2 TB, la dimensione del database di 4 TB e il volume totale della tabella di 1 TB. Il valore della dimensione totale del database restituito dal calcolo al termine dell'operazione di repack è il seguente:

  `2TB (Table size) + 4 TB (Database size) - 1TB (Table bloat) = 5TB`

  È possibile stimare approssimativamente la velocità dell'operazione di repack campionando il tasso di crescita in byte tra due momenti nel tempo. Se il tasso di crescita è di 1 GB al minuto, possono essere necessari 1000 minuti o 16,6 ore circa per completare l'operazione iniziale di creazione della tabella. Oltre alla creazione iniziale della tabella, deve applicare `pg_repack` anche le modifiche maturate. Il tempo necessario dipende dalla velocità di applicazione delle modifiche in corso e delle modifiche maturate.

> **Note**
>
> È possibile utilizzare l'`pgstattuple`estensione per calcolare il rigonfiamento nella tabella. Per ulteriori informazioni, consulta pgstattuple.

- Il numero di righe nella tabella di `pg_repack's` log, secondo lo schema repack, rappresenta il volume delle modifiche in sospeso da applicare alla nuova tabella dopo il caricamento iniziale.

  È possibile controllare la tabella di `pg_repack's` registro `pg_stat_all_tables` per monitorare le modifiche applicate alla nuova tabella. `pg_stat_all_tables.n_live_tup`indica il numero di record in attesa di applicazione alla nuova tabella. Per ulteriori informazioni, vedere pg_stat_all_tables.

```
postgres=>SELECT relname,n_live_tup FROM pg_stat_all_tables WHERE schemaname =
 'repack' AND relname ILIKE '%log%';

-[ RECORD 1 ]---------
relname    | log_16490
n_live_tup | 2000000
```

- Puoi utilizzare l'`pg_stat_statements`estensione per scoprire il tempo impiegato da ogni fase dell'operazione di repack. Ciò è utile in preparazione all'applicazione della stessa operazione di reimballaggio in un ambiente di produzione. È possibile modificare la `LIMIT` clausola per estendere ulteriormente l'output.

```
postgres=>SELECT
     SUBSTR(query, 1, 100) query,
     round((round(total_exec_time::numeric, 6) / 1000 / 60),4)
 total_exec_time_in_minutes
 FROM
     pg_stat_statements
 WHERE
     query ILIKE '%repack%'
 ORDER BY
     total_exec_time DESC LIMIT 5;

 query                                                                |
 total_exec_time_in_minutes
```

```
------------------------------------------------------------------------
+---------------------------
 CREATE UNIQUE INDEX index_16493 ON repack.table_16490 USING btree (a) |
        6.8627
 INSERT INTO repack.table_16490 SELECT a FROM ONLY public.t1           |
        6.4150
 SELECT repack.repack_apply($1, $2, $3, $4, $5, $6)                    |
        0.5395
 SELECT repack.repack_drop($1, $2)                                     |
        0.0004
 SELECT repack.repack_swap($1)                                         |
        0.0004
(5 rows)
```

Il reimballaggio è un' out-of-place operazione completa, quindi la tabella originale non ne risente e non prevediamo problemi imprevisti che richiedano il ripristino della tabella originale. Se il repack fallisce in modo imprevisto, è necessario verificare la causa dell'errore e risolverlo.

Dopo aver risolto il problema, rilascia e ricrea l'`pg_repack`estensione nel database in cui esiste la tabella e riprova il passaggio. `pg_repack` Inoltre, la disponibilità di risorse di calcolo e l'accessibilità simultanea della tabella svolgono un ruolo cruciale nel completamento tempestivo dell'operazione di repack.

Aggiornamento e utilizzo dell'estensione PLV8

PLV8 è un'estensione del linguaggio Javascript affidabile per PostgreSQL. Puoi utilizzarlo per le procedure archiviate, i trigger e altri codici procedurali richiamabili da SQL. Questa estensione della lingua è supportata da tutte le versioni correnti di PostgreSQL.

Se si utilizza PLV8 e si aggiorna PostgreSQL a una nuova versione PLV8, si sfrutta immediatamente la nuova estensione. Le fasi seguenti consentono di sincronizzare i metadati del catalogo con la nuova versione di PLV8. Queste fasi sono facoltative ma fortemente consigliate per evitare avvisi di mancata corrispondenza dei metadati.

Il processo di aggiornamento elimina tutte le funzioni di PLV8 esistenti. Pertanto, consigliamo di creare uno snapshot dell'istanza database di RDS for PostgreSQL prima dell'aggiornamento. Per ulteriori informazioni, consulta Creazione di uno snapshot DB per un'istanza DB Single-AZ.

Per sincronizzare i metadati del catalogo con una nuova versione di PLV8

1. Verificare la necessità dell'aggiornamento. Per procedere, eseguire il comando seguente durante la connessione alla propria istanza.

   ```
   SELECT * FROM pg_available_extensions WHERE name IN ('plv8','plls','plcoffee');
   ```

   Se i risultati contengono valori per una versione installata inferiore rispetto alla versione predefinita, continuare con la procedura per aggiornare le estensioni. Ad esempio, il seguente set di risultati indica che è necessario eseguire l'aggiornamento:

   ```
   name    | default_version | installed_version |                    comment
   --------+-----------------+-------------------
   +-----------------------------------------------
   plls    | 2.1.0           | 1.5.3             | PL/LiveScript (v8) trusted
    procedural language
   plcoffee| 2.1.0           | 1.5.3             | PL/CoffeeScript (v8) trusted
    procedural language
   plv8    | 2.1.0           | 1.5.3             | PL/JavaScript (v8) trusted
    procedural language
   (3 rows)
   ```

2. Crea uno snapshot della tua istanza database RDS for PostgreSQL se non lo hai ancora fatto. Puoi continuare con la seguente procedura una volta creata la snapshot.

3. Ottieni il numero delle funzioni di PLV8 nell'istanza database in modo che sia possibile convalidarne la disponibilità dopo l'aggiornamento. Ad esempio, la seguente query SQL restituisce il numero di funzioni scritte in plv8, plcoffee e plls.

```
SELECT proname, nspname, lanname
FROM pg_proc p, pg_language l, pg_namespace n
WHERE p.prolang = l.oid
AND n.oid = p.pronamespace
AND lanname IN ('plv8','plcoffee','plls');
```

4. Utilizzare pg_dump per creare un file dump solo schema. Ad esempio, crea un file nel computer client nella directory `/tmp`.

```
./pg_dump -Fc --schema-only -U master postgres >/tmp/test.dmp
```

In questo esempio vengono utilizzate le seguenti opzioni:

- `-Fc` – Formato personalizzato
- --schema-only – Esecuzione del dump solo dei comandi necessari per creare lo schema (funzioni in questo caso)
- `-U` – Nome utente principale di RDS
- `database` – Nome del database nell'istanza database

Per ulteriori informazioni su pg_dump, consulta pg_dump nella documentazione di PostgreSQL.

5. Estrarre l'istruzione DDL "CREATE FUNCTION" presente nel file dump. Gli esempi seguenti utilizzano il comando `grep` per estrarre l'istruzione DDL che crea le funzioni e salvarle in un file. Verrà utilizzata nelle fasi successive per ricreare le funzioni.

```
./pg_restore -l /tmp/test.dmp | grep FUNCTION > /tmp/function_list/
```

Per ulteriori informazioni su pg_restore, consulta pg_dump nella documentazione di PostgreSQL.

6. Eliminare le funzioni e le estensioni. Il seguente esempio elimina qualsiasi oggetto basato su PLV8. L'opzione a cascata assicura l'eliminazione di ogni elemento dipendente.

```
DROP EXTENSION plv8 CASCADE;
```

Se l'istanza PostgreSQL contiene degli oggetti basati su plcoffee o plls, ripetere la procedura per quelle estensioni.

7. Creare le estensioni. Il seguente esempio crea le estensioni plv8, plcoffee e plls.

```
CREATE EXTENSION plv8;
CREATE EXTENSION plcoffee;
CREATE EXTENSION plls;
```

8. Creare le funzioni utilizzando il file dump e il file "driver".

   Il seguente esempio ricrea le funzioni estratte in precedenza.

```
./pg_restore -U master -d postgres -Fc -L /tmp/function_list /tmp/test.dmp
```

9. Verifica che tutte le funzioni siano state ricreate utilizzando la seguente query.

```
SELECT * FROM pg_available_extensions WHERE name IN ('plv8','plls','plcoffee');
```

   PLV8 versione 2 aggiunge questa riga aggiuntiva al set di risultati:

```
    proname    |  nspname   | lanname
---------------+------------+----------
 plv8_version  | pg_catalog | plv8
```

Utilizzo di PL/Rust per scrivere funzioni PostgreSQL nel linguaggio Rust

PL/Rust è un'estensione attendibile del linguaggio Rust per PostgreSQL. È possibile utilizzarlo per stored procedure, funzioni e altri codici procedurali richiamabili da SQL. L'estensione del linguaggio PL/Rust è disponibile nelle seguenti versioni:

- RDS per PostgreSQL 16.1 e versioni successive 16
- RDS per PostgreSQL 15.2-R2 e versioni successive alla 15
- RDS per PostgreSQL 14.9 e versioni successive alla 14
- RDS per PostgreSQL 13.12 e versioni successive alla 13

Per ulteriori informazioni, vedere PL/Rust on. GitHub

Argomenti

Configurazione di PL/Rust

Per installare l'estensione plrust sull'istanza database, aggiungere plrust al parametro `shared_preload_libraries` nel gruppo di parametri database associato all'istanza database. Dopo aver installato l'estensione plrust, è possibile creare funzioni.

Per modificare il parametro `shared_preload_libraries`, l'istanza database deve essere associata a un gruppo di parametri personalizzato. Per ulteriori informazioni sulla creazione di un gruppo di parametri personalizzato, consulta Utilizzo di gruppi di parametri.

È possibile installare l'estensione plrust utilizzando o il AWS Management Console . AWS CLI

I passaggi seguenti si basano sull'ipotesi che l'istanza database sia associata a un gruppo di parametri personalizzato.

Console

Installare l'estensione plrust nel parametro **`shared_preload_libraries`**

Eseguire i seguenti passaggi utilizzando un account membro del gruppo `rds_superuser` (ruolo).

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, scegli Databases (Database).
3. Scegliere il nome dell'istanza database per visualizzarne i dettagli.
4. Aprire la scheda Configurazione dell'istanza database e trovare il link al gruppo di parametri dell'istanza database.
5. Scegliere il link per aprire i parametri personalizzati associati all'istanza database.
6. Nel campo di ricerca Parametri, digita `shared_pre` per trovare il parametro **`shared_preload_libraries`**.

7. Scegli Edit parameters (Modifica parametri) per accedere ai valori delle proprietà.
8. Aggiungere plrust all'elenco nel campo Valori. Utilizza una virgola per separare gli elementi nell'elenco di valori.
9. Riavviare l'istanza database per applicare le modifiche al parametro `shared_preload_libraries`. Il completamento del riavvio iniziale potrebbe richiedere più tempo.
10. Quando l'istanza è disponibile, verificare che plrust sia stato inizializzato. Utilizzare `psql` per connettersi all'istanza database ed eseguire il comando riportato di seguito.

    ```
    SHOW shared_preload_libraries;
    ```

    L'aspetto dell'output sarà simile al seguente:

    ```
    shared_preload_libraries
    --------------------------
    rdsutils,plrust
    (1 row)
    ```

AWS CLI

Installare l'estensione plrust nel parametro shared_preload_libraries

Eseguire i seguenti passaggi utilizzando un account membro del gruppo `rds_superuser` (ruolo).

1. Usa il modify-db-parameter-group AWS CLI comando per aggiungere plrust al `shared_preload_libraries` parametro.

    ```
    aws rds modify-db-parameter-group \
       --db-parameter-group-name custom-param-group-name \
       --parameters
     "ParameterName=shared_preload_libraries,ParameterValue=plrust,ApplyMethod=pending-
    reboot" \
       --region aws-region
    ```

2. Utilizzate il reboot-db-instance AWS CLI comando per riavviare l'istanza DB e inizializzare la libreria plrust. Il completamento del riavvio iniziale potrebbe richiedere più tempo.

    ```
    aws rds reboot-db-instance \
        --db-instance-identifier your-instance \
    ```

```
    --region aws-region
```

3. Quando l'istanza è disponibile, è possibile verificare se plrust è stato inizializzato. Utilizzare `psql` per connettersi all'istanza database ed eseguire il comando riportato di seguito.

```
SHOW shared_preload_libraries;
```

L'aspetto dell'output sarà simile al seguente:

```
shared_preload_libraries
-------------------------
rdsutils,plrust
(1 row)
```

Creazione di funzioni con PL/Rust

PL/Rust compilerà la funzione come libreria dinamica, la caricherà e la eseguirà.

La seguente funzione Rust filtra i multipli da un array.

```
postgres=> CREATE LANGUAGE plrust;
CREATE EXTENSION
```

```
CREATE OR REPLACE FUNCTION filter_multiples(a BIGINT[], multiple BIGINT) RETURNS
 BIGINT[]
    IMMUTABLE STRICT
    LANGUAGE PLRUST AS
$$
    Ok(Some(a.into_iter().filter(|x| x.unwrap() % multiple != 0).collect()))
$$;

WITH gen_values AS (
SELECT ARRAY(SELECT * FROM generate_series(1,100)) as arr)
SELECT filter_multiples(arr, 3)
from gen_values;
```

Utilizzo dei formati crate con PL/Rust

A partire dalle versioni 15.4, 14.9 e 13.12 di Amazon RDS for PostgreSQL, PL/Rust supporta le seguenti casse:

- `aes`
- `ctr`
- `rand`

A partire dalle versioni RDS per PostgreSQL 15.5-R2, 14.10-R2 e 13.13-R2, PL/Rust supporta due casse aggiuntive:

- `croaring-rs`
- `num-bigint`

Per questi formati sono supportate solo le funzionalità predefinite. Le nuove versioni di RDS per PostgreSQL potrebbero contenere versioni aggiornate di questi formati e le relative versioni precedenti potrebbero non essere più supportate.

Segui le best practice per eseguire un aggiornamento della versione principale per verificare se le tue funzioni PL/Rust sono compatibili con la nuova versione principale. Per ulteriori informazioni, consulta il blog Best practices for upgrading Amazon RDS to major and minor versions of PostgreSQL (Best practice per l'aggiornamento di Amazon RDS alle versioni principali e secondarie di PostgreSQL) e Aggiornamento del motore del database PostgreSQL per Amazon RDS nella Guida per l'utente Amazon RDS.

Esempi di utilizzo delle dipendenze durante la creazione di una funzione PL/Rust sono disponibili in Use dependencies (Utilizzo delle dipendenze).

Limitazioni del linguaggio PL/Rust

Per impostazione predefinita, gli utenti del database non possono utilizzare PL/Rust. Per fornire l'accesso a PL/Rust, connettersi come utente con il privilegio rds_superuser ed eseguire il seguente comando:

```
postgres=> GRANT USAGE ON LANGUAGE PLRUST TO user;
```

Gestione dei dati spaziali con estensione PostGIS

PostGIS è un'estensione di PostgreSQL per l'archiviazione e la gestione delle informazioni spaziali. Per ulteriori informazioni su PostGIS, consulta PostGIS.net.

A partire dalla versione 10.5, PostgreSQL supporta la libreria libprotobuf 1.3.0 utilizzata da PostGIS per lavorare con i dati delle tile vettoriali delle mappe.

La configurazione dell'estensione PostGIS richiede i privilegi `rds_superuser`. Ti consigliamo di creare un utente (ruolo) per gestire l'estensione PostGIS e i dati spaziali. L'estensione PostGIS e i relativi componenti aggiungono migliaia di funzioni a PostgreSQL. Considera la possibilità di creare l'estensione PostGIS nel proprio schema se ciò ha senso per il tuo caso d'uso. Nell'esempio seguente viene illustrato come installare l'estensione nel proprio database, ma questa operazione non è necessaria.

Argomenti

Passaggio 1: Creazione di un utente (ruolo) per gestire l'estensione PostGIS

Per prima cosa, esegui la connessione a un'istanza database RDS per PostgreSQL come utente con i privilegi `rds_superuser`. Se hai mantenuto il nome di default durante la configurazione dell'istanza, esegui la connessione come `postgres`.

```
psql --host=111122223333.aws-region.rds.amazonaws.com --port=5432 --username=postgres
 --password
```

Crea un ruolo separato (utente) per amministrare l'estensione PostGIS.

```
postgres=>  CREATE ROLE gis_admin LOGIN PASSWORD 'change_me';
```

```
CREATE ROLE
```

Concedi a questo ruolo i privilegi `rds_superuser` per consentire l'installazione dell'estensione.

```
postgres=> GRANT rds_superuser TO gis_admin;
GRANT
```

Crea un database da utilizzare per gli artefatti PostGIS. Questa fase è facoltativa. In alternativa, puoi creare uno schema nel database utente per le estensioni PostGIS, ma anche questa operazione non è necessaria.

```
postgres=> CREATE DATABASE lab_gis;
CREATE DATABASE
```

Concedi a `gis_admin` tutti i privilegi per il database `lab_gis`.

```
postgres=> GRANT ALL PRIVILEGES ON DATABASE lab_gis TO gis_admin;
GRANT
```

Esci dalla sessione ed esegui nuovamente la connessione all'istanza database RDS per PostgreSQL come `gis_admin`.

```
postgres=> psql --host=111122223333.aws-region.rds.amazonaws.com --port=5432 --
username=gis_admin --password --dbname=lab_gis
Password for user gis_admin:...
lab_gis=>
```

Continua a configurare l'estensione come descritto nei passaggi successivi.

Passaggio 2: Caricamento delle estensioni di PostGIS

L'estensione PostGIS include diverse estensioni correlate che interagiscono per fornire funzionalità geospaziali. A seconda del caso d'uso, è possibile che le estensioni create in questo passaggio non siano tutte necessarie.

Utilizzare `CREATE EXTENSION` le istruzioni per caricare le estensioni PostGIS.

```
CREATE EXTENSION postgis;
CREATE EXTENSION
CREATE EXTENSION postgis_raster;
CREATE EXTENSION
```

```
CREATE EXTENSION fuzzystrmatch;
CREATE EXTENSION
CREATE EXTENSION postgis_tiger_geocoder;
CREATE EXTENSION
CREATE EXTENSION postgis_topology;
CREATE EXTENSION
CREATE EXTENSION address_standardizer_data_us;
CREATE EXTENSION
```

È possibile verificare i risultati eseguendo la query SQL mostrata nel seguente esempio, che elenca le estensioni e i relativi proprietari.

```
SELECT n.nspname AS "Name",
  pg_catalog.pg_get_userbyid(n.nspowner) AS "Owner"
  FROM pg_catalog.pg_namespace n
  WHERE n.nspname !~ '^pg_' AND n.nspname <> 'information_schema'
  ORDER BY 1;
List of schemas
     Name     |   Owner
--------------+-----------
 public       | postgres
 tiger        | rdsadmin
 tiger_data   | rdsadmin
 topology     | rdsadmin
(4 rows)
```

Passaggio 3: Trasferimento della proprietà delle estensioni

Usare le istruzioni ALTER SCHEMA per trasferire la proprietà degli schemi al ruolo `gis_admin`.

```
ALTER SCHEMA tiger OWNER TO gis_admin;
ALTER SCHEMA
ALTER SCHEMA tiger_data OWNER TO gis_admin;
ALTER SCHEMA
ALTER SCHEMA topology OWNER TO gis_admin;
ALTER SCHEMA
```

È possibile confermare la modifica della proprietà eseguendo la seguente query SQL. Oppure è possibile utilizzare il meta-comando \dn dalla riga di comando psql.

```
SELECT n.nspname AS "Name",
```

```
  pg_catalog.pg_get_userbyid(n.nspowner) AS "Owner"
  FROM pg_catalog.pg_namespace n
  WHERE n.nspname !~ '^pg_' AND n.nspname <> 'information_schema'
  ORDER BY 1;

       List of schemas
     Name     |     Owner
--------------+---------------
 public       | postgres
 tiger        | gis_admin
 tiger_data   | gis_admin
 topology     | gis_admin
(4 rows)
```

Fase 4: Trasferimento della proprietà degli oggetti PostGIS

Usare la funzione seguente per trasferire la proprietà degli oggetti PostGIS al ruolo `gis_admin`. Eseguire la seguente istruzione dal prompt di psql per creare la funzione.

```
CREATE FUNCTION exec(text) returns text language plpgsql volatile AS $f$ BEGIN EXECUTE
 $1; RETURN $1; END; $f$;
CREATE FUNCTION
```

Successivamente, eseguire la seguente query per eseguire la funzione `exec` che a sua volta esegue le istruzioni e altera le autorizzazioni.

```
SELECT exec('ALTER TABLE ' || quote_ident(s.nspname) || '.' || quote_ident(s.relname)
 || ' OWNER TO gis_admin;')
  FROM (
    SELECT nspname, relname
    FROM pg_class c JOIN pg_namespace n ON (c.relnamespace = n.oid)
    WHERE nspname in ('tiger','topology') AND
    relkind IN ('r','S','v') ORDER BY relkind = 'S')
s;
```

Passaggio 5: Verificare le estensioni

Per evitare di dover specificare il nome dello schema, aggiungi lo schema `tiger` al percorso di ricerca usando il seguente comando.

```
SET search_path=public,tiger;
```

```
SET
```

Verifica lo schema `tiger` usando la seguente istruzione SELECT.

```
SELECT address, streetname, streettypeabbrev, zip
 FROM normalize_address('1 Devonshire Place, Boston, MA 02109') AS na;
address | streetname | streettypeabbrev |  zip
---------+------------+------------------+-------
       1 | Devonshire | Pl               | 02109
(1 row)
```

Per ulteriori informazioni su questa estensione, consulta Tiger Geocoder nella documentazione di PostGIS.

Verifica lo schema `topology` usando la seguente istruzione `SELECT`. Questa richiama la funzione `createtopology` per registrare un nuovo oggetto topologia (my_new_topo) con l'identificatore di riferimento spaziale specificato (26986) e la tolleranza predefinita (0,5). Per saperne di più, consulta la CreateTopologydocumentazione di PostGIS.

```
SELECT topology.createtopology('my_new_topo',26986,0.5);
 createtopology
----------------
              1
(1 row)
```

Passaggio 6: Aggiornamento dell'estensione PostGIS

Ogni nuova versione di PostgreSQL supporta una o più versioni dell'estensione PostGIS compatibili con tale versione. L'aggiornamento del motore PostgreSQL a una nuova versione non aggiorna automaticamente l'estensione PostGIS. Prima di aggiornare il motore PostgreSQL, in genere si aggiorna PostGIS alla versione più recente disponibile per la versione di PostgreSQL corrente. Per informazioni dettagliate, vedi Versioni dell'estensione PostGIS.

Dopo l'aggiornamento del motore PostgreSQL, si aggiorna nuovamente l'estensione PostGIS alla versione supportata per la versione del motore PostgreSQL aggiornata. Per ulteriori informazioni sull'aggiornamento del motore PostgreSQL, consulta Come eseguire l'aggiornamento a una versione principale.

Puoi verificare la disponibilità di aggiornamenti della versione dell'estensione PostGIS sull'istanza database RDS per PostgreSQL in qualsiasi momento. Per farlo, esegui il comando seguente. Questa funzione è disponibile con PostGIS 2.5.0 e versioni successive.

```
SELECT postGIS_extensions_upgrade();
```

Se l'applicazione non supporta la versione più recente di PostGIS, puoi installare una versione precedente di PostGIS disponibile nella versione principale, come indicato di seguito.

```
CREATE EXTENSION postgis VERSION "2.5.5";
```

Se desideri eseguire l'aggiornamento a una versione PostGIS specifica da una versione precedente, puoi anche utilizzare il seguente comando.

```
ALTER EXTENSION postgis UPDATE TO "2.5.5";
```

A seconda della versione da cui si esegue l'aggiornamento, potrebbe essere necessario utilizzare nuovamente questa funzione. Il risultato della prima esecuzione della funzione determina se è necessaria una funzione di aggiornamento aggiuntiva. Ad esempio, questo si verifica per l'aggiornamento da PostGIS 2 a PostGIS 3. Per ulteriori informazioni, consulta Aggiornamento di PostGIS 2 a PostGIS 3.

Se l'estensione è stata aggiornata in preparazione a un aggiornamento della versione principale del motore PostgreSQL, puoi continuare con altre attività preliminari. Per ulteriori informazioni, consulta Come eseguire l'aggiornamento a una versione principale.

Versioni dell’estensione PostGIS

Ti consigliamo di installare le versioni di tutte le estensioni, ad esempio PostGIS, come elencato in Versioni delle estensioni per Amazon RDS per PostgreSQL nelle Note di rilascio di Amazon RDS per PostgreSQL. Per ottenere un elenco delle versioni disponibili nella versione, utilizza il comando seguente.

```
SELECT * FROM pg_available_extension_versions WHERE name='postgis';
```

Puoi trovare le informazioni sulle versioni nelle sezioni seguenti delle Note di rilascio di Amazon RDS per PostgreSQL:

- Estensioni PostgreSQL versione 16 supportate su Amazon RDS

- Estensioni di PostgreSQL versione 15 supportate su Amazon RDS
- Estensioni di PostgreSQL versione 14 supportate su Amazon RDS
- Estensioni di PostgreSQL versione 13 supportate su Amazon RDS
- Estensioni di PostgreSQL versione 12 supportate su Amazon RDS
- Estensioni di PostgreSQL versione 11 supportate su Amazon RDS
- Estensioni di PostgreSQL versione 10 supportate su Amazon RDS
- Estensioni di PostgreSQL versione 9.6.x supportate su Amazon RDS

Aggiornamento di PostGIS 2 a PostGIS 3

A partire dalla versione 3.0, la funzionalità raster di PostGIS è ora un'estensione separata, `postgis_raster`. Questa estensione dispone di un proprio percorso di installazione e aggiornamento. Ciò rimuove dall'estensione `postgis` core dozzine di funzioni, tipi di dati e altri artefatti necessari per l'elaborazione di immagini raster. Ciò significa che se il caso d'uso non richiede l'elaborazione raster, non è necessario installare l'estensione `postgis_raster`.

Nel seguente esempio di aggiornamento, il primo comando di aggiornamento estrae la funzionalità raster nell'estensione `postgis_raster`. È quindi necessario un secondo comando di aggiornamento per eseguire l'aggiornamento di `postgres_raster` alla nuova versione.

Per eseguire l'aggiornamento da PostGIS 2 a PostGIS 3

1. Identifica la versione predefinita di PostGIS disponibile per la versione PostgreSQL sul l'istanza database RDS per PostgreSQL. A questo scopo, esegui la query seguente.

```
SELECT * FROM pg_available_extensions
    WHERE default_version > installed_version;
  name   | default_version | installed_version |                          comment
---------+-----------------+-------------------
+------------------------------------------------------------
 postgis | 3.1.4           | 2.3.7             | PostGIS geometry and geography
 spatial types and functions
(1 row)
```

2. Identifica le versioni di PostGIS installate in ogni database sull'istanza database RDS per PostgreSQL. In altre parole, esegui la query su ogni database utente come riportato di seguito.

```
SELECT
```

```
    e.extname AS "Name",
    e.extversion AS "Version",
    n.nspname AS "Schema",
    c.description AS "Description"
FROM
    pg_catalog.pg_extension e
    LEFT JOIN pg_catalog.pg_namespace n ON n.oid = e.extnamespace
    LEFT JOIN pg_catalog.pg_description c ON c.objoid = e.oid
    AND c.classoid = 'pg_catalog.pg_extension'::pg_catalog.regclass
WHERE
    e.extname LIKE '%postgis%'
ORDER BY
    1;
  Name   | Version | Schema |                            Description
---------+---------+--------
+-------------------------------------------------------------------------
 postgis | 2.3.7   | public | PostGIS geometry, geography, and raster spatial types
 and functions
(1 row)
```

Questa discrepanza tra la versione predefinita (PostGIS 3.1.4) e la versione installata (PostGIS 2.3.7) indica che è necessario aggiornare l'estensione PostGIS.

```
ALTER EXTENSION postgis UPDATE;
ALTER EXTENSION
WARNING: unpackaging raster
WARNING: PostGIS Raster functionality has been unpackaged
```

3. Esegui la seguente query per verificare che la funzionalità raster sia ora contenuta nel proprio pacchetto.

```
SELECT
    probin,
    count(*)
FROM
    pg_proc
WHERE
    probin LIKE '%postgis%'
GROUP BY
    probin;
          probin          | count
--------------------------+-------
```

```
 $libdir/rtpostgis-2.3    | 107
 $libdir/postgis-3        | 487
(2 rows)
```

L'output mostra che c'è ancora una differenza tra le versioni. Le funzioni PostGIS sono versione 3 (postgis-3), mentre le funzioni raster (rtpostgis) sono versione 2 (rtpostgis-2.3). Per completare l'aggiornamento, esegui nuovamente il comando di aggiornamento, come riportato di seguito.

```
postgres=> SELECT postgis_extensions_upgrade();
```

Puoi ignorare i messaggi di avviso in sicurezza. Esegui nuovamente la seguente query per verificare che l'aggiornamento sia stato completato. L'aggiornamento è completato quando PostGIS e tutte le estensioni correlate non sono contrassegnate come necessarie di aggiornamento.

```
SELECT postgis_full_version();
```

4. Utilizza la seguente query per visualizzare il processo di aggiornamento completato e le estensioni impacchettate separatamente e verifica che le relative versioni corrispondano.

```
SELECT
    e.extname AS "Name",
    e.extversion AS "Version",
    n.nspname AS "Schema",
    c.description AS "Description"
FROM
    pg_catalog.pg_extension e
    LEFT JOIN pg_catalog.pg_namespace n ON n.oid = e.extnamespace
    LEFT JOIN pg_catalog.pg_description c ON c.objoid = e.oid
        AND c.classoid = 'pg_catalog.pg_extension'::pg_catalog.regclass
WHERE
    e.extname LIKE '%postgis%'
ORDER BY
    1;
      Name      | Version | Schema |                            Description
----------------+---------+--------
+------------------------------------------------------------------------
 postgis        | 3.1.5   | public | PostGIS geometry, geography, and raster
 spatial types and functions
 postgis_raster | 3.1.5   | public | PostGIS raster types and functions
```

```
(2 rows)
```

L'output mostra che l'estensione PostGIS 2 è stata aggiornata a PostGIS 3 e che `postgis` e l'estensione `postgis_raster` ora separata sono entrambe versione 3.1.5.

Al termine dell'aggiornamento, se non prevedi di utilizzare la funzionalità raster, puoi rimuovere l'estensione come segue.

```
DROP EXTENSION postgis_raster;
```

Utilizzo dei wrapper di dati esterni supportati per Amazon RDS for PostgreSQL

Un wrapper di dati esterni (FDW) è uno specifico tipo di estensione che consente l'accesso a dati esterni. Ad esempio, l'estensione `oracle_fdw` consente al cluster di database RDS for PostgreSQL di interagire con i database Oracle. Come ulteriore esempio, utilizzando l'estensione `postgres_fdw` nativa di PostgreSQL è possibile accedere ai dati memorizzati in istanze database di PostgreSQL esterne ad Amazon RDS for PostgreSQL.

Di seguito sono disponibili le informazioni sui diversi wrapper di dati esterni di PostgreSQL supportati.

Argomenti

Utilizzo dell'estensione log_fdw per accedere al registro di database utilizzando SQL

L'istanza database RDS per PostgreSQL supporta l'estensione `log_fdw` che consente di accedere al log del motore del database utilizzando un'interfaccia SQL. L'estensione `log_fdw` offre due nuove funzioni che semplificano la creazione di tabelle esterne per i registri di database:

- `list_postgres_log_files` – Elenca i file nella directory dei registri di database e le dimensioni dei file in byte.
- `create_foreign_table_for_log_file(table_name text, server_name text, log_file_name text)` – Crea una tabella esterna per il file specificato nel database corrente.

Tutte le funzioni create da `log_fdw` sono di proprietà di `rds_superuser`. I membri del ruolo `rds_superuser` possono concedere l'accesso a queste funzioni ad altri utenti del database.

Per impostazione predefinita, i file di log vengono generati da Amazon RDS nel formato `stderr` (errore standard), come specificato nel parametro `log_destination`. Esistono solo due opzioni per

questo parametro, `stderr` e `csvlog` (valori separati da virgola, CSV). Se aggiungi l'opzione `csvlog` al parametro, Amazon RDS genera entrambi i log `stderr` e `csvlog`. Ciò può influire sulla capacità di archiviazione del cluster di database, quindi è necessario tenere conto degli altri parametri che influiscono sulla gestione dei log. Per ulteriori informazioni, consulta Impostazione della destinazione del registro (`stderr, csvlog`).

Un vantaggio della generazione dei registri `csvlog` è che l'estensione `log_fdw` consente di costruire le tabelle esterne con i dati suddivisi in diverse colonne. Per eseguire questa operazione, l'istanza deve essere associata a un gruppo parametri del database personalizzato in modo da poter modificare l'impostazione per `log_destination`. Per ulteriori informazioni su come fare, consulta Utilizzo dei parametri sull'istanza database RDS for PostgreSQL.

L'esempio seguente presuppone che il parametro `log_destination` includa `cvslog`.

Per utilizzare l'estensione log_fdw

1. Installa l'estensione `log_fdw`.

```
postgres=> CREATE EXTENSION log_fdw;
CREATE EXTENSION
```

2. Creare un server log come wrapper di dati esterno.

```
postgres=> CREATE SERVER log_server FOREIGN DATA WRAPPER log_fdw;
CREATE SERVER
```

3. Selezionare tutti gli elementi da un elenco di file di registro.

```
postgres=> SELECT * FROM list_postgres_log_files() ORDER BY 1;
```

   Di seguito è riportata una risposta di esempio.

```
          file_name           | file_size_bytes
------------------------------+-----------------
 postgresql.log.2023-08-09-22.csv |            1111
 postgresql.log.2023-08-09-23.csv |            1172
 postgresql.log.2023-08-10-00.csv |            1744
 postgresql.log.2023-08-10-01.csv |            1102
(4 rows)
```

4. Creare una tabella con una singola colonna "log_entry" per i file selezionato.

```
postgres=> SELECT create_foreign_table_for_log_file('my_postgres_error_log',
     'log_server', 'postgresql.log.2023-08-09-22.csv');
```

La risposta non fornisce dettagli diversi da quello che la tabella ora esiste.

```
----------------------------------
(1 row)
```

5. Selezionare un campione del file di registro. Il seguente codice recupera l'ora del log e la descrizione del messaggio di errore.

```
postgres=> SELECT log_time, message FROM my_postgres_error_log ORDER BY 1;
```

Di seguito è riportata una risposta di esempio.

```
             log_time             |                                 message
----------------------------------
+------------------------------------------------------------------------------
Tue Aug 09 15:45:18.172 2023 PDT | ending log output to stderr
Tue Aug 09 15:45:18.175 2023 PDT | database system was interrupted; last known up
 at 2023-08-09 22:43:34 UTC
Tue Aug 09 15:45:18.223 2023 PDT | checkpoint record is at 0/90002E0
Tue Aug 09 15:45:18.223 2023 PDT | redo record is at 0/90002A8; shutdown FALSE
Tue Aug 09 15:45:18.223 2023 PDT | next transaction ID: 0/1879; next OID: 24578
Tue Aug 09 15:45:18.223 2023 PDT | next MultiXactId: 1; next MultiXactOffset: 0
Tue Aug 09 15:45:18.223 2023 PDT | oldest unfrozen transaction ID: 1822, in
 database 1
(7 rows)
```

Utilizzo dell'estensione postgres_fdw per accedere a dati esterni

È possibile accedere ai dati in una tabella su un server di database remoto con l'estensione postgres_fdw. Se si imposta una connessione remota dall'istanza database di PostgreSQL, l'accesso è disponibile anche alla replica di lettura.

Usare postgres_fdw per accedere al server remoto del database

1. Installare l'estensione postgres_fdw.

```
CREATE EXTENSION postgres_fdw;
```

2. Creare un server di dati esterni utilizzando CREATE SERVER.

```
CREATE SERVER foreign_server
FOREIGN DATA WRAPPER postgres_fdw
OPTIONS (host 'xxx.xx.xxx.xx', port '5432', dbname 'foreign_db');
```

3. Creare una mappatura dell'utente per identificare il ruolo da utilizzare sul server remoto.

```
CREATE USER MAPPING FOR local_user
SERVER foreign_server
OPTIONS (user 'foreign_user', password 'password');
```

4. Creare una tabella che esegua la mappatura della tabella sul server remoto.

```
CREATE FOREIGN TABLE foreign_table (
        id integer NOT NULL,
        data text)
SERVER foreign_server
OPTIONS (schema_name 'some_schema', table_name 'some_table');
```

Interazione con i database MySQL utilizzando l'estensione mysql_fdw

Per accedere a un database compatibile con MySQL dall'istanza database RDS per PostgreSQL è possibile installare e utilizzare l'estensione `mysql_fdw`. Questo wrapper di dati esterni consente di interagire con RDS per MySQL, Aurora MySQL, MariaDB e altri database compatibili con MySQL. La connessione dall'istanza database RDS per PostgreSQL al database MySQL è crittografata in base al miglior tentativo a seconda delle configurazioni di client e server. Tuttavia, se lo si desidera, è possibile imporre l'utilizzo della crittografia. Per ulteriori informazioni, consulta Utilizzo della crittografia in transito con l'estensione.

L'estensione `mysql_fdw` è supportata su Amazon RDS per PostgreSQL 14.2, 13.6 e versioni successive. Supporta le operazioni di select, insert, update e delete da un database RDS for PostgreSQL su tabelle contenuto in un'istanza database compatibile con MySQL.

Argomenti

- Configurazione del database RDS per PostgreSQL per l'utilizzo dell'estensione mysql_fdw

- Esempio: utilizzo di un database RDS per MySQL da RDS per PostgreSQL
- Utilizzo della crittografia in transito con l'estensione

Configurazione del database RDS per PostgreSQL per l'utilizzo dell'estensione mysql_fdw

La configurazione dell'estensione `mysql_fdw` sull'istanza database RDS per PostgreSQL comporta il caricamento dell'estensione nell'istanza database e quindi la creazione del punto di connessione all'istanza database MySQL. Per tale attività, è necessario disporre delle seguenti informazioni sull'istanza database MySQL:

- Nome host o endpoint. Per trovare l'endpoint di un'istanza database RDS per MySQL è possibile utilizzare la console. Scegliere la scheda Connectivity & security (Connettività e sicurezza) e cercare nella sezione Endpoint and port (Endpoint e porta).
- Numero della porta. La porta di default per MySQL è 3306.
- Nome del database. L'identificatore del database.

È inoltre necessario fornire l'accesso al gruppo di sicurezza o alla lista di controllo degli accessi (ACL) per la porta MySQL 3306. L'istanza database RDS per PostgreSQL e l'istanza database RDS per MySQL necessitano dell'accesso alla porta 3306. Se l'accesso non è configurato correttamente, quando si cerca di connettersi alla tabella compatibile con MySQL comparirà un messaggio di errore simile al seguente:

```
ERROR: failed to connect to MySQL: Can't connect to MySQL server on 'hostname.aws-
region.rds.amazonaws.com:3306' (110)
```

Nella seguente procedura, l'utente (utilizzando l'account `rds_superuser`) crea il server esterno. Quindi concede l'accesso al server esterno a specifici utenti. Questi utenti creano quindi i propri mapping agli account utente MySQL appropriati per interagire con l'istanza database MySQL.

Per utilizzare mysql_fdw per accedere al server database MySQL

1. Effettuare la connessione all'istanza database PostgreSQL utilizzando un account che dispone del ruolo `rds_superuser`. Se al momento della creazione dell'istanza database RDS per PostgreSQL sono stati accettati i valori predefiniti, il nome utente è `postgres` e lo strumento a riga di comando `psql` può essere usato per collegarsi come segue:

```
psql --host=your-DB-instance.aws-region.rds.amazonaws.com --port=5432 --
username=postgres --password
```

2. Installare l'estensione `mysql_fdw` come segue:

```
postgres=> CREATE EXTENSION mysql_fdw;
CREATE EXTENSION
```

Dopo aver installato l'estensione sull'istanza database RDS per PostgreSQL imposta il server esterno che fornisce la connessione a un database MySQL.

Per creare il server esterno

Esegui queste attività sull'istanza database RDS per PostgreSQL. La procedura presuppone che l'utente sia connesso come utente con i privilegi di `rds_superuser`, come `postgres`.

1. Creazione di un server esterno nell'istanza database RDS for PostgreSQL:

```
postgres=> CREATE SERVER mysql-db FOREIGN DATA WRAPPER mysql_fdw OPTIONS (host 'db-
name.111122223333.aws-region.rds.amazonaws.com', port '3306');
CREATE SERVER
```

2. Concedere agli utenti appropriati l'accesso al server esterno. Questi dovrebbero essere utenti non amministratori, cioè utenti senza il ruolo `rds_superuser`.

```
postgres=> GRANT USAGE ON FOREIGN SERVER mysql-db to user1;
GRANT
```

Gli utenti PostgreSQL creano e gestiscono le proprie connessioni al database MySQL tramite il server esterno.

Esempio: utilizzo di un database RDS per MySQL da RDS per PostgreSQL

Supponi di disporre di una semplice tabella su un'istanza database RDS per PostgreSQL. Gli utenti di RDS per PostgreSQL desiderano eseguire query sugli elementi (`SELECT`), `INSERT`, `UPDATE` e `DELETE` contenute in tale tabella. Supponiamo che l'estensione `mysql_fdw` sia stata creata nell'istanza database RDS for PostgreSQL, come descritto nella procedura precedente. Dopo aver

effettuato la connessione all'istanza database RDS for PostgreSQL come utente con i privilegi `rds_superuser`, è possibile procedere con i seguenti passaggi.

1. Nell'istanza database RDS per PostgreSQL crea un server esterno:

```
test=> CREATE SERVER mysqldb FOREIGN DATA WRAPPER mysql_fdw OPTIONS (host 'your-DB.aws-region.rds.amazonaws.com', port '3306');
CREATE SERVER
```

2. Concedere l'utilizzo a un utente che non dispone delle autorizzazioni `rds_superuser`, ad esempio `user1`:

```
test=> GRANT USAGE ON FOREIGN SERVER mysqldb TO user1;
GRANT
```

3. Connettersi come *`user1`* e quindi creare una mappatura per l'utente MySQL:

```
test=> CREATE USER MAPPING FOR user1 SERVER mysqldb OPTIONS (username 'myuser',
 password 'mypassword');
CREATE USER MAPPING
```

4. Creare di una tabella esterna collegata a una tabella MySQL:

```
test=> CREATE FOREIGN TABLE mytab (a int, b text) SERVER mysqldb OPTIONS (dbname
 'test', table_name '');
CREATE FOREIGN TABLE
```

5. Eseguire una semplice query sulla tabella esterna:

```
test=> SELECT * FROM mytab;
a |   b
---+-------
1 | apple
(1 row)
```

6. È possibile aggiungere, modificare e rimuovere i dati dalla tabella MySQL. Ad esempio:

```
test=> INSERT INTO mytab values (2, 'mango');
INSERT 0 1
```

   Eseguire nuovamente la query `SELECT` per visualizzare i risultati:

```
test=> SELECT * FROM mytab ORDER BY 1;
 a |   b
---+-------
1 | apple
2 | mango
(2 rows)
```

Utilizzo della crittografia in transito con l'estensione

La connessione a MySQL da RDS per PostgreSQL utilizza la crittografia in transito (TLS/SSL) per impostazione predefinita. Tuttavia, la connessione torna a essere non crittografata quando la configurazione di client e server differiscono. È possibile applicare la crittografia a tutte le connessioni in uscita specificando l'opzione `REQUIRE SSL` sugli account utente RDS for MySQL. Lo stesso approccio funziona anche per gli account utente MariaDB e Aurora MySQL.

Per gli account utente MySQL configurati su `REQUIRE SSL`, il tentativo di connessione non riesce se non è possibile stabilire una connessione sicura.

Per applicare la crittografia agli account utente esistenti del database MySQL è possibile utilizzare il comando `ALTER USER`. La sintassi varia a seconda della versione MySQL, come mostrato nella tabella seguente. Per ulteriori informazioni, consultare la voce ALTER USER nel Manuale di riferimento di MySQL.

MySQL 5.7, MySQL 8.0	MySQL 5.6
`ALTER USER 'user'@'%' REQUIRE SSL;`	`GRANT USAGE ON *.* to 'user'@'%' REQUIRE SSL;`

Per ulteriori informazioni sull'estensione `mysql_fdw`, consultare la documentazione di mysql_fdw.

Interazione con un database Oracle utilizzando l'estensione oracle_fdw

Per accedere a un database Oracle dall'istanza database RDS for PostgreSQL è possibile installare e utilizzare l'estensione `oracle_fdw`. Questa estensione è un wrapper di dati esterni per database Oracle. Per ulteriori informazioni sull'estensione, consultare la documentazione di oracle_fdw.

L'estensione `oracle_fdw` è supportata su RDS for PostgreSQL 12.7, 13.3 e versioni successive.

Argomenti

Attivazione dell'estensione oracle_fdw

Per utilizzare l'estensione oracle_fdw, eseguire la procedura riportata di seguito.

Come attivare l'estensione oracle_fdw

- Eseguire il seguente comando utilizzando un account con le autorizzazioni `rds_superuser`.

```
CREATE EXTENSION oracle_fdw;
```

Esempio: utilizzo di un server esterno collegato a un database Amazon RDS for Oracle

L'esempio seguente mostra l'utilizzo di un server esterno collegato a un database Amazon RDS for Oracle.

Come creare un server esterno collegato a un database RDS for Oracle

1. Annotare le seguenti informazioni sull'istanza database RDS for Oracle:
   - Endpoint
   - Porta
   - Nome del database
2. Creare un server esterno.

```
test=> CREATE SERVER oradb FOREIGN DATA WRAPPER oracle_fdw OPTIONS (dbserver
 '//endpoint:port/DB_name');
CREATE SERVER
```

3. Concedere l'utilizzo a un utente che non dispone dei privilegi `rds_superuser`, ad esempio`user1`.

```
test=> GRANT USAGE ON FOREIGN SERVER oradb TO user1;
GRANT
```

4. Connettersi come `user1` e creare una mappatura a un utente Oracle.

```
test=> CREATE USER MAPPING FOR user1 SERVER oradb OPTIONS (user 'oracleuser',
 password 'mypassword');
CREATE USER MAPPING
```

5. Creare una tabella esterna collegata a una tabella Oracle.

```
test=> CREATE FOREIGN TABLE mytab (a int) SERVER oradb OPTIONS (table 'MYTABLE');
CREATE FOREIGN TABLE
```

6. Eseguire una query sulla tabella esterna.

```
test=>  SELECT * FROM mytab;
a
---
1
(1 row)
```

Se la query segnala il seguente errore, controllare il gruppo di sicurezza e la lista di controllo degli accessi (ACL) per assicurarsi che entrambe le istanze possano comunicare.

```
ERROR: connection for foreign table "mytab" cannot be established
DETAIL: ORA-12170: TNS:Connect timeout occurred
```

Utilizzo della crittografia in transito

La crittografia da PostgreSQL a Oracle in transito si basa su una combinazione di parametri di configurazione client e server. Per un esempio di utilizzo di Oracle 21c, consultare Informazioni sui valori per la negoziazione di crittografia e integrità nella documentazione Oracle. Il client utilizzato per oracle_fdw su Amazon RDS è configurato con `ACCEPTED`, il che significa che la crittografia dipende dalla configurazione del database server Oracle.

Se il database si trova su RDS for Oracle, consultare Crittografia di rete nativa Oracle per configurare la crittografia.

Informazioni sulla visualizzazione pg_user_mappings e sulle autorizzazioni

Il catalogo PostgreSQL `pg_user_mapping` archivia la mappatura da un utente RDS for PostgreSQL all'utente in un server remoto di dati esterni. L'accesso al catalogo è limitato, ma puoi usare la visualizzazione `pg_user_mappings` per vedere le mappature. Di seguito è possibile trovare un esempio che mostra come si applicano le autorizzazioni con un database Oracle, sebbene le stesse informazioni si applichino più in generale a qualsiasi wrapper di dati esterno.

Nel seguente output sono presenti ruoli e autorizzazioni mappati su tre diversi utenti di esempio. Gli utenti `rdssu1` e `rdssu2` sono membri del ruolo `rds_superuser`, mentre `user1` non lo è. Nell'esempio viene utilizzato il metacomando `psql` `\du` per elencare i ruoli esistenti.

```
test=>  \du
                                                                 List of roles
    Role name    |                            Attributes                            |
                 Member of
-----------------+------------------------------------------------------------------
+-----------------------------------------------------------
 rdssu1          |                                                                  |
 {rds_superuser}
 rdssu2          |                                                                  |
 {rds_superuser}
 user1           |                                                                  | {}
```

Tutti gli utenti, inclusi gli utenti che godono dei privilegi `rds_superuser`, sono autorizzati a visualizzare le proprie mappature utente (`umoptions`) nella tabella `pg_user_mappings`. Come mostrato nell'esempio seguente, quando `rdssu1` cerca di ottenere tutte le mappature utente, viene generato un errore anche se gode dei privilegi `rdssu1rds_superuser`:

```
test=> SELECT * FROM pg_user_mapping;
ERROR: permission denied for table pg_user_mapping
```

Di seguito vengono riportati alcuni esempi.

```
test=> SET SESSION AUTHORIZATION rdssu1;
SET
test=> SELECT * FROM pg_user_mappings;
 umid  | srvid | srvname | umuser | usename    |            umoptions
-------+-------+---------+--------+------------+---------------------------------
 16414 | 16411 | oradb   |  16412 | user1      |
 16423 | 16411 | oradb   |  16421 | rdssu1     | {user=oracleuser,password=mypwd}
```

```
 16424 | 16411 | oradb   |  16422 | rdssu2     |
 (3 rows)

test=> SET SESSION AUTHORIZATION rdssu2;
SET
test=> SELECT * FROM pg_user_mappings;
 umid  | srvid | srvname | umuser | usename    |            umoptions
-------+-------+---------+--------+------------+---------------------------------
 16414 | 16411 | oradb   |  16412 | user1      |
 16423 | 16411 | oradb   |  16421 | rdssu1     |
 16424 | 16411 | oradb   |  16422 | rdssu2     | {user=oracleuser,password=mypwd}
 (3 rows)

test=> SET SESSION AUTHORIZATION user1;
SET
test=> SELECT * FROM pg_user_mappings;
 umid  | srvid | srvname | umuser | usename    |           umoptions
-------+-------+---------+--------+------------+-------------------------------
 16414 | 16411 | oradb   |  16412 | user1      | {user=oracleuser,password=mypwd}
 16423 | 16411 | oradb   |  16421 | rdssu1     |
 16424 | 16411 | oradb   |  16422 | rdssu2     |
 (3 rows)
```

A causa delle differenze nell'implementazione di `information_schema._pg_user_mappings` e `pg_catalog.pg_user_mappings`, un `rds_superuser` creato manualmente richiede autorizzazioni aggiuntive per visualizzare le password in `pg_catalog.pg_user_mappings`.

Non sono necessarie autorizzazioni aggiuntive per un `rds_superuser` che desideri visualizzare le password in `information_schema._pg_user_mappings`.

Gli utenti che non dispongono del ruolo `rds_superuser` possono visualizzare le password in `pg_user_mappings` solo nelle seguenti condizioni:

- L'utente corrente è l'utente mappato e possiede il server oppure detiene il privilegio `USAGE` su di esso.
- L'utente corrente è il proprietario del server e la mappatura è per `PUBLIC`.

Interazione con i database MySQL utilizzando l'estensione mysql_fdw

È possibile utilizzare l'estensione `tds_fdw` per PostgreSQL per accedere ai database che supportano il protocollo TDS (Tabular Data Stream), ad esempio i database Sybase e Microsoft SQL

Server. Questo wrapper di dati esterni consente di connettersi dalla propria istanza database RDS for PostgreSQL ai database che utilizzano il protocollo TDS, incluso Amazon RDS for Microsoft SQL Server. Per ulteriori informazioni, consultare la documentazione di tds-fdw/tds_fdw su GitHub.

L'estensione `tds_fdw` è supportata su Amazon RDS for PostgreSQL versioni 14.2, 13.6 e successive.

Configurazione del database Aurora PostgreSQL per l'utilizzo dell'estensione tds_fdw

Nelle procedure seguenti, è possibile trovare un esempio di configurazione e utilizzo di `tds_fdw` con un'istanza database RDS for PostgreSQL. Prima di potersi connettere a un database di SQL Server utilizzando `tds_fdw` è necessario disporre delle seguenti informazioni sull'istanza:

- Nome host o endpoint. Per trovare l'endpoint di un'istanza database RDS for SQL Server è possibile utilizzare la console. Scegliere la scheda Connectivity & security (Connettività e sicurezza) e cercare nella sezione Endpoint and port (Endpoint e porta).
- Numero della porta. Il numero di porta predefinito per Microsoft SQL Server è 1433.
- Nome del database. L'identificatore del database.

È inoltre necessario fornire l'accesso al gruppo di sicurezza o alla lista di controllo degli accessi (ACL) per la porta SQL Server 1433. Sia l'istanza database RDS for PostgreSQL che l'istanza database RDS for MySQL Server necessitano dell'accesso alla porta 1433. Se l'accesso non è configurato correttamente, quando si tenta di eseguire una query su Microsoft SQL Server viene visualizzato il seguente messaggio di errore:

```
ERROR: DB-Library error: DB #: 20009, DB Msg: Unable to connect:
Adaptive Server is unavailable or does not exist (mssql2019.aws-
region.rds.amazonaws.com), OS #: 0, OS Msg: Success, Level: 9
```

Per utilizzare tds_fdw per connettersi a un database di SQL Server

1. Collegarsi all'istanza database PostgreSQL utilizzando un account che dispone del ruolo `rds_superuser`:

   ```
   psql --host=your-DB-instance.aws-region.rds.amazonaws.com --port=5432 --
   username=test --password
   ```

2. Installare l'estensione `tds_fdw`:

```
test=> CREATE EXTENSION tds_fdw;
CREATE EXTENSION
```

Dopo che l'estensione è stata installata sull'istanza database RDS for PostgreSQL, è necessario configurare il server esterno.

Per creare il server esterno

Eseguire queste attività sull'istanza database RDS for PostgreSQL utilizzando un account che dispone dei privilegi `rds_superuser`.

1. Creazione di un server esterno nell'istanza database RDS for PostgreSQL:

   ```
   test=> CREATE SERVER sqlserverdb FOREIGN DATA WRAPPER tds_fdw OPTIONS
    (servername 'mssql2019.aws-region.rds.amazonaws.com', port '1433', database
    'tds_fdw_testing');
   CREATE SERVER
   ```

   Per accedere ai dati non ASCII sul lato SQLServer, crea un collegamento server con l'opzione character_set nell'istanza database RDS per PostgreSQL:

   ```
   test=> CREATE SERVER sqlserverdb FOREIGN DATA WRAPPER tds_fdw OPTIONS (servername
    'mssql2019.aws-region.rds.amazonaws.com', port '1433', database 'tds_fdw_testing',
    character_set 'UTF-8');
   CREATE SERVER
   ```

2. Concedere le autorizzazioni a un utente che non dispone del ruolo `rds_superuser`, ad esempio `user1`:

   ```
   test=> GRANT USAGE ON FOREIGN SERVER sqlserverdb TO user1;
   ```

3. Collegarsi come user1 e quindi creare una mappatura per l'utente SQL Server:

   ```
   test=> CREATE USER MAPPING FOR user1 SERVER sqlserverdb OPTIONS (username
    'sqlserveruser', password 'password');
   CREATE USER MAPPING
   ```

4. Creare una tabella esterna collegata a una tabella SQL Server:

```
test=> CREATE FOREIGN TABLE mytab (a int) SERVER sqlserverdb OPTIONS (table
 'MYTABLE');
CREATE FOREIGN TABLE
```

5. Eseguire una query sulla tabella esterna:

```
test=> SELECT * FROM mytab;
 a
---
 1
(1 row)
```

Utilizzo della crittografia in transito per la connessione

La connessione da RDS per PostgreSQL verso SQL Server utilizza la crittografia in transito (TLS/SSL) in base alla configurazione del database SQL Server. Se SQL Server non è configurato per la crittografia, il client RDS per PostgreSQL che effettua la richiesta al database di SQL Server torna a comunicare in modalità non crittografata.

È possibile imporre l'utilizzo della crittografia per la connessione alle istanze database RDS for SQL Server impostando il parametro `rds.force_ssl`. Per scoprire come fare, consultare Imposizione dell'utilizzo di SSL per le connessioni all'istanza database. Per ulteriori informazioni sulla configurazione di SSL/TLS per RDS for SQL Server, consultare Utilizzo di SSL con un'istanza database Microsoft SQL Server.

Utilizzo di Trusted Language Extensions per PostgreSQL

Trusted Language Extensions per PostgreSQL è un kit di sviluppo open source per la creazione di estensioni di PostgreSQL. Ti consente di creare estensioni di PostgreSQL ad alte prestazioni ed eseguirle in sicurezza sull'istanza database RDS per PostgreSQL. Con Trusted Language Extensions (TLE) per PostgreSQL, puoi creare estensioni di PostgreSQL che seguono l'approccio documentato per estendere le funzionalità di PostgreSQL. Per ulteriori informazioni, consulta Packaging Related Objects into an Extension (Creazione di pacchetti di oggetti correlati in un'estensione) nella documentazione PostgreSQL.

Uno dei principali vantaggi di TLE è che è possibile utilizzarlo in ambienti che non forniscono accesso al file system alla base dell'istanza PostgreSQL. In precedenza, l'installazione di una nuova estensione richiedeva l'accesso al file system. TLE rimuove questo vincolo. Fornisce un ambiente di sviluppo per creare nuove estensioni per qualsiasi database PostgreSQL, compresi quelli in esecuzione sulle istanze database RDS per PostgreSQL.

TLE è progettato per impedire l'accesso a risorse non sicure per le estensioni create utilizzando TLE. Il suo ambiente di esecuzione limita l'impatto di qualsiasi difetto dell'estensione a una singola connessione al database. TLE inoltre offre agli amministratori di database un controllo dettagliato su chi può installare le estensioni e fornisce un modello di autorizzazioni per eseguirle.

TLE è supportato sulle seguenti versioni di RDS per PostgreSQL:

- Versione 16.1 e successive 16 versioni
- Versione 15.2 e successive 15 versioni
- Versione 14.5 e successive 14 versioni
- Versione 13.12 e successive 13 versioni

L'ambiente di sviluppo e il runtime di Trusted Language Extensions sono compressi come estensione `pg_tle` di PostgreSQL versione 1.0.1. Supporta la creazione di estensioni in Perl JavaScript, Tcl, PL/pgSQL e SQL. L'estensione `pg_tle` si installa nell'istanza database RDS per PostgreSQL nello stesso modo in cui si installano altre estensioni di PostgreSQL. Dopo l'impostazione di `pg_tle`, gli sviluppatori possono utilizzarlo per creare nuove estensioni di PostgreSQL, note come estensioni TLE.

Negli argomenti seguenti sono disponibili informazioni su come impostare Trusted Language Extensions e come iniziare a creare le proprie estensioni TLE.

Argomenti

Terminology

Per comprendere meglio Trusted Language Extensions, consulta il seguente glossario dei termini usati in questo argomento.

Trusted Language Extensions per PostgreSQL

: Trusted Language Extensions per PostgreSQL è il nome ufficiale del kit di sviluppo open source fornito come estensione `pg_tle`. È disponibile per l'uso su qualsiasi sistema PostgreSQL. Per ulteriori informazioni, consulta aws/pg_tle on. GitHub

Trusted Language Extensions

: Trusted Language Extensions è il nome abbreviato di Trusted Language Extensions per PostgreSQL. Questo nome abbreviato e la sua abbreviazione (TLE) vengono utilizzati anche in questa documentazione.

linguaggio attendibile

: Un linguaggio attendibile è un linguaggio di programmazione o di script con attributi di sicurezza specifici. Ad esempio, i linguaggi attendibili in genere limitano l'accesso al file system e l'uso di proprietà di rete specificate. Il kit di sviluppo TLE è progettato per supportare linguaggi attendibili. PostgreSQL supporta diversi linguaggi utilizzati per creare estensioni attendibili o non attendibili. Per un esempio, vedi Trusted and Untrusted PL/Perl (PL/Perl attendibile e non attendibile) nella

documentazione di PostgreSQL. Quando crei un'estensione utilizzando Trusted Language Extensions, l'estensione utilizza intrinsecamente meccanismi di linguaggio attendibile.

Estensione TLE

Un'estensione TLE è un'estensione di PostgreSQL creata utilizzando il kit di sviluppo Trusted Language Extensions (TLE).

Requisiti per l'utilizzo di Trusted Language Extensions per PostgreSQL

I seguenti sono i requisiti per l'impostazione e l'utilizzo del kit di sviluppo TLE.

- Versioni RDS per PostgreSQL – Trusted Language Extensions supportate su RDS per PostgreSQL versioni 13,12 e versioni 13 successive, 14,5 e versioni 14 successive e 15,2 e solo versioni successive.
  - Per aggiornare l'istanza RDS per PostgreSQL, consulta Aggiornamento del motore del database PostgreSQL per Amazon RDS.
  - Se non disponi ancora di un'istanza database Amazon RDS che esegue PostgreSQL, puoi eseguirne la creazione. Per ulteriori informazioni, consulta un'istanza database RDS per PostgreSQL, consulta Creazione e connessione di un'istanza database PostgreSQL.
- Richiede i privilegi **`rds_superuser`**: per impostare e configurare l'estensione `pg_tle`, il ruolo utente del database deve disporre delle autorizzazioni del ruolo `rds_superuser`. Per impostazione predefinita, questo ruolo viene concesso all'utente `postgres` che crea l'istanza database RDS per PostgreSQL.
- Richiede un gruppo di parametri database personalizzato: è necessario configurare l'istanza database RDS per PostgreSQL con un gruppo di parametri database personalizzato.
  - Se non si configura l'istanza database RDS per PostgreSQL con un gruppo di parametri database personalizzato, è necessario crearne uno e associarlo all'istanza database RDS per PostgreSQL. Per un breve riepilogo dei passaggi, consulta Creazione e applicazione di un gruppo di parametri database personalizzato.
  - Se è già stata eseguita la configurazione dell'istanza database RDS per PostgreSQL utilizzando un gruppo di parametri database personalizzato, puoi impostare Trusted Language Extensions. Per informazioni dettagliate, vedi Impostazione di Trusted Language Extensions nell'istanza database RDS per PostgreSQL.

Creazione e applicazione di un gruppo di parametri database personalizzato

Utilizza i seguenti passaggi per creare un gruppo di parametri database personalizzato e configurare l'istanza database RDS per PostgreSQL per utilizzarlo.

Console

Per creare un gruppo di parametri database personalizzato e utilizzarlo con l'istanza database RDS per PostgreSQL

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel menu di Amazon RDS scegli Parameter groups (Gruppi di parametri).
3. Scegli Create parameter group (Crea gruppo di parametri).
4. Nella pagina Parameter group details (Dettagli del gruppo di parametri) immetti le seguenti informazioni.
   - Per Parameter group family (Famiglia del gruppo di parametri), scegli postgres14.
   - Per Type (Tipo), scegli il gruppo di parametri database.
   - Per Group name (Nome gruppo), assegna al gruppo di parametri un nome significativo nel contesto delle operazioni.
   - In Description (Descrizione), immetti una descrizione utile in modo che gli altri membri del team possano trovarla facilmente.
5. Scegli Crea. Il gruppo di parametri database personalizzato viene creato nella Regione AWS. Ora puoi modificare l'istanza database RDS per PostgreSQL che sarà possibile utilizzare seguendo i prossimi passaggi.
6. Scegli Databases (Database) dal menu Amazon RDS.
7. Scegli l'istanza database RDS per PostgreSQL che desideri utilizzare con TLE tra le opzioni elencate, quindi scegli Modify (Modifica).
8. Nella pagina Modify DB instance settings (Modifica le impostazioni dell'istanza database), trova Database options (Opzioni del database) nella sezione Additional configuration (Configurazione aggiuntiva) e scegli il gruppo di parametri database personalizzato con il selettore.
9. Per salvare la modifica seleziona Continua (Continua).
10. Scegli Apply immediately (Applica immediatamente) in modo da poter continuare a impostare l'istanza database RDS per PostgreSQL per utilizzare TLE.

Per continuare a impostare il sistema per Trusted Language Extensions, consulta Impostazione di Trusted Language Extensions nell'istanza database RDS per PostgreSQL.

Per ulteriori informazioni sull'utilizzo di gruppi di parametri database, consulta Utilizzo di gruppi di parametri DB in un'istanza DB.

AWS CLI

Puoi evitare di specificare l'argomento `--region` quando usi i comandi dell'interfaccia della linea di comando configurando AWS CLI con la Regione AWS predefinita. Per ulteriori informazioni, consulta Nozioni di base sulla configurazione nella Guida per l'utente di AWS Command Line Interface .

Per creare un gruppo di parametri database personalizzato e utilizzarlo con l'istanza database RDS per PostgreSQL

1. Usa il create-db-parameter-group AWS CLI comando per creare un gruppo di parametri DB personalizzato basato su per il tuo. Regione AWS

   macOSUnixPer, o: Linux

   ```
   aws rds create-db-parameter-group \
     --region aws-region \
     --db-parameter-group-name custom-params-for-pg-tle \
     --db-parameter-group-family postgres14 \
     --description "My custom DB parameter group for Trusted Language Extensions"
   ```

   Per Windows:

   ```
   aws rds create-db-parameter-group ^
     --region aws-region ^
     --db-parameter-group-name custom-params-for-pg-tle ^
     --db-parameter-group-family postgres14 ^
     --description "My custom DB parameter group for Trusted Language Extensions"
   ```

   Il gruppo di parametri database personalizzato è disponibile nella Regione AWS, in modo che puoi modificare l'istanza database RDS per PostgreSQL per utilizzarlo.

2. Usa il modify-db-instance AWS CLI comando per applicare il gruppo di parametri DB personalizzato DB. la tua istanza DB RDS per PostgreSQL. Questo comando riavvia immediatamente l'istanza attiva.

PerLinux, o: macOS Unix

```
aws rds modify-db-instance \
  --region aws-region \
  --db-instance-identifier your-instance-name \
  --db-parameter-group-name custom-params-for-pg-tle \
  --apply-immediately
```

Per Windows:

```
aws rds modify-db-instance ^
  --region aws-region ^
  --db-instance-identifier your-instance-name ^
  --db-parameter-group-name custom-params-for-pg-tle ^
  --apply-immediately
```

Per continuare a impostare il sistema per Trusted Language Extensions, consulta Impostazione di Trusted Language Extensions nell'istanza database RDS per PostgreSQL.

Per ulteriori informazioni, consulta Utilizzo di gruppi di parametri.

Impostazione di Trusted Language Extensions nell'istanza database RDS per PostgreSQL

I passaggi seguenti si basano sull'ipotesi che l'istanza database RDS per PostgreSQL sia associata a un gruppo di parametri personalizzato del database. È possibile utilizzare AWS Management Console o AWS CLI per questi passaggi.

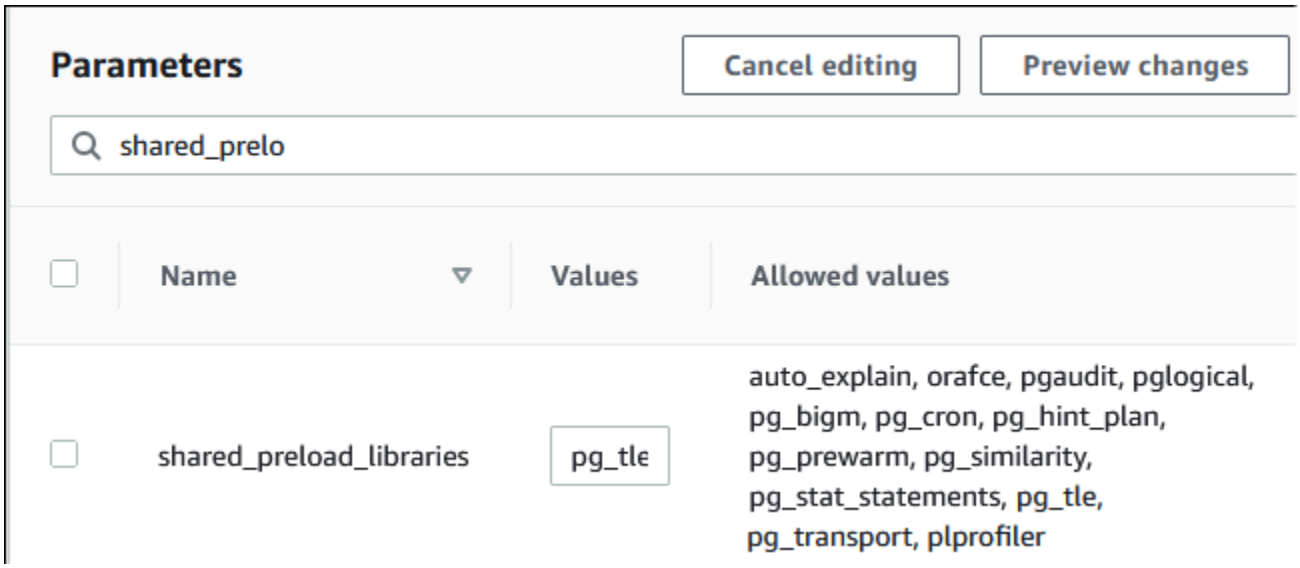
Quando imposti Trusted Language Extensions nell'istanza database RDS per PostgreSQL, lo installi in un database specifico che deve essere utilizzato dagli utenti del database che dispongono delle relative autorizzazioni.

Console

Per impostare Trusted Language Extensions

Esegui i seguenti passaggi utilizzando un account membro del gruppo `rds_superuser` (ruolo).

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, scegli l' istanza database RDS per PostgreSQL.
3. Apri la scheda Configurazione per l' l'istanza database RDS per PostgreSQL. Tra i dettagli dell'istanza, individua il collegamento Parameter group (Gruppo di parametri).
4. Scegli il collegamento per aprire i parametri personalizzati associati l'istanza database RDS per PostgreSQL.
5. Nel campo di ricerca Parametri, digita `shared_pre` per trovare il parametro `shared_preload_libraries`.
6. Scegli Edit parameters (Modifica parametri) per accedere ai valori delle proprietà.
7. Aggiungi `pg_tle` all'elenco nel campo Values (Valori). Utilizza una virgola per separare gli elementi nell'elenco di valori.



8. Riavvia l'istanza database RDS per PostgreSQL in modo che la modifica al parametro `shared_preload_libraries` diventi effettiva.
9. Quando l'istanza è disponibile, verifica che `pg_tle` sia stato inizializzato. Utilizza `psql` per connetterti all'istanza database RDS per PostgreSQL, quindi esegui il comando seguente.

```
SHOW shared_preload_libraries;
shared_preload_libraries
--------------------------
rdsutils,pg_tle
(1 row)
```

10. Con `pg_tle` inizializzato, puoi ora creare l'estensione.

```
CREATE EXTENSION pg_tle;
```

Per verificare che l'estensione sia installata, puoi usare il seguente metacomando `psql`.

```
labdb=> \dx
                          List of installed extensions
  Name   | Version |   Schema   |                Description
---------+---------+------------+-----------------------------------------
 pg_tle  | 1.0.1   | pgtle      | Trusted-Language Extensions for PostgreSQL
 plpgsql | 1.0     | pg_catalog | PL/pgSQL procedural language
```

11. Assegna il ruolo `pgtle_admin` al nome utente principale che hai creato per l'istanza database RDS per PostgreSQL al momento dell'impostazione. Se hai accettato l'impostazione predefinita, il valore è `postgres`.

```
labdb=> GRANT pgtle_admin TO postgres;
GRANT ROLE
```

Per verificare se la concessione è avvenuta, utilizza il metacomando `psql` come illustrato nell'esempio seguente. Nell'output vengono visualizzati solo i ruoli `pgtle_admin` e `postgres`. Per ulteriori informazioni, consulta Comprendere il ruolo rds_superuser.

```
labdb=> \du
                           List of roles
    Role name    |           Attributes            |               Member of
-----------------+---------------------------------
+-----------------------------------
pgtle_admin     | Cannot login                     | {}
postgres        | Create role, Create DB          +| {rds_superuser,pgtle_admin}
                | Password valid until infinity    |...
```

12. Chiudi la sessione `psql` usando il metacomando `\q`.

```
\q
```

Per iniziare a creare le estensioni TLE, consulta Esempio: creazione di un'estensione Trusted Language Extensions utilizzando SQL.

AWS CLI

Puoi evitare di specificare l'argomento `--region` quando usi i comandi CLI configurando AWS CLI con la Regione AWS predefinita. Per ulteriori informazioni, consulta Nozioni di base sulla configurazione nella Guida per l'utente di AWS Command Line Interface .

Per impostare Trusted Language Extensions

1. Usa il modify-db-parameter-group AWS CLI comando per aggiungere `pg_tle` al `shared_preload_libraries` parametro.

```
aws rds modify-db-parameter-group \
   --db-parameter-group-name custom-param-group-name \
   --parameters
 "ParameterName=shared_preload_libraries,ParameterValue=pg_tle,ApplyMethod=pending-
reboot" \
   --region aws-region
```

2. Usa il reboot-db-instance AWS CLI comando per riavviare l' e inizializzare la libreria. `pg_tle`

```
aws rds reboot-db-instance \
    --db-instance-identifier your-instance \
    --region aws-region
```

3. Quando l'istanza è disponibile, verifica che `pg_tle` sia stato inizializzato. Utilizza `psql` per connetterti all'istanza database RDS per PostgreSQL, quindi esegui il comando seguente.

```
SHOW shared_preload_libraries;
shared_preload_libraries
--------------------------
rdsutils,pg_tle
(1 row)
```

   Con `pg_tle` inizializzato, puoi ora creare l'estensione.

```
CREATE EXTENSION pg_tle;
```

4. Assegna il ruolo `pgtle_admin` al nome utente principale che hai creato per l'istanza database RDS per PostgreSQL al momento dell'impostazione. Se hai accettato l'impostazione predefinita, il valore è `postgres`.

```
GRANT pgtle_admin TO postgres;
GRANT ROLE
```

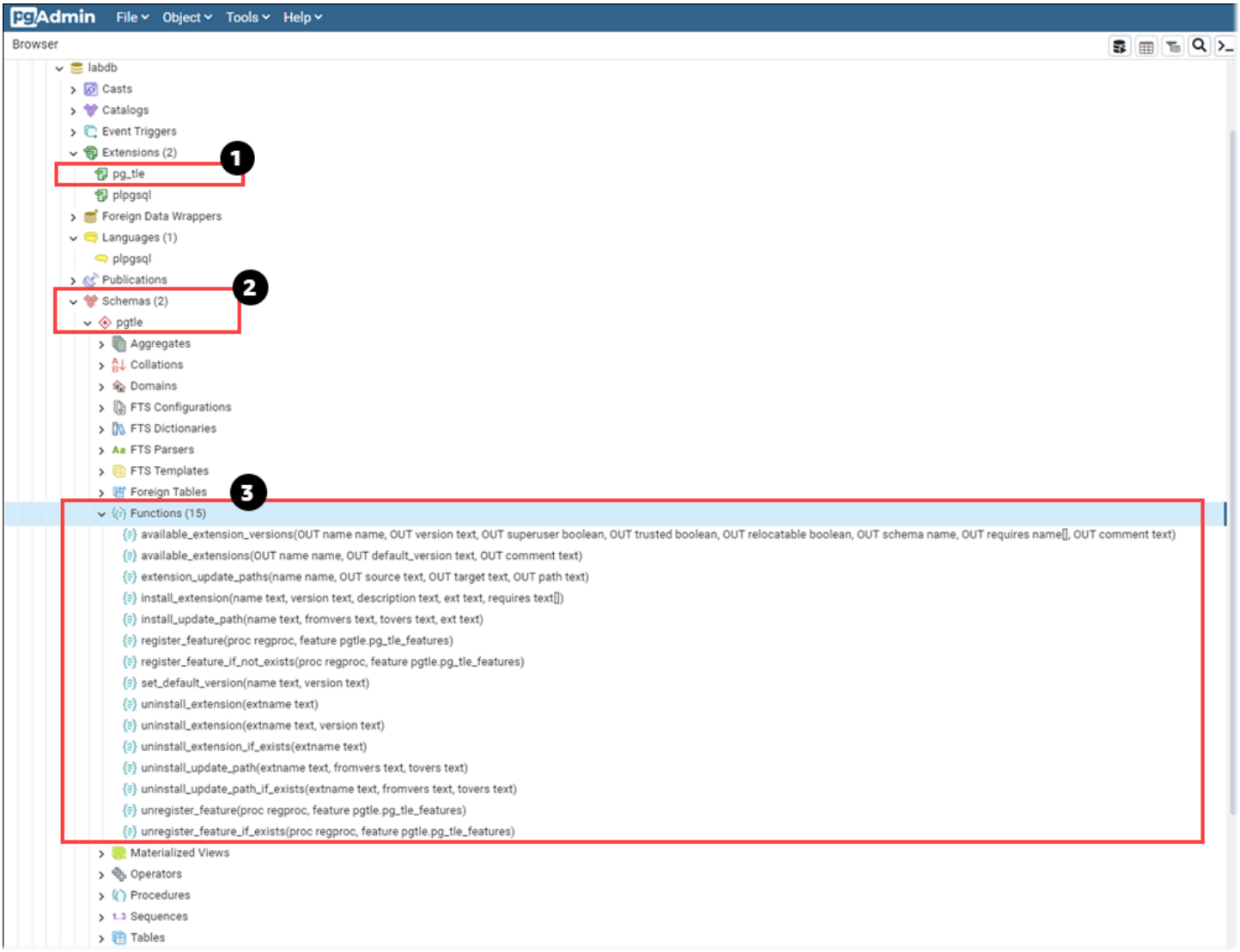
5. Chiudi la sessione `psql` come indicato di seguito.

```
labdb=> \q
```

Per iniziare a creare le estensioni TLE, consulta Esempio: creazione di un'estensione Trusted Language Extensions utilizzando SQL.

Panoramica di Trusted Language Extensions per PostgreSQL

Trusted Language Extensions per PostgreSQL è un'estensione di PostgreSQL che si installa nell'istanza database RDS per PostgreSQL nello stesso modo in cui si impostano le altre estensioni di PostgreSQL. Nell'immagine seguente di un database di esempio nello strumento client pgAdmin, è possibile vedere alcuni dei componenti che compongono l'estensione `pg_tle`.



È possibile vedere i dettagli riportati di seguito.

1. Il kit di sviluppo Trusted Language Extensions (TLE) per PostgreSQL è fornito nel pacchetto come estensione `pg_tle`. Pertanto, `pg_tle` viene aggiunto alle estensioni disponibili per il database in cui è installato.
2. TLE ha un proprio schema, `pgtle`. Questo schema contiene funzioni helper (3) per l'installazione e la gestione delle estensioni create.
3. TLE offre oltre una dozzina di funzioni helper per l'installazione, la registrazione e la gestione delle estensioni. Per ulteriori informazioni su queste funzioni, consulta Riferimento per le funzioni per Trusted Language Extensions per PostgreSQL.

Altri componenti dell'estensione `pg_tle` sono:

- Il ruolo **`pgtle_admin`**: il ruolo `pgtle_admin` viene creato quando viene installata l'estensione `pg_tle`. Questo ruolo include privilegi e deve essere trattato come tale. Ti consigliamo vivamente di seguire il principio del privilegio minimo quando concedi il ruolo `pgtle_admin` agli utenti del database. In altre parole, concedi il ruolo `pgtle_admin` solo agli utenti del database autorizzati a creare, installare e gestire nuove estensioni TLE, ad esempio `postgres`.
- La tabella **`pgtle.feature_info`**: la tabella `pgtle.feature_info` è una tabella protetta che contiene informazioni sulle estensioni TLE, sugli hook e sulle stored procedure e funzioni personalizzate che utilizzano. Se disponi di privilegi `pgtle_admin`, usa le seguenti funzioni Trusted Language Extensions per aggiungere e aggiornare le informazioni nella tabella.
  - pgtle.register_feature
  - pgtle.register_feature_if_not_exists
  - pgtle.unregister_feature
  - pgtle.unregister_feature_if_exists

Creazione di estensioni TLE per RDS per PostgreSQL

È possibile installare qualsiasi estensione creata con TLE in qualsiasi istanza database RDS per PostgreSQL in cui è installata l'estensione `pg_tle`. L'estensione `pg_tle` si riferisce al database PostgreSQL in cui è installata. Le estensioni create utilizzando TLE si riferiscono allo stesso database.

Usa le varie funzioni `pgtle` per installare il codice che costituisce la tua estensione TLE. Le seguenti funzioni di Trusted Language Extensions richiedono tutte il ruolo `pgtle_admin`.

- pgtle.install_extension
- pgtle.install_update_path
- pgtle.register_feature
- pgtle.register_feature_if_not_exists
- pgtle.set_default_version
- pgtle.uninstall_extension(name)
- pgtle.uninstall_extension(name, version)
- pgtle.uninstall_extension_if_exists
- pgtle.uninstall_update_path
- pgtle.uninstall_update_path_if_exists

- pgtle.unregister_feature
- pgtle.unregister_feature_if_exists

Esempio: creazione di un'estensione Trusted Language Extensions utilizzando SQL

L'esempio seguente mostra come creare un'estensione TLE denominata `pg_distance` che contiene alcune funzioni SQL per il calcolo delle distanze utilizzando formule diverse. Nell'elenco, puoi trovare la funzione per il calcolo della distanza di Manhattan e la funzione per il calcolo della distanza euclidea. Per ulteriori informazioni sulla differenza tra queste formule, consulta Taxicab geometry (Geometria del taxi) e Euclidean geometry (Geometria euclidea) in Wikipedia.

È possibile utilizzare questo esempio nell'istanza database RDS per PostgreSQL se l'estensione `pg_tle` è impostata come descritto in dettaglio in Impostazione di Trusted Language Extensions nell'istanza database RDS per PostgreSQL.

> **Note**
>
> È necessario disporre dei privilegi del ruolo `pgtle_admin` per seguire questa procedura.

Per creare l'estensione TLE di esempio

I passaggi seguenti utilizzano un database di esempio denominato `labdb`. Questo database è di proprietà dell'utente `postgres` principale. Il ruolo `postgres` dispone anche delle autorizzazioni del ruolo `pgtle_admin`.

1. Utilizza `psql` per connetterti l'istanza database RDS per PostgreSQL.

   ```
   psql --host=db-instance-123456789012.aws-region.rds.amazonaws.com
   --port=5432 --username=postgres --password --dbname=labdb
   ```

2. Crea un'estensione TLE denominata `pg_distance` copiando il seguente codice e incollandolo nella console della sessione `psql`.

   ```
   SELECT pgtle.install_extension
   (
    'pg_distance',
    '0.1',
     'Distance functions for two points',
   $_pg_tle_$
   ```

```
    CREATE FUNCTION dist(x1 float8, y1 float8, x2 float8, y2 float8, norm int)
    RETURNS float8
    AS $$
      SELECT (abs(x2 - x1) ^ norm + abs(y2 - y1) ^ norm) ^ (1::float8 / norm);
    $$ LANGUAGE SQL;

    CREATE FUNCTION manhattan_dist(x1 float8, y1 float8, x2 float8, y2 float8)
    RETURNS float8
    AS $$
      SELECT dist(x1, y1, x2, y2, 1);
    $$ LANGUAGE SQL;

    CREATE FUNCTION euclidean_dist(x1 float8, y1 float8, x2 float8, y2 float8)
    RETURNS float8
    AS $$
      SELECT dist(x1, y1, x2, y2, 2);
    $$ LANGUAGE SQL;
$_pg_tle_$
);
```

Viene visualizzato l'output riportato di seguito.

```
install_extension
---------------
 t
(1 row)
```

Gli artefatti che costituiscono l'estensione `pg_distance` sono ora installati nel database. Questi artefatti includono il file di controllo e il codice dell'estensione, che devono essere presenti in modo che l'estensione possa essere creata utilizzando il comando `CREATE EXTENSION`. In altre parole, è comunque necessario creare l'estensione per rendere le funzioni disponibili agli utenti del database.

3. Per creare l'estensione, usa il comando `CREATE EXTENSION` come per qualsiasi altra estensione. Come per altre estensioni, l'utente del database deve disporre delle autorizzazioni `CREATE` nel database.

```
CREATE EXTENSION pg_distance;
```

4. Per testare l'estensione TLE `pg_distance`, puoi usarla per calcolare la distanza di Manhattan tra quattro punti.

```
labdb=> SELECT manhattan_dist(1, 1, 5, 5);
8
```

Per calcolare la distanza euclidea tra lo stesso set di punti, puoi usare quanto segue.

```
labdb=> SELECT euclidean_dist(1, 1, 5, 5);
5.656854249492381
```

L'estensione `pg_distance` carica le funzioni nel database e le rende disponibili a tutti gli utenti con le autorizzazioni per il database.

Modifica dell'estensione TLE

Per migliorare le prestazioni delle query per le funzioni contenute nell'estensione TLE, aggiungi i seguenti due attributi PostgreSQL alle specifiche.

- `IMMUTABLE`: l'attributo `IMMUTABLE` garantisce che l'ottimizzatore di query possa utilizzare le ottimizzazioni per migliorare i tempi di risposta delle query. Per ulteriori informazioni, consulta Function Volatility Categories (Categorie della volatilità delle funzioni) nella documentazione di PostgreSQL.
- `PARALLEL SAFE`: l'attributo `PARALLEL SAFE` è un altro attributo che consente a PostgreSQL di eseguire la funzione in modalità parallela. Per ulteriori informazioni, consultare CREATE FUNCTION nella documentazione di PostgreSQL.

Nell'esempio seguente, puoi vedere come viene utilizzata la funzione `pgtle.install_update_path` per aggiungere questi attributi a ogni funzione per creare la versione `0.2` dell'estensione TLE `pg_distance`. Per ulteriori informazioni su questa funzione, consulta pgtle.install_update_path. È necessario avere il ruolo `pgtle_admin` necessario per eseguire questa operazione.

Per aggiornare un'estensione TLE esistente e specificare la versione predefinita

1. Esegui la connessione all'istanza database RDS per PostgreSQL utilizzando `psql` o un altro strumento client, ad esempio pgAdmin.

```
psql --host=db-instance-123456789012.aws-region.rds.amazonaws.com
```

```
--port=5432 --username=postgres --password --dbname=labdb
```

2. Modifica l'estensione TLE esistente copiando il seguente codice e incollandolo nella console della sessione `psql`.

```
SELECT pgtle.install_update_path
(
 'pg_distance',
 '0.1',
 '0.2',
$_pg_tle_$
    CREATE OR REPLACE FUNCTION dist(x1 float8, y1 float8, x2 float8, y2 float8,
 norm int)
    RETURNS float8
    AS $$
      SELECT (abs(x2 - x1) ^ norm + abs(y2 - y1) ^ norm) ^ (1::float8 / norm);
    $$ LANGUAGE SQL IMMUTABLE PARALLEL SAFE;

    CREATE OR REPLACE FUNCTION manhattan_dist(x1 float8, y1 float8, x2 float8, y2
 float8)
    RETURNS float8
    AS $$
      SELECT dist(x1, y1, x2, y2, 1);
    $$ LANGUAGE SQL IMMUTABLE PARALLEL SAFE;

    CREATE OR REPLACE FUNCTION euclidean_dist(x1 float8, y1 float8, x2 float8, y2
 float8)
    RETURNS float8
    AS $$
      SELECT dist(x1, y1, x2, y2, 2);
    $$ LANGUAGE SQL IMMUTABLE PARALLEL SAFE;
$_pg_tle_$
);
```

Viene visualizzata una risposta simile alla seguente.

```
install_update_path
--------------------
 t
(1 row)
```

È possibile impostare questa versione dell'estensione come versione predefinita, in modo che gli utenti del database non debbano specificare una versione quando creano o aggiornano l'estensione nel database.

3. Per specificare che la versione modificata (versione 0.2) dell'estensione TLE è la versione predefinita, usa la funzione `pgtle.set_default_version` come mostrato nell'esempio seguente.

```
SELECT pgtle.set_default_version('pg_distance', '0.2');
```

Per ulteriori informazioni su questa funzione, consulta pgtle.set_default_version.

4. Una volta inserito il codice, puoi aggiornare l'estensione TLE installata nel modo consueto, usando il comando `ALTER EXTENSION ... UPDATE`, come mostrato di seguito:

```
ALTER EXTENSION pg_distance UPDATE;
```

Eliminazione delle estensioni TLE da un database

Puoi eliminare le estensioni TLE usando il comando `DROP EXTENSION` nello stesso modo che impieghi per le altre estensioni di PostgreSQL. L'eliminazione dell'estensione non rimuove i file di installazione che costituiscono l'estensione, il che consente agli utenti di ricrearla. Per rimuovere l'estensione e i relativi file di installazione, esegui la seguente procedura in due passaggi.

Per eliminare l'estensione TLE e rimuovere i file di installazione

1. Utilizza `psql` o un altro strumento cliente per connetterti all'istanza database RDS per PostgreSQL.

```
psql --host=.111122223333.aws-region.rds.amazonaws.com --port=5432 --
username=postgres --password --dbname=dbname
```

2. Elimina l'estensione come faresti per qualsiasi estensione di PostgreSQL.

```
DROP EXTENSION your-TLE-extension
```

Ad esempio, se crei l'estensione `pg_distance` come descritto in Esempio: creazione di un'estensione Trusted Language Extensions utilizzando SQL, puoi eliminarla come segue.

```
DROP EXTENSION pg_distance;
```

Viene visualizzato l'output che conferma che l'estensione è stata eliminata, come segue.

```
DROP EXTENSION
```

A questo punto, l'estensione non è più attiva nel database. Tuttavia, i file di installazione e il file di controllo sono ancora disponibili nel database, quindi gli utenti del database possono creare nuovamente l'estensione, se lo desiderano.

- Se vuoi lasciare intatti i file delle estensioni in modo che gli utenti del database possano creare l'estensione TLE, puoi fermarti qui.
- Se desideri rimuovere tutti i file che costituiscono l'estensione, continua con il passaggio successivo.

3. Per rimuovere tutti i file di installazione per l'estensione, usa la funzione `pgtle.uninstall_extension`. Questa funzione rimuove tutto il codice e i file di controllo dell'estensione.

```
SELECT pgtle.uninstall_extension('your-tle-extension-name');
```

Ad esempio, per rimuovere tutti i file di installazione `pg_distance`, utilizza il comando seguente.

```
SELECT pgtle.uninstall_extension('pg_distance');
 uninstall_extension
---------------------
 t
(1 row)
```

Disinstallazione di Trusted Language Extensions per PostgreSQL

Se non desideri più creare le estensioni TLE utilizzando TLE, puoi eliminare l'estensione `pg_tle` e rimuovere tutti gli artefatti. Questa azione include l'eliminazione di qualsiasi estensione TLE nel database e l'eliminazione dello schema `pgtle`.

Per eliminare l'estensione **`pg_tle`** e il relativo schema da un database

1. Utilizza `psql` o un altro strumento cliente per connetterti all'istanza database RDS per PostgreSQL.

   ```
   psql --host=.111122223333.aws-region.rds.amazonaws.com --port=5432 --
   username=postgres --password --dbname=dbname
   ```

2. Elimina l'estensione `pg_tle` dal database. Se il database ha le estensioni TLE ancora in esecuzione nel database, devi eliminare anche quelle estensioni. A questo scopo, puoi utilizzare la parola chiave `CASCADE`, come illustrato di seguito.

   ```
   DROP EXTENSION pg_tle CASCADE;
   ```

   Se l'estensione `pg_tle` non è ancora attiva nel database, non è necessario utilizzare la parola chiave `CASCADE`.

3. Elimina lo schema `pgtle`. Questa azione rimuove tutte le funzioni di gestione dal database.

   ```
   DROP SCHEMA pgtle CASCADE;
   ```

   Il comando restituisce quanto segue al termine del processo.

   ```
   DROP SCHEMA
   ```

   L'estensione `pg_tle`, il relativo schema, le funzioni e tutti gli artefatti vengono rimossi. Per creare nuove estensioni utilizzando TLE, ripeti il processo di impostazione. Per ulteriori informazioni, consulta Impostazione di Trusted Language Extensions nell'istanza database RDS per PostgreSQL.

Utilizzo di hook PostgreSQL con le estensioni TLE

Un hook è un meccanismo di callback disponibile in PostgreSQL che consente agli sviluppatori di chiamare funzioni personalizzate o altre routine durante le normali operazioni del database. Il kit di sviluppo TLE supporta gli hook PostgreSQL per poter integrare le funzioni personalizzate con il comportamento di PostgreSQL in fase di esecuzione. Ad esempio, puoi utilizzare un hook per associare il processo di autenticazione al codice personalizzato o per modificare il processo di pianificazione ed esecuzione delle query in base alle tue esigenze specifiche.

Le estensioni TLE possono utilizzare gli hook. Se l'ambito dell'hook è globale, si applica a tutti i database. Pertanto, se l'estensione TLE utilizza un hook globale, è necessario crearla in tutti i database a cui gli utenti possono accedere.

Quando usi `pg_tle` per creare le estensioni Trusted Language Extensions, puoi utilizzare gli hook disponibili da un'API SQL per creare le funzioni della tua estensione. È necessario registrare gli hook con `pg_tle`. Per alcuni hook, potrebbe essere necessario impostare anche vari parametri di configurazione. Ad esempio, l'hook di controllo `passcode` può essere impostato su attivo, disattivo oppure obbligatorio. Per ulteriori informazioni sui requisiti specifici per gli hook `pg_tle` disponibili, consulta Riferimento per gli hook per Trusted Language Extensions per PostgreSQL.

Esempio: creazione di un'estensione che utilizza un hook PostgreSQL

L'esempio discusso in questa sezione utilizza un hook PostgreSQL per controllare la password fornita durante specifiche operazioni SQL e impedisce agli utenti del database di impostare le proprie password su una qualsiasi di quelle contenute nella tabella `password_check.bad_passwords`. La tabella contiene le prime dieci opzioni di password più utilizzate, ma facilmente violabili.

Per configurare questo esempio nell'istanza database RDS per PostgreSQL, devi aver già installato Trusted Language Extensions. Per informazioni dettagliate, vedi Impostazione di Trusted Language Extensions nell'istanza database RDS per PostgreSQL.

Per configurare l'esempio dell'hook di controllo della password

1. Utilizza `psql` per connetterti l'istanza database RDS per PostgreSQL.

```
psql --host=db-instance-123456789012.aws-region.rds.amazonaws.com
--port=5432 --username=postgres --password --dbname=labdb
```

2. Copia il codice da Codice di hook di controllo della password e incollalo nel database.

```
SELECT pgtle.install_extension (
  'my_password_check_rules',
  '1.0',
  'Do not let users use the 10 most commonly used passwords',
$_pgtle_$
  CREATE SCHEMA password_check;
  REVOKE ALL ON SCHEMA password_check FROM PUBLIC;
  GRANT USAGE ON SCHEMA password_check TO PUBLIC;

  CREATE TABLE password_check.bad_passwords (plaintext) AS
```

```
 VALUES
   ('123456'),
   ('password'),
   ('12345678'),
   ('qwerty'),
   ('123456789'),
   ('12345'),
   ('1234'),
   ('111111'),
   ('1234567'),
   ('dragon');
 CREATE UNIQUE INDEX ON password_check.bad_passwords (plaintext);

 CREATE FUNCTION password_check.passcheck_hook(username text, password text,
password_type pgtle.password_types, valid_until timestamptz, valid_null boolean)
 RETURNS void AS $$
   DECLARE
     invalid bool := false;
   BEGIN
     IF password_type = 'PASSWORD_TYPE_MD5' THEN
       SELECT EXISTS(
         SELECT 1
         FROM password_check.bad_passwords bp
         WHERE ('md5' || md5(bp.plaintext || username)) = password
       ) INTO invalid;
       IF invalid THEN
         RAISE EXCEPTION 'Cannot use passwords from the common password
dictionary';
       END IF;
     ELSIF password_type = 'PASSWORD_TYPE_PLAINTEXT' THEN
       SELECT EXISTS(
         SELECT 1
         FROM password_check.bad_passwords bp
         WHERE bp.plaintext = password
       ) INTO invalid;
       IF invalid THEN
         RAISE EXCEPTION 'Cannot use passwords from the common common password
dictionary';
       END IF;
     END IF;
   END
 $$ LANGUAGE plpgsql SECURITY DEFINER;

 GRANT EXECUTE ON FUNCTION password_check.passcheck_hook TO PUBLIC;
```

```
  SELECT pgtle.register_feature('password_check.passcheck_hook', 'passcheck');
$_pgtle_$
);
```

Quando l'estensione è stata caricata nel database, viene visualizzato un output come il seguente.

```
 install_extension
-------------------
 t
(1 row)
```

3. Mentre sei ancora connesso al database, puoi creare l'estensione.

```
CREATE EXTENSION my_password_check_rules;
```

4. È possibile confermare che l'estensione è stata creata nel database utilizzando il seguente metacomando `psql`.

```
\dx
                        List of installed extensions
          Name           | Version |   Schema   |
 Description
-------------------------+---------+------------
+-------------------------------------------------------------
 my_password_check_rules | 1.0     | public     | Prevent use of any of the top-ten
 most common bad passwords
 pg_tle                  | 1.0.1   | pgtle      | Trusted-Language Extensions for
 PostgreSQL
 plpgsql                 | 1.0     | pg_catalog | PL/pgSQL procedural language
(3 rows)
```

5. Apri un' AWS CLI altra sessione di terminale per lavorare con. È necessario modificare il gruppo di parametri database personalizzato per attivare l'hook di controllo della password. A tale scopo, utilizzate il comando modify-db-parameter-groupCLI come illustrato nell'esempio seguente.

```
aws rds modify-db-parameter-group \
    --region aws-region \
    --db-parameter-group-name your-custom-parameter-group \
```

```
    --parameters
 "ParameterName=pgtle.enable_password_check,ParameterValue=on,ApplyMethod=immediate"
```

Quando il parametro viene attivato correttamente, viene visualizzato un output come il seguente.

```
(
    "DBParameterGroupName": "docs-lab-parameters-for-tle"
}
```

Per rendere effettive le modifiche all'impostazione del gruppo di parametri possono essere necessari alcuni minuti. Tuttavia, questo parametro è dinamico, quindi non è necessario riavviare l'istanza database RDS per PostgreSQL perché l'impostazione diventi effettiva.

6. Apri la sessione `psql` ed esegui una query sul database per verificare che l'hook di controllo della password sia stato attivato.

```
labdb=> SHOW pgtle.enable_password_check;
pgtle.enable_password_check
----------------------------
on
(1 row)
```

L'hook di controllo della password è ora attivo. Puoi testarlo creando un nuovo ruolo e utilizzando una delle password errate, come illustrato nell'esempio seguente.

```
CREATE ROLE test_role PASSWORD 'password';
ERROR:  Cannot use passwords from the common password dictionary
CONTEXT:  PL/pgSQL function
 password_check.passcheck_hook(text,text,pgtle.password_types,timestamp with time
 zone,boolean) line 21 at RAISE
SQL statement "SELECT password_check.passcheck_hook(
    $1::pg_catalog.text,
    $2::pg_catalog.text,
    $3::pgtle.password_types,
    $4::pg_catalog.timestamptz,
    $5::pg_catalog.bool)"
```

L'output è stato formattato per ragioni di leggibilità.

L'esempio seguente mostra che il comportamento `\password` del metacomando interattivo `pgsql` è influenzato anche dall'hook di controllo della password.

```
postgres=> SET password_encryption TO 'md5';
SET
postgres=> \password
Enter new password for user "postgres":*****
Enter it again:*****
ERROR:  Cannot use passwords from the common password dictionary
CONTEXT:  PL/pgSQL function
 password_check.passcheck_hook(text,text,pgtle.password_types,timestamp with time
 zone,boolean) line 12 at RAISE
SQL statement "SELECT password_check.passcheck_hook($1::pg_catalog.text,
 $2::pg_catalog.text, $3::pgtle.password_types, $4::pg_catalog.timestamptz,
 $5::pg_catalog.bool)"
```

Puoi eliminare questa estensione TLE e disinstallare i file di origine, se lo desideri. Per ulteriori informazioni, consulta Eliminazione delle estensioni TLE da un database.

Codice di hook di controllo della password

Il codice di esempio mostrato qui definisce le specifiche per l'estensione TLE `my_password_check_rules`. Quando copi questo codice e lo incolli nel database, il codice dell'estensione `my_password_check_rules` viene caricato nel database e l'hook `password_check` viene registrato per essere utilizzato dall'estensione.

```
SELECT pgtle.install_extension (
  'my_password_check_rules',
  '1.0',
  'Do not let users use the 10 most commonly used passwords',
$_pgtle_$
  CREATE SCHEMA password_check;
  REVOKE ALL ON SCHEMA password_check FROM PUBLIC;
  GRANT USAGE ON SCHEMA password_check TO PUBLIC;

  CREATE TABLE password_check.bad_passwords (plaintext) AS
  VALUES
    ('123456'),
    ('password'),
    ('12345678'),
    ('qwerty'),
    ('123456789'),
```

```
    ('12345'),
    ('1234'),
    ('111111'),
    ('1234567'),
    ('dragon');
  CREATE UNIQUE INDEX ON password_check.bad_passwords (plaintext);

  CREATE FUNCTION password_check.passcheck_hook(username text, password text,
 password_type pgtle.password_types, valid_until timestamptz, valid_null boolean)
  RETURNS void AS $$
    DECLARE
      invalid bool := false;
    BEGIN
      IF password_type = 'PASSWORD_TYPE_MD5' THEN
        SELECT EXISTS(
          SELECT 1
          FROM password_check.bad_passwords bp
          WHERE ('md5' || md5(bp.plaintext || username)) = password
        ) INTO invalid;
        IF invalid THEN
          RAISE EXCEPTION 'Cannot use passwords from the common password dictionary';
        END IF;
      ELSIF password_type = 'PASSWORD_TYPE_PLAINTEXT' THEN
        SELECT EXISTS(
          SELECT 1
          FROM password_check.bad_passwords bp
          WHERE bp.plaintext = password
        ) INTO invalid;
        IF invalid THEN
          RAISE EXCEPTION 'Cannot use passwords from the common common password
 dictionary';
        END IF;
      END IF;
    END
  $$ LANGUAGE plpgsql SECURITY DEFINER;

  GRANT EXECUTE ON FUNCTION password_check.passcheck_hook TO PUBLIC;

  SELECT pgtle.register_feature('password_check.passcheck_hook', 'passcheck');
$_pgtle_$
);
```

Utilizzo dei tipi di dati personalizzati in TLE

PostgreSQL supporta comandi per registrare nuovi tipi di base (noti anche come tipi scalari) per gestire in modo efficiente strutture di dati complesse nel database. Un tipo di base consente di personalizzare il modo in cui i dati vengono archiviati internamente e come convertirli da e verso una rappresentazione testuale esterna. Questi tipi di dati personalizzati sono utili per estendere il supporto di PostgreSQL ai domini funzionali in cui un tipo integrato come numero o testo non può fornire una semantica di ricerca sufficiente.

RDS per PostgreSQL consente di creare tipi di dati personalizzati in Trusted Language Extensions e di definire funzioni che supportano le operazioni SQL e di indicizzazione per questi nuovi tipi di dati. I tipi di dati personalizzati sono disponibili per le seguenti versioni:

- RDS per PostgreSQL 15.4 e versioni successive alla 15
- RDS per PostgreSQL 14.9 e versioni successive alla 14
- RDS per PostgreSQL 13.12 e versioni successive alla 13

Per ulteriori informazioni, consulta la pagina Trusted Language Base types.

Riferimento per le funzioni per Trusted Language Extensions per PostgreSQL

Esamina la seguente documentazione di riferimento sulle funzioni disponibili in Trusted Language Extensions per PostgreSQL. Usa queste funzioni per installare, registrare, aggiornare e gestire le tue estensioni TLE, ovvero le estensioni PostgreSQL che sviluppi utilizzando il kit di sviluppo Trusted Language Extensions.

Argomenti

- pgtle.available_extensions
- pgtle.available_extension_versions
- pgtle.extension_update_paths
- pgtle.install_extension
- pgtle.install_update_path
- pgtle.register_feature
- pgtle.register_feature_if_not_exists

- pgtle.set_default_version
- pgtle.uninstall_extension(name)
- pgtle.uninstall_extension(name, version)
- pgtle.uninstall_extension_if_exists
- pgtle.uninstall_update_path
- pgtle.uninstall_update_path_if_exists
- pgtle.unregister_feature
- pgtle.unregister_feature_if_exists

pgtle.available_extensions

La funzione `pgtle.available_extensions` è progettata per restituire un set. Restituisce tutte le estensioni TLE disponibili nel database. Ogni riga restituita contiene informazioni su una singola estensione TLE.

Prototipo di funzione

```
pgtle.available_extensions()
```

Ruolo

Nessuna.

Argomenti

Nessuna.

Output

- `name`: il nome dell'estensione TLE.
- `default_version`: la versione dell'estensione TLE da usare quando la funzione `CREATE EXTENSION` viene chiamata senza una versione specifica.
- `description`: una descrizione più dettagliata dell'estensione TLE.

Esempio di utilizzo

```
SELECT * FROM pgtle.available_extensions();
```

pgtle.available_extension_versions

La funzione `available_extension_versions` è progettata per restituire un set. Restituisce l'elenco di tutte le estensioni TLE disponibili e le relative versioni. Ogni riga contiene informazioni su una versione specifica dell'estensione TLE indicata, incluso se è richiesto un ruolo specifico.

Prototipo di funzione

```
pgtle.available_extension_versions()
```

Ruolo

Nessuna.

Argomenti

Nessuna.

Output

- `name`: il nome dell'estensione TLE.
- `version`: la versione dell'estensione TLE.
- `superuser`: questo valore è sempre `false` per le estensioni TLE. Le autorizzazioni necessarie per creare o aggiornare l'estensione TLE sono uguali a quelle per creare altri oggetti nel database specificato.
- `trusted`: questo valore è sempre `false` per un'estensione TLE.
- `relocatable`: questo valore è sempre `false` per un'estensione TLE.
- `schema`: specifica il nome dello schema in cui è installata l'estensione TLE.
- `requires`: un array contenente i nomi di altre estensioni necessarie a questa estensione TLE.
- `description`: una descrizione dettagliata dell'estensione TLE.

Per ulteriori informazioni sulle estensioni PostgreSQL, consulta Packaging Related Objects into an Extension > Extension Files (Creazione di pacchetti di oggetti correlati in un'estensione > File di estensione) nella documentazione PostgreSQL.

Esempio di utilizzo

```
SELECT * FROM pgtle.available_extension_versions();
```

pgtle.extension_update_paths

La funzione `extension_update_paths` è progettata per restituire un set. Restituisce l'elenco di tutti i possibili percorsi di aggiornamento per un'estensione TLE. Ogni riga include gli aggiornamenti o i downgrade disponibili per l'estensione TLE.

Prototipo di funzione

```
pgtle.extension_update_paths(name)
```

Ruolo

Nessuna.

Argomenti

`name`: il nome dell'estensione TLE da cui ottenere i percorsi di aggiornamento.

Output

- `source`: la versione di origine di un aggiornamento.
- `target`: la versione di destinazione di un aggiornamento.
- `path`: il percorso di aggiornamento utilizzato per aggiornare un'estensione TLE dalla versione `source` alla versione `target`, ad esempio`0.1--0.2`.

Esempio di utilizzo

```
SELECT * FROM pgtle.extension_update_paths('your-TLE');
```

pgtle.install_extension

La funzione `install_extension` consente di installare gli artefatti che costituiscono l'estensione TLE nel database, dopodiché può essere creata utilizzando il comando `CREATE EXTENSION`.

Prototipo di funzione

```
pgtle.install_extension(name text, version text, description text, ext text, requires
 text[] DEFAULT NULL::text[])
```

Ruolo

Nessuna.

Argomenti

- `name`: il nome dell'estensione TLE. Questo valore viene utilizzato per la chiamata `CREATE EXTENSION`.
- `version`: la versione dell'estensione TLE.
- `description`: una descrizione dettagliata dell'estensione TLE. Questa descrizione viene visualizzata nel campo `comment` in `pgtle.available_extensions()`.
- `ext`: il contenuto dell'estensione TLE. Questo valore include gli oggetti, come le funzioni.
- `requires`: un parametro facoltativo che specifica le dipendenze per l'estensione TLE. L'estensione `pg_tle` viene aggiunta automaticamente come dipendenza.

Molti di questi argomenti sono uguali a quelli inclusi in un file di controllo delle estensioni per l'installazione di un'estensione di PostgreSQL nel file system di un'istanza PostgreSQL. Per ulteriori informazioni, consulta Extension Files (File di estensione) in Packaging Related Objects into an Extension (Creazione di pacchetti di oggetti correlati in un'estensione) nella documentazione PostgreSQL.

Output

Questa funzione restituisce `OK` in caso di esito positivo e `NULL` in caso di errore.

- `OK`: l'estensione TLE è stata installata correttamente nel database.
- `NULL`: l'estensione TLE non è stata installata correttamente nel database.

Esempio di utilizzo

```
SELECT pgtle.install_extension(
 'pg_tle_test',
 '0.1',
 'My first pg_tle extension',
$_pgtle_$
  CREATE FUNCTION my_test()
  RETURNS INT
  AS $$
```

```
    SELECT 42;
  $$ LANGUAGE SQL IMMUTABLE;
$_pgtle_$
);
```

pgtle.install_update_path

La funzione `install_update_path` fornisce il percorso di aggiornamento tra due diverse versioni di un'estensione TLE. Questa funzione consente agli utenti dell'estensione TLE di aggiornarne la versione utilizzando la sintassi `ALTER EXTENSION ... UPDATE`.

Prototipo di funzione

```
pgtle.install_update_path(name text, fromvers text, tovers text, ext text)
```

Ruolo

`pgtle_admin`

Argomenti

- `name`: il nome dell'estensione TLE. Questo valore viene utilizzato per la chiamata `CREATE EXTENSION`.
- `fromvers`: la versione di origine dell'estensione TLE per l'aggiornamento.
- `tovers`: la versione di destinazione dell'estensione TLE per l'aggiornamento.
- `ext`: i contenuti dell'aggiornamento. Questo valore include gli oggetti, come le funzioni.

Output

Nessuna.

Esempio di utilizzo

```
SELECT pgtle.install_update_path('pg_tle_test', '0.1', '0.2',
  $_pgtle_$
    CREATE OR REPLACE FUNCTION my_test()
    RETURNS INT
    AS $$
      SELECT 21;
    $$ LANGUAGE SQL IMMUTABLE;
```

```
  $_pgtle_$
);
```

pgtle.register_feature

La funzione `register_feature` aggiunge la funzionalità PostgreSQL interna specificata alla tabella `pgtle.feature_info`. Gli hook PostgreSQL sono un esempio di funzionalità interna di PostgreSQL. Il kit di sviluppo Trusted Language Extensions supporta l'uso degli hook PostgreSQL. Attualmente, questa funzione supporta la seguente funzionalità.

- `passcheck`: registra l'hook di verifica della password con la procedura o la funzione che personalizza il comportamento di verifica della password di PostgreSQL.

Prototipo di funzione

```
pgtle.register_feature(proc regproc, feature pg_tle_feature)
```

Ruolo

`pgtle_admin`

Argomenti

- `proc`: il nome di una procedura o funzione memorizzata da utilizzare per la funzionalità.
- `feature`: il nome della funzionalità `pg_tle` (ad esempio `passcheck`) da registrare con la funzione.

Output

Nessuna.

Esempio di utilizzo

```
SELECT pgtle.register_feature('pw_hook', 'passcheck');
```

pgtle.register_feature_if_not_exists

La funzione `pgtle.register_feature_if_not_exists` aggiunge la funzionalità PostgreSQL specificata alla tabella `pgtle.feature_info` e identifica l'estensione TLE o un'altra procedura

o funzione che utilizza la funzionalità. Per ulteriori informazioni sugli hook e su Trusted Language Extensions, consulta Utilizzo di hook PostgreSQL con le estensioni TLE.

Prototipo di funzione

```
pgtle.register_feature_if_not_exists(proc regproc, feature pg_tle_feature)
```

Ruolo

`pgtle_admin`

Argomenti

- `proc`: il nome di una stored procedure o una funzione che contiene la logica (codice) da utilizzare come funzionalità dell'estensione TLE. Ad esempio, il codice `pw_hook`.
- `feature`: il nome della funzionalità PostgreSQL da registrare per la funzione TLE. Attualmente, l'unica funzionalità disponibile è l'hook `passcheck`. Per ulteriori informazioni, consulta Hook di verifica della password (passcheck).

Output

Restituisce `true` dopo aver registrato la funzionalità per l'estensione specificata. Restituisce `false` se la funzionalità è già registrata.

Esempio di utilizzo

```
SELECT pgtle.register_feature_if_not_exists('pw_hook', 'passcheck');
```

pgtle.set_default_version

La funzione `set_default_version` ti consente di specificare una `default_version` per la tua estensione TLE. È possibile utilizzare questa funzione per definire un percorso di aggiornamento e designare la versione come predefinita per l'estensione TLE. Quando gli utenti del database specificano l'estensione TLE nei comandi `CREATE EXTENSION` e `ALTER EXTENSION ... UPDATE`, la versione specificata dell'estensione TLE viene creata nel database per tali utenti.

Questa funzione restituisce `true` in caso di esito positivo. Se l'estensione TLE specificata nell'argomento `name` non esiste, la funzione restituisce un errore. Analogamente, se la `version` dell'estensione TLE non esiste, la funzione restituisce un errore.

Prototipo di funzione

```
pgtle.set_default_version(name text, version text)
```

Ruolo

`pgtle_admin`

Argomenti

- `name`: il nome dell'estensione TLE. Questo valore viene utilizzato per la chiamata `CREATE EXTENSION`.
- `version`: la versione dell'estensione TLE da impostare come predefinita.

Output

- `true`: quando l'impostazione della versione predefinita ha esito positivo, la funzione restituisce `true`.
- `ERROR`: restituisce un messaggio di errore se non esiste un'estensione TLE con il nome o la versione specificati.

Esempio di utilizzo

```
SELECT * FROM pgtle.set_default_version('my-extension', '1.1');
```

pgtle.uninstall_extension(name)

La funzione `uninstall_extension` rimuove tutte le versioni di un'estensione TLE da un database. Questa funzione impedisce alle future chiamate `CREATE EXTENSION` di installare l'estensione TLE. Se l'estensione TLE non esiste nel database, viene generato un errore.

La funzione `uninstall_extension` non elimina un'estensione TLE attualmente attiva nel database. Per rimuovere un'estensione TLE attualmente attiva, è necessario chiamare esplicitamente `DROP EXTENSION` per rimuoverla.

Prototipo di funzione

```
pgtle.uninstall_extension(extname text)
```

Ruolo

`pgtle_admin`

Argomenti

- `extname`: il nome dell'estensione TLE da disinstallare. Questo nome è quello utilizzato con `CREATE EXTENSION` per caricare l'estensione TLE da usare in un determinato database.

Output

Nessuna.

Esempio di utilizzo

```
SELECT * FROM pgtle.uninstall_extension('pg_tle_test');
```

pgtle.uninstall_extension(name, version)

La funzione `uninstall_extension(name, version)` rimuove la versione specificata dell'estensione TLE dal database. Questa funzione impedisce alle chiamate `CREATE EXTENSION` e `ALTER EXTENSION` di installare o aggiornare un'estensione TLE alla versione specificata. Questa funzione rimuove anche tutti i percorsi di aggiornamento per la versione specificata dell'estensione TLE. La funzione non disinstalla l'estensione TLE se è attualmente attiva nel database. È necessario chiamare esplicitamente `DROP EXTENSION` per rimuovere l'estensione TLE. Per disinstallare tutte le versioni di un'estensione TLE, consulta pgtle.uninstall_extension(name).

Prototipo di funzione

```
pgtle.uninstall_extension(extname text, version text)
```

Ruolo

`pgtle_admin`

Argomenti

- `extname`: il nome dell'estensione TLE. Questo valore viene utilizzato per la chiamata `CREATE EXTENSION`.

- `version`: la versione dell'estensione TLE da disinstallare dal database.

Output

Nessuna.

Esempio di utilizzo

```
SELECT * FROM pgtle.uninstall_extension('pg_tle_test', '0.2');
```

pgtle.uninstall_extension_if_exists

La funzione `uninstall_extension_if_exists` rimuove tutte le versioni di un'estensione TLE da un determinato database. Se l'estensione TLE non esiste, la funzione non restituisce alcun avviso (non viene generato alcun messaggio di errore). Se l'estensione specificata è attualmente attiva in un database, non viene eliminata dalla funzione. È necessario chiamare esplicitamente `DROP EXTENSION` per rimuovere l'estensione TLE prima di utilizzare questa funzione per disinstallarne gli artefatti.

Prototipo di funzione

```
pgtle.uninstall_extension_if_exists(extname text)
```

Ruolo

`pgtle_admin`

Argomenti

- `extname`: il nome dell'estensione TLE. Questo valore viene utilizzato per la chiamata `CREATE EXTENSION`.

Output

La funzione `uninstall_extension_if_exists` restituisce `true` dopo aver disinstallato l'estensione specificata. Se l'estensione specificata non esiste, la funzione restituisce `false`.

- `true`: restituisce `true` dopo aver disinstallato l'estensione TLE.

- `false`: restituisce `false` quando l'estensione TLE non esiste nel database.

Esempio di utilizzo

```
SELECT * FROM pgtle.uninstall_extension_if_exists('pg_tle_test');
```

pgtle.uninstall_update_path

La funzione `uninstall_update_path` rimuove il percorso di aggiornamento specificato da un'estensione TLE. In tal modo `ALTER EXTENSION ... UPDATE TO` non può utilizzarlo come percorso di aggiornamento.

Se l'estensione TLE è attualmente utilizzata da una delle versioni di questo percorso di aggiornamento, rimane nel database.

Se il percorso di aggiornamento specificato non esiste, la funzione genera un errore.

Prototipo di funzione

```
pgtle.uninstall_update_path(extname text, fromvers text, tovers text)
```

Ruolo

`pgtle_admin`

Argomenti

- `extname`: il nome dell'estensione TLE. Questo valore viene utilizzato per la chiamata `CREATE EXTENSION`.
- `fromvers`: la versione di origine dell'estensione TLE utilizzata nel percorso di aggiornamento.
- `tovers`: la versione di destinazione dell'estensione TLE utilizzata nel percorso di aggiornamento.

Output

Nessuna.

Esempio di utilizzo

```
SELECT * FROM pgtle.uninstall_update_path('pg_tle_test', '0.1', '0.2');
```

pgtle.uninstall_update_path_if_exists

La funzione `uninstall_update_path_if_exists` è simile a `uninstall_update_path` in quanto rimuove il percorso di aggiornamento specificato da un'estensione TLE. Tuttavia, se il percorso di aggiornamento non esiste, questa funzione non genera un messaggio di errore e restituisce `false`.

Prototipo di funzione

```
pgtle.uninstall_update_path_if_exists(extname text, fromvers text, tovers text)
```

Ruolo

`pgtle_admin`

Argomenti

- `extname`: il nome dell'estensione TLE. Questo valore viene utilizzato per la chiamata `CREATE EXTENSION`.
- `fromvers`: la versione di origine dell'estensione TLE utilizzata nel percorso di aggiornamento.
- `tovers`: la versione di destinazione dell'estensione TLE utilizzata nel percorso di aggiornamento.

Output

- `true`: la funzione ha aggiornato correttamente il percorso dell'estensione TLE.
- `false`: la funzione non è stata in grado di aggiornare il percorso dell'estensione TLE.

Esempio di utilizzo

```
SELECT * FROM pgtle.uninstall_update_path_if_exists('pg_tle_test', '0.1', '0.2');
```

pgtle.unregister_feature

La funzione `unregister_feature` fornisce un modo per rimuovere le funzioni registrate per utilizzare la funzionalità `pg_tle`, ad esempio gli hook. Per ulteriori informazioni sulla registrazione di una funzionalità, consulta pgtle.register_feature.

Prototipo di funzione

```
pgtle.unregister_feature(proc regproc, feature pg_tle_features)
```

Ruolo

`pgtle_admin`

Argomenti

- `proc`: il nome di una funzione memorizzata da registrare con una funzionalità `pg_tle`.
- `feature`: il nome della funzionalità `pg_tle` da registrare con la funzione. Ad esempio, `passcheck` è una funzionalità che può essere registrata per essere utilizzata dalle estensioni Trusted Language Extensions sviluppate. Per ulteriori informazioni, consulta Hook di verifica della password (passcheck).

Output

Nessuna.

Esempio di utilizzo

```
SELECT * FROM pgtle.unregister_feature('pw_hook', 'passcheck');
```

pgtle.unregister_feature_if_exists

La funzione `unregister_feature` fornisce un modo per rimuovere le funzioni registrate per utilizzare la funzionalità `pg_tle`, ad esempio gli hook. Per ulteriori informazioni, consulta Utilizzo di hook PostgreSQL con le estensioni TLE. Restituisce `true` dopo aver completato l'annullamento della registrazione della funzionalità. Restituisce `false` se la funzionalità non è stata registrata.

Per informazioni sulla registrazione delle funzionalità `pg_tle` per le estensioni TLE, consulta pgtle.register_feature.

Prototipo di funzione

```
pgtle.unregister_feature_if_exists('proc regproc', 'feature pg_tle_features')
```

Ruolo

`pgtle_admin`

Argomenti

- `proc`: il nome della funzione memorizzata che è stata registrata per includere una funzionalità `pg_tle`.
- `feature`: il nome della funzionalità `pg_tle` registrata con l'estensione Trusted Language Extensions.

Output

Restituisce `true` o `false`, come indicato di seguito.

- `true`: la funzione ha completato l'annullamento della registrazione della funzionalità dall'estensione.
- `false`: la funzione non è stata in grado di annullare la registrazione della funzionalità dall'estensione TLE.

Esempio di utilizzo

```
SELECT * FROM pgtle.unregister_feature_if_exists('pw_hook', 'passcheck');
```

Riferimento per gli hook per Trusted Language Extensions per PostgreSQL

Trusted Language Extensions per PostgreSQL supporta gli hook PostgreSQL. Un hook è un meccanismo di callback interno che gli sviluppatori possono usare per estendere le funzionalità di base di PostgreSQL. Utilizzando gli hook, gli sviluppatori possono implementare le proprie funzioni o procedure da utilizzare durante varie operazioni del database, modificando così il comportamento di PostgreSQL in qualche modo. Ad esempio, puoi usare un hook `passcheck` per personalizzare il modo in cui PostgreSQL gestisce le password fornite durante la creazione o la modifica delle password per gli utenti (ruoli).

Esamina la seguente documentazione per conoscere gli hook disponibili per le tue estensioni TLE.

Argomenti

- Hook di verifica della password (passcheck)

Hook di verifica della password (passcheck)

L'hook `passcheck` viene utilizzato per personalizzare il comportamento di PostgreSQL durante il processo di verifica della password per i seguenti comandi SQL e il metacomando `psql`.

- `CREATE ROLE username ...PASSWORD`: per ulteriori informazioni, consulta CREATE ROLE nella documentazione di PostgreSQL.
- `ALTER ROLE username...PASSWORD`: per ulteriori informazioni, consulta ALTER ROLE nella documentazione di PostgreSQL.
- `\password username`: questo metacomando `psql` interattivo modifica in modo sicuro la password per l'utente specificato eseguendo l'hashing della password prima di utilizzare in modo trasparente la sintassi `ALTER ROLE ... PASSWORD`. Il metacomando è un wrapper sicuro per il comando `ALTER ROLE ... PASSWORD`, quindi l'hook si applica al comportamento del metacomando `psql`.

Per vedere un esempio, consulta Codice di hook di controllo della password.

Prototipo di funzione

```
passcheck_hook(username text, password text, password_type pgtle.password_types,
 valid_until timestamptz, valid_null boolean)
```

Argomenti

La funzione dell'hook `passcheck` accetta i seguenti argomenti:

- `username`: il nome (come testo) del ruolo (nome utente) che imposta una password.
- `password`: la password in chiaro o con hash. La password immessa deve corrispondere al tipo specificato in `password_type`.
- `password_type`: specifica il formato `pgtle.password_type` della password, che può essere costituito da una delle seguenti opzioni.
  - `PASSWORD_TYPE_PLAINTEXT`: una password in testo semplice.
  - `PASSWORD_TYPE_MD5`: una password che è stata sottoposta a hash utilizzando l'algoritmo MD5 (message digest 5).
  - `PASSWORD_TYPE_SCRAM_SHA_256`: una password che è stata sottoposta a hash utilizzando l'algoritmo SCRAM-SHA-256.

- `valid_until`: specifica l'ora in cui la password diventa non valida. Questo argomento è facoltativo. Se utilizzi questo argomento, specifica l'ora come valore `timestamptz`.
- `valid_null`: se questo booleano è impostato su `true`, l'opzione `valid_until` è impostata su `NULL`.

Configurazione

La funzione `pgtle.enable_password_check` controlla se l'hook passcheck è attivo. L'hook passcheck ha tre possibili impostazioni.

- `off`: disattiva l'hook `passcheck` di verifica della password. Si tratta del valore di default.
- on: attiva l'hook `passcode` di verifica della password in modo che le password vengano confrontate con la tabella.
- `require`: richiede la definizione di un hook di verifica della password.

Note per l'utilizzo

Per attivare o disattivare l'hook `passcheck`, è necessario modificare il gruppo di parametri database personalizzato per l'istanza database RDS per PostgreSQL.

Per LinuxmacOS, oUnix:

```
aws rds modify-db-parameter-group \
    --region aws-region \
    --db-parameter-group-name your-custom-parameter-group \
    --parameters
 "ParameterName=pgtle.enable_password_check,ParameterValue=on,ApplyMethod=immediate"
```

Per Windows:

```
aws rds modify-db-parameter-group ^
    --region aws-region ^
    --db-parameter-group-name your-custom-parameter-group ^
    --parameters
 "ParameterName=pgtle.enable_password_check,ParameterValue=on,ApplyMethod=immediate"
```

Esempi di codice per Amazon RDS con SDK AWS

I seguenti esempi di codice mostrano come usare Amazon RDS con un kit di sviluppo AWS software (SDK).

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le operazioni mostrino come richiamare le singole funzioni del servizio, è possibile visualizzarle contestualizzate negli scenari correlati e negli esempi tra servizi.

Scenari: esempi di codice che mostrano come eseguire un'attività specifica richiamando più funzioni all'interno dello stesso servizio.

Esempi cross-service: applicazioni di esempio che funzionano su più servizi Servizi AWS.

Per un elenco completo di guide ed esempi di codice per sviluppatori AWS SDK, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Nozioni di base

Hello Amazon RDS

Gli esempi di codice seguenti mostrano come iniziare a utilizzare Amazon RDS.

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
using System;
using System.Threading.Tasks;
using Amazon.RDS;
using Amazon.RDS.Model;

namespace RDSActions;
```

```
public static class HelloRds
{
    static async Task Main(string[] args)
    {
        var rdsClient = new AmazonRDSClient();

        Console.WriteLine($"Hello Amazon RDS! Following are some of your DB
 instances:");
        Console.WriteLine();

        // You can use await and any of the async methods to get a response.
        // Let's get the first twenty DB instances.
        var response = await rdsClient.DescribeDBInstancesAsync(
            new DescribeDBInstancesRequest()
            {
                MaxRecords = 20 // Must be between 20 and 100.
            });

        foreach (var instance in response.DBInstances)
        {
            Console.WriteLine($"\tDB name: {instance.DBName}");
            Console.WriteLine($"\tArn: {instance.DBInstanceArn}");
            Console.WriteLine($"\tIdentifier: {instance.DBInstanceIdentifier}");
            Console.WriteLine();
        }
    }
}
```

- Per informazioni dettagliate sull'API, consulta DescribeDBInstances nella Documentazione di riferimento delle API di AWS SDK for .NET .

C++

SDK per C++

> **Note**
>
> C'è dell'altro GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Codice per il file CMake C MakeLists .txt.

```
# Set the minimum required version of CMake for this project.
cmake_minimum_required(VERSION 3.13)

# Set the AWS service components used by this project.
set(SERVICE_COMPONENTS rds)

# Set this project's name.
project("hello_rds")

# Set the C++ standard to use to build this target.
# At least C++ 11 is required for the AWS SDK for C++.
set(CMAKE_CXX_STANDARD 11)

# Use the MSVC variable to determine if this is a Windows build.
set(WINDOWS_BUILD ${MSVC})

if (WINDOWS_BUILD) # Set the location where CMake can find the installed
 libraries for the AWS SDK.
    string(REPLACE ";" "/aws-cpp-sdk-all;" SYSTEM_MODULE_PATH
 "${CMAKE_SYSTEM_PREFIX_PATH}/aws-cpp-sdk-all")
    list(APPEND CMAKE_PREFIX_PATH ${SYSTEM_MODULE_PATH})
endif ()

# Find the AWS SDK for C++ package.
find_package(AWSSDK REQUIRED COMPONENTS ${SERVICE_COMPONENTS})

if (WINDOWS_BUILD AND AWSSDK_INSTALL_AS_SHARED_LIBS)
     # Copy relevant AWS SDK for C++ libraries into the current binary directory
 for running and debugging.

     # set(BIN_SUB_DIR "/Debug") # If you are building from the command line, you
 may need to uncomment this
                                 # and set the proper subdirectory to the
 executables' location.

     AWSSDK_CPY_DYN_LIBS(SERVICE_COMPONENTS ""
 ${CMAKE_CURRENT_BINARY_DIR}${BIN_SUB_DIR})
endif ()

add_executable(${PROJECT_NAME}
        hello_rds.cpp)
```

```
target_link_libraries(${PROJECT_NAME}
        ${AWSSDK_LINK_LIBRARIES})
```

Codice per il file origine hello_rds.cpp.

```
#include <aws/core/Aws.h>
#include <aws/rds/RDSClient.h>
#include <aws/rds/model/DescribeDBInstancesRequest.h>
#include <iostream>

/*
 *  A "Hello Rds" starter application which initializes an Amazon Relational
 Database Service (Amazon RDS) client and
 *  describes the Amazon RDS instances.
 *
 *  main function
 *
 *  Usage: 'hello_rds'
 *
 */

int main(int argc, char **argv) {
    Aws::SDKOptions options;
    // Optionally change the log level for debugging.
//   options.loggingOptions.logLevel = Utils::Logging::LogLevel::Debug;
    Aws::InitAPI(options); // Should only be called once.
    int result = 0;
    {
        Aws::Client::ClientConfiguration clientConfig;
        // Optional: Set to the AWS Region (overrides config file).
        // clientConfig.region = "us-east-1";

        Aws::RDS::RDSClient rdsClient(clientConfig);
        Aws::String marker;
        std::vector<Aws::String> instanceDBIDs;

        do {
            Aws::RDS::Model::DescribeDBInstancesRequest request;

            if (!marker.empty()) {
                request.SetMarker(marker);
            }
```

```
            Aws::RDS::Model::DescribeDBInstancesOutcome outcome =
                    rdsClient.DescribeDBInstances(request);

            if (outcome.IsSuccess()) {
                for (auto &instance: outcome.GetResult().GetDBInstances()) {
                    instanceDBIDs.push_back(instance.GetDBInstanceIdentifier());
                }
                marker = outcome.GetResult().GetMarker();
            } else {
                result = 1;
                std::cerr << "Error with RDS::DescribeDBInstances. "
                          << outcome.GetError().GetMessage()
                          << std::endl;
                break;
            }
        } while (!marker.empty());

        std::cout << instanceDBIDs.size() << " RDS instances found." <<
 std::endl;
        for (auto &instanceDBID: instanceDBIDs) {
            std::cout << "   Instance: " << instanceDBID << std::endl;
        }
    }

    Aws::ShutdownAPI(options); // Should only be called once.
    return result;
}
```

- Per informazioni dettagliate sull'API, consulta DescribeDBInstances nella Documentazione di riferimento delle API di AWS SDK for C++ .

Go

SDK per Go V2

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
package main

import (
 "context"
 "fmt"

 "github.com/aws/aws-sdk-go-v2/aws"
 "github.com/aws/aws-sdk-go-v2/config"
 "github.com/aws/aws-sdk-go-v2/service/rds"
)

// main uses the AWS SDK for Go V2 to create an Amazon Relational Database
 Service (Amazon RDS)
// client and list up to 20 DB instances in your account.
// This example uses the default settings specified in your shared credentials
// and config files.
func main() {
 sdkConfig, err := config.LoadDefaultConfig(context.TODO())
 if err != nil {
  fmt.Println("Couldn't load default configuration. Have you set up your AWS
 account?")
  fmt.Println(err)
  return
 }
 rdsClient := rds.NewFromConfig(sdkConfig)
 const maxInstances = 20
 fmt.Printf("Let's list up to %v DB instances.\n", maxInstances)
 output, err := rdsClient.DescribeDBInstances(context.TODO(),
  &rds.DescribeDBInstancesInput{MaxRecords: aws.Int32(maxInstances)})
 if err != nil {
  fmt.Printf("Couldn't list DB instances: %v\n", err)
  return
 }
 if len(output.DBInstances) == 0 {
  fmt.Println("No DB instances found.")
 } else {
  for _, instance := range output.DBInstances {
   fmt.Printf("DB instance %v has database %v.\n",
 *instance.DBInstanceIdentifier,
    *instance.DBName)
  }
 }
```

```
}
```

- Per informazioni dettagliate sull'API, consulta DescribeDBInstances nella Documentazione di riferimento delle API di AWS SDK for Go .

Java

SDK per Java 2.x

> **Note**
>
> C'è dell'altro GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.rds.RdsClient;
import software.amazon.awssdk.services.rds.model.DescribeDbInstancesResponse;
import software.amazon.awssdk.services.rds.model.DBInstance;
import software.amazon.awssdk.services.rds.model.RdsException;
import java.util.List;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class DescribeDBInstances {

    public static void main(String[] args) {
        Region region = Region.US_EAST_1;
        RdsClient rdsClient = RdsClient.builder()
                .region(region)
                .build();
```

```
        describeInstances(rdsClient);
        rdsClient.close();
    }

    public static void describeInstances(RdsClient rdsClient) {
        try {
            DescribeDbInstancesResponse response =
 rdsClient.describeDBInstances();
            List<DBInstance> instanceList = response.dbInstances();
            for (DBInstance instance : instanceList) {
                System.out.println("Instance ARN is: " +
 instance.dbInstanceArn());
                System.out.println("The Engine is " + instance.engine());
                System.out.println("Connection endpoint is" +
 instance.endpoint().address());
            }

        } catch (RdsException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }
    }
}
```

- Per informazioni dettagliate sull'API, consulta DescribeDBInstances nella Documentazione di riferimento delle API di AWS SDK for Java 2.x .

Esempi di codice

- Azioni per Amazon RDS tramite SDK AWS
  - Utilizzo CreateDBInstance con un AWS SDK o una CLI
  - Utilizzo CreateDBParameterGroup con un AWS SDK o una CLI
  - Utilizzo CreateDBSnapshot con un AWS SDK o una CLI
  - Utilizzo DeleteDBInstance con un AWS SDK o una CLI
  - Utilizzo DeleteDBParameterGroup con un AWS SDK o una CLI
  - Utilizzo DescribeAccountAttributes con un AWS SDK o una CLI
  - Utilizzo DescribeDBEngineVersions con un AWS SDK o una CLI
  - Utilizzo DescribeDBInstances con un AWS SDK o una CLI

- Utilizzo DescribeDBParameterGroups con un AWS SDK o una CLI
- Utilizzo DescribeDBParameters con un AWS SDK o una CLI
- Utilizzo DescribeDBSnapshots con un AWS SDK o una CLI
- Utilizzo DescribeOrderableDBInstanceOptions con un AWS SDK o una CLI
- Utilizzo GenerateRDSAuthToken con un AWS SDK o una CLI
- Utilizzo ModifyDBInstance con un AWS SDK o una CLI
- Utilizzo ModifyDBParameterGroup con un AWS SDK o una CLI
- Utilizzo RebootDBInstance con un AWS SDK o una CLI

- Scenari per Amazon RDS che utilizzano SDK AWS
  - Inizia a usare le istanze DB di Amazon RDS utilizzando un SDK AWS
- Esempi serverless per Amazon RDS che utilizzano SDK AWS
  - Connessione a un database Amazon RDS in una funzione Lambda
- Esempi di servizi multipli per Amazon RDS che utilizzano SDK AWS
  - Creazione di un tracciatore di elementi di lavoro di Aurora Serverless

Azioni per Amazon RDS tramite SDK AWS

I seguenti esempi di codice mostrano come eseguire singole azioni Amazon RDS con gli AWS SDK. Questi estratti chiamano l'API Amazon RDS e sono estratti di codice da programmi più grandi che devono essere eseguiti in modo contestuale. Ogni esempio include un collegamento a GitHub, dove puoi trovare le istruzioni per la configurazione e l'esecuzione del codice.

Gli esempi seguenti includono solo le operazioni più comunemente utilizzate. Per un elenco completo, consulta la documentazione di riferimento dell'API Amazon Relational Database Service (Amazon RDS).

Esempi

- Utilizzo CreateDBInstance con un AWS SDK o una CLI
- Utilizzo CreateDBParameterGroup con un AWS SDK o una CLI
- Utilizzo CreateDBSnapshot con un AWS SDK o una CLI
- Utilizzo DeleteDBInstance con un AWS SDK o una CLI
- Utilizzo DeleteDBParameterGroup con un AWS SDK o una CLI

- Utilizzo DescribeAccountAttributes con un AWS SDK o una CLI
- Utilizzo DescribeDBEngineVersions con un AWS SDK o una CLI
- Utilizzo DescribeDBInstances con un AWS SDK o una CLI
- Utilizzo DescribeDBParameterGroups con un AWS SDK o una CLI
- Utilizzo DescribeDBParameters con un AWS SDK o una CLI
- Utilizzo DescribeDBSnapshots con un AWS SDK o una CLI
- Utilizzo DescribeOrderableDBInstanceOptions con un AWS SDK o una CLI
- Utilizzo GenerateRDSAuthToken con un AWS SDK o una CLI
- Utilizzo ModifyDBInstance con un AWS SDK o una CLI
- Utilizzo ModifyDBParameterGroup con un AWS SDK o una CLI
- Utilizzo RebootDBInstance con un AWS SDK o una CLI

Utilizzo **`CreateDBInstance`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`CreateDBInstance`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Nozioni di base sulle istanze DB

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    /// <summary>
    /// Create an RDS DB instance with a particular set of properties. Use the
action DescribeDBInstancesAsync
```

```
    /// to determine when the DB instance is ready to use.
    /// </summary>
    /// <param name="dbName">Name for the DB instance.</param>
    /// <param name="dbInstanceIdentifier">DB instance identifier.</param>
    /// <param name="parameterGroupName">DB parameter group to associate with the
instance.</param>
    /// <param name="dbEngine">The engine for the DB instance.</param>
    /// <param name="dbEngineVersion">Version for the DB instance.</param>
    /// <param name="instanceClass">Class for the DB instance.</param>
    /// <param name="allocatedStorage">The amount of storage in gibibytes (GiB)
to allocate to the DB instance.</param>
    /// <param name="adminName">Admin user name.</param>
    /// <param name="adminPassword">Admin user password.</param>
    /// <returns>DB instance object.</returns>
    public async Task<DBInstance> CreateDBInstance(string dbName, string
dbInstanceIdentifier,
        string parameterGroupName, string dbEngine, string dbEngineVersion,
        string instanceClass, int allocatedStorage, string adminName, string
adminPassword)
    {
        var response = await _amazonRDS.CreateDBInstanceAsync(
            new CreateDBInstanceRequest()
            {
                DBName = dbName,
                DBInstanceIdentifier = dbInstanceIdentifier,
                DBParameterGroupName = parameterGroupName,
                Engine = dbEngine,
                EngineVersion = dbEngineVersion,
                DBInstanceClass = instanceClass,
                AllocatedStorage = allocatedStorage,
                MasterUsername = adminName,
                MasterUserPassword = adminPassword
            });

        return response.DBInstance;
    }
```

- Per informazioni dettagliate sull'API, consulta CreateDBInstance nella Documentazione di riferimento delle API di AWS SDK for .NET .

C++

SDK per C++

> **Note**
>
> C'è dell'altro GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
        Aws::Client::ClientConfiguration clientConfig;
        // Optional: Set to the AWS Region (overrides config file).
        // clientConfig.region = "us-east-1";

    Aws::RDS::RDSClient client(clientConfig);

        Aws::RDS::Model::CreateDBInstanceRequest request;
        request.SetDBName(DB_NAME);
        request.SetDBInstanceIdentifier(DB_INSTANCE_IDENTIFIER);
        request.SetDBParameterGroupName(PARAMETER_GROUP_NAME);
        request.SetEngine(engineVersion.GetEngine());
        request.SetEngineVersion(engineVersion.GetEngineVersion());
        request.SetDBInstanceClass(dbInstanceClass);
        request.SetStorageType(DB_STORAGE_TYPE);
        request.SetAllocatedStorage(DB_ALLOCATED_STORAGE);
        request.SetMasterUsername(administratorName);
        request.SetMasterUserPassword(administratorPassword);

        Aws::RDS::Model::CreateDBInstanceOutcome outcome =
                client.CreateDBInstance(request);

        if (outcome.IsSuccess()) {
            std::cout << "The DB instance creation has started."
                      << std::endl;
        }
        else {
            std::cerr << "Error with RDS::CreateDBInstance. "
                      << outcome.GetError().GetMessage()
                      << std::endl;
            cleanUpResources(PARAMETER_GROUP_NAME, "", client);
            return false;
        }
```

- Per informazioni dettagliate sull'API, consulta CreateDBInstance nella Documentazione di riferimento delle API di AWS SDK for C++ .

CLI

AWS CLI

Per creare un'istanza DB

L'`create-db-instance`esempio seguente utilizza le opzioni richieste per avviare una nuova istanza DB.

```
aws rds create-db-instance \
    --db-instance-identifier test-mysql-instance \
    --db-instance-class db.t3.micro \
    --engine mysql \
    --master-username admin \
    --master-user-password secret99 \
    --allocated-storage 20
```

Output:

```
{
    "DBInstance": {
        "DBInstanceIdentifier": "test-mysql-instance",
        "DBInstanceClass": "db.t3.micro",
        "Engine": "mysql",
        "DBInstanceStatus": "creating",
        "MasterUsername": "admin",
        "AllocatedStorage": 20,
        "PreferredBackupWindow": "12:55-13:25",
        "BackupRetentionPeriod": 1,
        "DBSecurityGroups": [],
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sg-12345abc",
                "Status": "active"
            }
        ],
        "DBParameterGroups": [
```

```
            {
                "DBParameterGroupName": "default.mysql5.7",
                "ParameterApplyStatus": "in-sync"
            }
        ],
        "DBSubnetGroup": {
            "DBSubnetGroupName": "default",
            "DBSubnetGroupDescription": "default",
            "VpcId": "vpc-2ff2ff2f",
            "SubnetGroupStatus": "Complete",
            "Subnets": [
                {
                    "SubnetIdentifier": "subnet-########",
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2c"
                    },
                    "SubnetStatus": "Active"
                },
                {
                    "SubnetIdentifier": "subnet-########",
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2d"
                    },
                    "SubnetStatus": "Active"
                },
                {
                    "SubnetIdentifier": "subnet-########",
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2a"
                    },
                    "SubnetStatus": "Active"
                },
                {
                    "SubnetIdentifier": "subnet-########",
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2b"
                    },
                    "SubnetStatus": "Active"
                }
            ]
        },
        "PreferredMaintenanceWindow": "sun:08:07-sun:08:37",
        "PendingModifiedValues": {
            "MasterUserPassword": "****"
```

```
        },
        "MultiAZ": false,
        "EngineVersion": "5.7.22",
        "AutoMinorVersionUpgrade": true,
        "ReadReplicaDBInstanceIdentifiers": [],
        "LicenseModel": "general-public-license",
        "OptionGroupMemberships": [
            {
                "OptionGroupName": "default:mysql-5-7",
                "Status": "in-sync"
            }
        ],
        "PubliclyAccessible": true,
        "StorageType": "gp2",
        "DbInstancePort": 0,
        "StorageEncrypted": false,
        "DbiResourceId": "db-5555EXAMPLE44444444EXAMPLE",
        "CACertificateIdentifier": "rds-ca-2019",
        "DomainMemberships": [],
        "CopyTagsToSnapshot": false,
        "MonitoringInterval": 0,
        "DBInstanceArn": "arn:aws:rds:us-west-2:123456789012:db:test-mysql-
instance",
        "IAMDatabaseAuthenticationEnabled": false,
        "PerformanceInsightsEnabled": false,
        "DeletionProtection": false,
        "AssociatedRoles": []
    }
}
```

Per ulteriori informazioni, consulta Creazione di un'istanza database Amazon RDS nella Amazon RDS User Guide.

- Per i dettagli sull'API, consulta CreateDBInstance in Command Reference AWS CLI .

Go

SDK per Go V2

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
type DbInstances struct {
 RdsClient *rds.Client
}



// CreateInstance creates a DB instance.
func (instances *DbInstances) CreateInstance(instanceName string, dbName string,
 dbEngine string, dbEngineVersion string, parameterGroupName string,
 dbInstanceClass string,
 storageType string, allocatedStorage int32, adminName string, adminPassword
 string) (
 *types.DBInstance, error) {
 output, err := instances.RdsClient.CreateDBInstance(context.TODO(),
 &rds.CreateDBInstanceInput{
  DBInstanceIdentifier: aws.String(instanceName),
  DBName:               aws.String(dbName),
  DBParameterGroupName: aws.String(parameterGroupName),
  Engine:               aws.String(dbEngine),
  EngineVersion:        aws.String(dbEngineVersion),
  DBInstanceClass:      aws.String(dbInstanceClass),
  StorageType:          aws.String(storageType),
  AllocatedStorage:     aws.Int32(allocatedStorage),
  MasterUsername:       aws.String(adminName),
  MasterUserPassword:   aws.String(adminPassword),
 })
 if err != nil {
  log.Printf("Couldn't create instance %v: %v\n", instanceName, err)
  return nil, err
 } else {
  return output.DBInstance, nil
```

```
 }
}
```

- Per informazioni dettagliate sull'API, consulta CreateDBInstance nella Documentazione di riferimento delle API di AWS SDK for Go .

Java

SDK per Java 2.x

> **Note**
>
> C'è dell'altro GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
import com.google.gson.Gson;
import
 software.amazon.awssdk.auth.credentials.EnvironmentVariableCredentialsProvider;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.rds.RdsClient;
import software.amazon.awssdk.services.rds.model.DescribeDbInstancesRequest;
import software.amazon.awssdk.services.rds.model.CreateDbInstanceRequest;
import software.amazon.awssdk.services.rds.model.CreateDbInstanceResponse;
import software.amazon.awssdk.services.rds.model.RdsException;
import software.amazon.awssdk.services.rds.model.DescribeDbInstancesResponse;
import software.amazon.awssdk.services.rds.model.DBInstance;
import software.amazon.awssdk.services.secretsmanager.SecretsManagerClient;
import
 software.amazon.awssdk.services.secretsmanager.model.GetSecretValueRequest;
import
 software.amazon.awssdk.services.secretsmanager.model.GetSecretValueResponse;

import java.util.List;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
```

```
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 *
 * This example requires an AWS Secrets Manager secret that contains the
 * database credentials. If you do not create a
 * secret, this example will not work. For more details, see:
 *
 * https://docs.aws.amazon.com/secretsmanager/latest/userguide/integrating_how-
services-use-secrets_RS.html
 *
 *
 */

public class CreateDBInstance {
    public static long sleepTime = 20;

    public static void main(String[] args) {
        final String usage = """

                Usage:
                    <dbInstanceIdentifier> <dbName> <secretName>

                Where:
                    dbInstanceIdentifier - The database instance identifier.\s
                    dbName - The database name.\s
                    secretName - The name of the AWS Secrets Manager secret that
 contains the database credentials."
                """;

        if (args.length != 3) {
            System.out.println(usage);
            System.exit(1);
        }

        String dbInstanceIdentifier = args[0];
        String dbName = args[1];
        String secretName = args[2];
        Gson gson = new Gson();
        User user = gson.fromJson(String.valueOf(getSecretValues(secretName)),
 User.class);
        Region region = Region.US_WEST_2;
        RdsClient rdsClient = RdsClient.builder()
```

```
                .region(region)
                .build();

        createDatabaseInstance(rdsClient, dbInstanceIdentifier, dbName,
user.getUsername(), user.getPassword());
        waitForInstanceReady(rdsClient, dbInstanceIdentifier);
        rdsClient.close();
    }

    private static SecretsManagerClient getSecretClient() {
        Region region = Region.US_WEST_2;
        return SecretsManagerClient.builder()
                .region(region)

.credentialsProvider(EnvironmentVariableCredentialsProvider.create())
                .build();
    }

    private static String getSecretValues(String secretName) {
        SecretsManagerClient secretClient = getSecretClient();
        GetSecretValueRequest valueRequest = GetSecretValueRequest.builder()
                .secretId(secretName)
                .build();

        GetSecretValueResponse valueResponse =
secretClient.getSecretValue(valueRequest);
        return valueResponse.secretString();
    }

    public static void createDatabaseInstance(RdsClient rdsClient,
            String dbInstanceIdentifier,
            String dbName,
            String userName,
            String userPassword) {

        try {
            CreateDbInstanceRequest instanceRequest =
CreateDbInstanceRequest.builder()
                    .dbInstanceIdentifier(dbInstanceIdentifier)
                    .allocatedStorage(100)
                    .dbName(dbName)
                    .engine("mysql")
                    .dbInstanceClass("db.m4.large")
                    .engineVersion("8.0")
```

```
                    .storageType("standard")
                    .masterUsername(userName)
                    .masterUserPassword(userPassword)
                    .build();

            CreateDbInstanceResponse response =
rdsClient.createDBInstance(instanceRequest);
            System.out.print("The status is " +
response.dbInstance().dbInstanceStatus());

        } catch (RdsException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }
    }

    // Waits until the database instance is available.
    public static void waitForInstanceReady(RdsClient rdsClient, String
dbInstanceIdentifier) {
        boolean instanceReady = false;
        String instanceReadyStr;
        System.out.println("Waiting for instance to become available.");
        try {
            DescribeDbInstancesRequest instanceRequest =
DescribeDbInstancesRequest.builder()
                    .dbInstanceIdentifier(dbInstanceIdentifier)
                    .build();

            // Loop until the cluster is ready.
            while (!instanceReady) {
                DescribeDbInstancesResponse response =
rdsClient.describeDBInstances(instanceRequest);
                List<DBInstance> instanceList = response.dbInstances();
                for (DBInstance instance : instanceList) {
                    instanceReadyStr = instance.dbInstanceStatus();
                    if (instanceReadyStr.contains("available"))
                        instanceReady = true;
                    else {
                        System.out.print(".");
                        Thread.sleep(sleepTime * 1000);
                    }
                }
            }
            System.out.println("Database instance is available!");
```

```
        } catch (RdsException | InterruptedException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }
```

- Per informazioni dettagliate sull'API, consulta CreateDBInstance nella Documentazione di riferimento delle API di AWS SDK for Java 2.x .

Kotlin

SDK per Kotlin

> **Note**
>
> C'è dell'altro GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
suspend fun createDatabaseInstance(
    dbInstanceIdentifierVal: String?,
    dbNamedbVal: String?,
    masterUsernameVal: String?,
    masterUserPasswordVal: String?
) {
    val instanceRequest = CreateDbInstanceRequest {
        dbInstanceIdentifier = dbInstanceIdentifierVal
        allocatedStorage = 100
        dbName = dbNamedbVal
        engine = "mysql"
        dbInstanceClass = "db.m4.large"
        engineVersion = "8.0"
        storageType = "standard"
        masterUsername = masterUsernameVal
        masterUserPassword = masterUserPasswordVal
    }

    RdsClient { region = "us-west-2" }.use { rdsClient ->
        val response = rdsClient.createDbInstance(instanceRequest)
```

```
        print("The status is ${response.dbInstance?.dbInstanceStatus}")
    }
}

// Waits until the database instance is available.
suspend fun waitForInstanceReady(dbInstanceIdentifierVal: String?) {
    val sleepTime: Long = 20
    var instanceReady = false
    var instanceReadyStr = ""
    println("Waiting for instance to become available.")

    val instanceRequest = DescribeDbInstancesRequest {
        dbInstanceIdentifier = dbInstanceIdentifierVal
    }

    RdsClient { region = "us-west-2" }.use { rdsClient ->
        while (!instanceReady) {
            val response = rdsClient.describeDbInstances(instanceRequest)
            val instanceList = response.dbInstances
            if (instanceList != null) {
                for (instance in instanceList) {
                    instanceReadyStr = instance.dbInstanceStatus.toString()
                    if (instanceReadyStr.contains("available")) {
                        instanceReady = true
                    } else {
                        println("...$instanceReadyStr")
                        delay(sleepTime * 1000)
                    }
                }
            }
        }
        println("Database instance is available!")
    }
}
```

- Per informazioni dettagliate sull'API, consulta CreateDBInstance nella Documentazione di riferimento delle API di AWS SDK per Kotlin.

PHP

SDK per PHP

> **Note**
>
> C'è dell'altro GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
require __DIR__ . '/vendor/autoload.php';

use Aws\Exception\AwsException;



$rdsClient = new Aws\Rds\RdsClient([
    'region' => 'us-east-2'
]);

$dbIdentifier = '<<{{db-identifier}}>>';
$dbClass = 'db.t2.micro';
$storage = 5;
$engine = 'MySQL';
$username = 'MyUser';
$password = 'MyPassword';

try {
    $result = $rdsClient->createDBInstance([
        'DBInstanceIdentifier' => $dbIdentifier,
        'DBInstanceClass' => $dbClass,
        'AllocatedStorage' => $storage,
        'Engine' => $engine,
        'MasterUsername' => $username,
        'MasterUserPassword' => $password,
    ]);
    var_dump($result);
} catch (AwsException $e) {
    echo $e->getMessage();
    echo "\n";
}
```

- Per informazioni dettagliate sull'API, consulta CreateDBInstance nella Documentazione di riferimento delle API di AWS SDK for PHP .

Python

SDK per Python (Boto3)

> **Note**
>
> C'è dell'altro GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
class InstanceWrapper:
    """Encapsulates Amazon RDS DB instance actions."""

    def __init__(self, rds_client):
        """
        :param rds_client: A Boto3 Amazon RDS client.
        """
        self.rds_client = rds_client

    @classmethod
    def from_client(cls):
        """
        Instantiates this class from a Boto3 client.
        """
        rds_client = boto3.client("rds")
        return cls(rds_client)


    def create_db_instance(
        self,
        db_name,
        instance_id,
        parameter_group_name,
        db_engine,
        db_engine_version,
```

```
        instance_class,
        storage_type,
        allocated_storage,
        admin_name,
        admin_password,
    ):
        """
        Creates a DB instance.

        :param db_name: The name of the database that is created in the DB
instance.
        :param instance_id: The ID to give the newly created DB instance.
        :param parameter_group_name: A parameter group to associate with the DB
instance.
        :param db_engine: The database engine of a database to create in the DB
instance.
        :param db_engine_version: The engine version for the created database.
        :param instance_class: The DB instance class for the newly created DB
instance.
        :param storage_type: The storage type of the DB instance.
        :param allocated_storage: The amount of storage allocated on the DB
instance, in GiBs.
        :param admin_name: The name of the admin user for the created database.
        :param admin_password: The admin password for the created database.
        :return: Data about the newly created DB instance.
        """
        try:
            response = self.rds_client.create_db_instance(
                DBName=db_name,
                DBInstanceIdentifier=instance_id,
                DBParameterGroupName=parameter_group_name,
                Engine=db_engine,
                EngineVersion=db_engine_version,
                DBInstanceClass=instance_class,
                StorageType=storage_type,
                AllocatedStorage=allocated_storage,
                MasterUsername=admin_name,
                MasterUserPassword=admin_password,
            )
            db_inst = response["DBInstance"]
        except ClientError as err:
            logger.error(
                "Couldn't create DB instance %s. Here's why: %s: %s",
                instance_id,
```

```
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return db_inst
```

- Per informazioni dettagliate sull'API, consulta CreateDBInstance nella Documentazione di riferimento delle API di AWS SDK per Python (Boto3).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`CreateDBParameterGroup`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`CreateDBParameterGroup`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Nozioni di base sulle istanze DB

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    /// <summary>
    /// Create a new DB parameter group. Use the action
DescribeDBParameterGroupsAsync
```

```
    /// to determine when the DB parameter group is ready to use.
    /// </summary>
    /// <param name="name">Name of the DB parameter group.</param>
    /// <param name="family">Family of the DB parameter group.</param>
    /// <param name="description">Description of the DB parameter group.</param>
    /// <returns>The new DB parameter group.</returns>
    public async Task<DBParameterGroup> CreateDBParameterGroup(
        string name, string family, string description)
    {
        var response = await _amazonRDS.CreateDBParameterGroupAsync(
            new CreateDBParameterGroupRequest()
            {
                DBParameterGroupName = name,
                DBParameterGroupFamily = family,
                Description = description
            });
        return response.DBParameterGroup;
    }
```

- Per i dettagli sull'API, consulta CreateDB ParameterGroup in AWS SDK for .NET API Reference.

C++

SDK per C++

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    Aws::Client::ClientConfiguration clientConfig;
    // Optional: Set to the AWS Region (overrides config file).
    // clientConfig.region = "us-east-1";

Aws::RDS::RDSClient client(clientConfig);

    Aws::RDS::Model::CreateDBParameterGroupRequest request;
```

```
        request.SetDBParameterGroupName(PARAMETER_GROUP_NAME);
        request.SetDBParameterGroupFamily(dbParameterGroupFamily);
        request.SetDescription("Example parameter group.");

        Aws::RDS::Model::CreateDBParameterGroupOutcome outcome =
                client.CreateDBParameterGroup(request);

        if (outcome.IsSuccess()) {
            std::cout << "The DB parameter group was successfully created."
                      << std::endl;
        }
        else {
            std::cerr << "Error with RDS::CreateDBParameterGroup. "
                      << outcome.GetError().GetMessage()
                      << std::endl;
            return false;
        }
```

- Per i dettagli sull'API, consulta CreateDB ParameterGroup in AWS SDK for C++ API Reference.

CLI

AWS CLI

Per creare un gruppo di parametri DB

L'`create-db-parameter-group`esempio seguente crea un gruppo di parametri DB.

```
aws rds create-db-parameter-group \
    --db-parameter-group-name mydbparametergroup \
    --db-parameter-group-family MySQL5.6 \
    --description "My new parameter group"
```

Output:

```
{
    "DBParameterGroup": {
        "DBParameterGroupName": "mydbparametergroup",
        "DBParameterGroupFamily": "mysql5.6",
        "Description": "My new parameter group",
```

```
        "DBParameterGroupArn": "arn:aws:rds:us-
east-1:123456789012:pg:mydbparametergroup"
    }
}
```

Per ulteriori informazioni, consulta Creating a DB Parameter Group nella Amazon RDS User Guide.

- Per i dettagli sull'API, consulta CreateDB ParameterGroup in AWS CLI Command Reference.

Go

SDK per Go V2

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
type DbInstances struct {
 RdsClient *rds.Client
}



// CreateParameterGroup creates a DB parameter group that is based on the
 specified
// parameter group family.
func (instances *DbInstances) CreateParameterGroup(
 parameterGroupName string, parameterGroupFamily string, description string) (
 *types.DBParameterGroup, error) {

 output, err := instances.RdsClient.CreateDBParameterGroup(context.TODO(),
  &rds.CreateDBParameterGroupInput{
   DBParameterGroupName:   aws.String(parameterGroupName),
   DBParameterGroupFamily: aws.String(parameterGroupFamily),
   Description:            aws.String(description),
  })
 if err != nil {
```

```
  log.Printf("Couldn't create parameter group %v: %v\n", parameterGroupName, err)
  return nil, err
 } else {
  return output.DBParameterGroup, err
 }
}
```

- Per i dettagli sull'API, consulta CreateDB ParameterGroup in AWS SDK for Go API Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
   public static void createDBParameterGroup(RdsClient rdsClient, String
dbGroupName, String dbParameterGroupFamily) {
       try {
           CreateDbParameterGroupRequest groupRequest =
CreateDbParameterGroupRequest.builder()
                   .dbParameterGroupName(dbGroupName)
                   .dbParameterGroupFamily(dbParameterGroupFamily)
                   .description("Created by using the AWS SDK for Java")
                   .build();

           CreateDbParameterGroupResponse response =
rdsClient.createDBParameterGroup(groupRequest);
           System.out.println("The group name is " +
response.dbParameterGroup().dbParameterGroupName());

       } catch (RdsException e) {
           System.out.println(e.getLocalizedMessage());
           System.exit(1);
       }
```

```
    }
```

- Per i dettagli sull'API, consulta CreateDB ParameterGroup in AWS SDK for Java 2.x API Reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
class InstanceWrapper:
    """Encapsulates Amazon RDS DB instance actions."""

    def __init__(self, rds_client):
        """
        :param rds_client: A Boto3 Amazon RDS client.
        """
        self.rds_client = rds_client

    @classmethod
    def from_client(cls):
        """
        Instantiates this class from a Boto3 client.
        """
        rds_client = boto3.client("rds")
        return cls(rds_client)


    def create_parameter_group(
        self, parameter_group_name, parameter_group_family, description
    ):
        """
        Creates a DB parameter group that is based on the specified parameter
 group
        family.
```

```
        :param parameter_group_name: The name of the newly created parameter
group.
        :param parameter_group_family: The family that is used as the basis of
the new
                                       parameter group.
        :param description: A description given to the parameter group.
        :return: Data about the newly created parameter group.
        """
        try:
            response = self.rds_client.create_db_parameter_group(
                DBParameterGroupName=parameter_group_name,
                DBParameterGroupFamily=parameter_group_family,
                Description=description,
            )
        except ClientError as err:
            logger.error(
                "Couldn't create parameter group %s. Here's why: %s: %s",
                parameter_group_name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return response
```

- Per i dettagli sull'API, consulta CreateDB ParameterGroup in AWS SDK for Python (Boto3) API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`CreateDBSnapshot`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`CreateDBSnapshot`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Nozioni di base sulle istanze DB

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    /// <summary>
    /// Create a snapshot of a DB instance.
    /// </summary>
    /// <param name="dbInstanceIdentifier">DB instance identifier.</param>
    /// <param name="snapshotIdentifier">Identifier for the snapshot.</param>
    /// <returns>DB snapshot object.</returns>
    public async Task<DBSnapshot> CreateDBSnapshot(string dbInstanceIdentifier,
string snapshotIdentifier)
    {
        var response = await _amazonRDS.CreateDBSnapshotAsync(
            new CreateDBSnapshotRequest()
            {
                DBSnapshotIdentifier = snapshotIdentifier,
                DBInstanceIdentifier = dbInstanceIdentifier
            });

        return response.DBSnapshot;
    }
```

- Per informazioni dettagliate sull'API, consulta CreateDBSnapshot nella Documentazione di riferimento dell'API AWS SDK for .NET .

C++

SDK per C++

> **Note**
>
> C'è dell'altro GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
        Aws::Client::ClientConfiguration clientConfig;
        // Optional: Set to the AWS Region (overrides config file).
        // clientConfig.region = "us-east-1";

    Aws::RDS::RDSClient client(clientConfig);

            Aws::RDS::Model::CreateDBSnapshotRequest request;
            request.SetDBInstanceIdentifier(DB_INSTANCE_IDENTIFIER);
            request.SetDBSnapshotIdentifier(snapshotID);

            Aws::RDS::Model::CreateDBSnapshotOutcome outcome =
                    client.CreateDBSnapshot(request);

            if (outcome.IsSuccess()) {
                std::cout << "Snapshot creation has started."
                          << std::endl;
            }
            else {
                std::cerr << "Error with RDS::CreateDBSnapshot. "
                          << outcome.GetError().GetMessage()
                          << std::endl;
                cleanUpResources(PARAMETER_GROUP_NAME, DB_INSTANCE_IDENTIFIER,
client);
                return false;
            }
```

- Per informazioni dettagliate sull'API, consulta CreateDBSnapshot nella Documentazione di riferimento dell'API AWS SDK for C++ .

CLI

AWS CLI

Per creare un'istantanea del DB

L'`create-db-snapshot`esempio seguente crea un'istantanea del DB.

```
aws rds create-db-snapshot \
    --db-instance-identifier database-mysql \
    --db-snapshot-identifier mydbsnapshot
```

Output:

```
{
    "DBSnapshot": {
        "DBSnapshotIdentifier": "mydbsnapshot",
        "DBInstanceIdentifier": "database-mysql",
        "Engine": "mysql",
        "AllocatedStorage": 100,
        "Status": "creating",
        "Port": 3306,
        "AvailabilityZone": "us-east-1b",
        "VpcId": "vpc-6594f31c",
        "InstanceCreateTime": "2019-04-30T15:45:53.663Z",
        "MasterUsername": "admin",
        "EngineVersion": "5.6.40",
        "LicenseModel": "general-public-license",
        "SnapshotType": "manual",
        "Iops": 1000,
        "OptionGroupName": "default:mysql-5-6",
        "PercentProgress": 0,
        "StorageType": "io1",
        "Encrypted": true,
        "KmsKeyId": "arn:aws:kms:us-east-1:123456789012:key/
AKIAIOSFODNN7EXAMPLE",
        "DBSnapshotArn": "arn:aws:rds:us-
east-1:123456789012:snapshot:mydbsnapshot",
        "IAMDatabaseAuthenticationEnabled": false,
        "ProcessorFeatures": [],
        "DbiResourceId": "db-AKIAIOSFODNN7EXAMPLE"
    }
}
```

Per ulteriori informazioni, consulta Creating a DB Snapshot nella Amazon RDS User Guide.

- Per i dettagli sull'API, consulta CreateDBSnapshot in Command Reference AWS CLI .

Go

SDK per Go V2

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
type DbInstances struct {
 RdsClient *rds.Client
}



// CreateSnapshot creates a snapshot of a DB instance.
func (instances *DbInstances) CreateSnapshot(instanceName string, snapshotName
 string) (
 *types.DBSnapshot, error) {
 output, err := instances.RdsClient.CreateDBSnapshot(context.TODO(),
 &rds.CreateDBSnapshotInput{
  DBInstanceIdentifier: aws.String(instanceName),
  DBSnapshotIdentifier: aws.String(snapshotName),
 })
 if err != nil {
  log.Printf("Couldn't create snapshot %v: %v\n", snapshotName, err)
  return nil, err
 } else {
  return output.DBSnapshot, nil
 }
}
```

- Per informazioni dettagliate sull'API, consulta CreateDBSnapshot nella Documentazione di riferimento dell'API AWS SDK for Go .

Java

SDK per Java 2.x

> **Note**
>
> C'è dell'altro GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
   // Create an Amazon RDS snapshot.
   public static void createSnapshot(RdsClient rdsClient, String
dbInstanceIdentifier, String dbSnapshotIdentifier) {
       try {
           CreateDbSnapshotRequest snapshotRequest =
CreateDbSnapshotRequest.builder()
                   .dbInstanceIdentifier(dbInstanceIdentifier)
                   .dbSnapshotIdentifier(dbSnapshotIdentifier)
                   .build();

           CreateDbSnapshotResponse response =
rdsClient.createDBSnapshot(snapshotRequest);
           System.out.println("The Snapshot id is " +
response.dbSnapshot().dbiResourceId());

       } catch (RdsException e) {
           System.out.println(e.getLocalizedMessage());
           System.exit(1);
       }
   }
```

- Per informazioni dettagliate sull'API, consulta CreateDBSnapshot nella Documentazione di riferimento dell'API AWS SDK for Java 2.x .

PHP

SDK per PHP

> **Note**
>
> C'è dell'altro GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
require __DIR__ . '/vendor/autoload.php';

use Aws\Exception\AwsException;



$rdsClient = new Aws\Rds\RdsClient([
    'region' => 'us-east-2'
]);

$dbIdentifier = '<<{{db-identifier}}>>';
$snapshotName = '<<{{backup_2018_12_25}}>>';

try {
    $result = $rdsClient->createDBSnapshot([
        'DBInstanceIdentifier' => $dbIdentifier,
        'DBSnapshotIdentifier' => $snapshotName,
    ]);
    var_dump($result);
} catch (AwsException $e) {
    echo $e->getMessage();
    echo "\n";
}
```

- Per informazioni dettagliate sull'API, consulta CreateDBSnapshot nella Documentazione di riferimento dell'API AWS SDK for PHP .

Python

SDK per Python (Boto3)

> **Note**
>
> C'è dell'altro GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
class InstanceWrapper:
    """Encapsulates Amazon RDS DB instance actions."""

    def __init__(self, rds_client):
        """
        :param rds_client: A Boto3 Amazon RDS client.
        """
        self.rds_client = rds_client

    @classmethod
    def from_client(cls):
        """
        Instantiates this class from a Boto3 client.
        """
        rds_client = boto3.client("rds")
        return cls(rds_client)


    def create_snapshot(self, snapshot_id, instance_id):
        """
        Creates a snapshot of a DB instance.

        :param snapshot_id: The ID to give the created snapshot.
        :param instance_id: The ID of the DB instance to snapshot.
        :return: Data about the newly created snapshot.
        """
        try:
            response = self.rds_client.create_db_snapshot(
                DBSnapshotIdentifier=snapshot_id,
 DBInstanceIdentifier=instance_id
            )
            snapshot = response["DBSnapshot"]
```

```
        except ClientError as err:
            logger.error(
                "Couldn't create snapshot of %s. Here's why: %s: %s",
                instance_id,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return snapshot
```

- Per informazioni dettagliate sull'API, consulta CreateDBSnapshot nella Documentazione di riferimento dell'API SDK AWS per Python (Boto3).

Ruby

SDK per Ruby

> **Note**
>
> C'è dell'altro GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
require "aws-sdk-rds"  # v2: require 'aws-sdk'

# Create a snapshot for an Amazon Relational Database Service (Amazon RDS)
# DB instance.
#
# @param rds_resource [Aws::RDS::Resource] The resource containing SDK logic.
# @param db_instance_name [String] The name of the Amazon RDS DB instance.
# @return [Aws::RDS::DBSnapshot, nil] The snapshot created, or nil if error.
def create_snapshot(rds_resource, db_instance_name)
  id = "snapshot-#{rand(10**6)}"
  db_instance = rds_resource.db_instance(db_instance_name)
  db_instance.create_snapshot({
                                db_snapshot_identifier: id
                              })
rescue Aws::Errors::ServiceError => e
```

```
  puts "Couldn't create DB instance snapshot #{id}:\n #{e.message}"
end
```

- Per informazioni dettagliate sull'API, consulta CreateDBSnapshot nella Documentazione di riferimento dell'API AWS SDK for Ruby .

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`DeleteDBInstance`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`DeleteDBInstance`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Nozioni di base sulle istanze DB

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
/// <summary>
/// Delete a particular DB instance.
/// </summary>
/// <param name="dbInstanceIdentifier">DB instance identifier.</param>
/// <returns>DB instance object.</returns>
public async Task<DBInstance> DeleteDBInstance(string dbInstanceIdentifier)
{
    var response = await _amazonRDS.DeleteDBInstanceAsync(
        new DeleteDBInstanceRequest()
```

```
            {
                DBInstanceIdentifier = dbInstanceIdentifier,
                SkipFinalSnapshot = true,
                DeleteAutomatedBackups = true
            });

        return response.DBInstance;
    }
```

- Per informazioni dettagliate sull'API, consulta DeleteDBInstance nella Documentazione di riferimento delle API di AWS SDK for .NET .

C++

SDK per C++

> **Note**
>
> C'è dell'altro GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
        Aws::Client::ClientConfiguration clientConfig;
        // Optional: Set to the AWS Region (overrides config file).
        // clientConfig.region = "us-east-1";

    Aws::RDS::RDSClient client(clientConfig);

            Aws::RDS::Model::DeleteDBInstanceRequest request;
            request.SetDBInstanceIdentifier(dbInstanceIdentifier);
            request.SetSkipFinalSnapshot(true);
            request.SetDeleteAutomatedBackups(true);

            Aws::RDS::Model::DeleteDBInstanceOutcome outcome =
                    client.DeleteDBInstance(request);

            if (outcome.IsSuccess()) {
                std::cout << "DB instance deletion has started."
                          << std::endl;
```

```
            }
            else {
                std::cerr << "Error with RDS::DeleteDBInstance. "
                          << outcome.GetError().GetMessage()
                          << std::endl;
                result = false;
            }
```

- Per informazioni dettagliate sull'API, consulta DeleteDBInstance nella Documentazione di riferimento delle API di AWS SDK for C++ .

CLI

AWS CLI

Per eliminare un'istanza DB

L'`delete-db-instance`esempio seguente elimina l'istanza DB specificata dopo aver creato uno snapshot DB finale denominato. `test-instance-final-snap`

```
aws rds delete-db-instance \
    --db-instance-identifier test-instance \
    --final-db-snapshot-identifier test-instance-final-snap
```

Output:

```
{
    "DBInstance": {
        "DBInstanceIdentifier": "test-instance",
        "DBInstanceStatus": "deleting",
        ...some output truncated...
    }
}
```

- Per i dettagli sull'API, consulta AWS CLI DeletedBInstance in Command Reference.

Go

SDK per Go V2

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
type DbInstances struct {
 RdsClient *rds.Client
}



// DeleteInstance deletes a DB instance.
func (instances *DbInstances) DeleteInstance(instanceName string) error {
 _, err := instances.RdsClient.DeleteDBInstance(context.TODO(),
 &rds.DeleteDBInstanceInput{
  DBInstanceIdentifier:   aws.String(instanceName),
  SkipFinalSnapshot:      true,
  DeleteAutomatedBackups: aws.Bool(true),
 })
 if err != nil {
  log.Printf("Couldn't delete instance %v: %v\n", instanceName, err)
  return err
 } else {
  return nil
 }
}
```

- Per informazioni dettagliate sull'API, consulta DeleteDBInstance nella Documentazione di riferimento delle API di AWS SDK for Go .

Java

SDK per Java 2.x

> **Note**
>
> C'è dell'altro GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.rds.RdsClient;
import software.amazon.awssdk.services.rds.model.DeleteDbInstanceRequest;
import software.amazon.awssdk.services.rds.model.DeleteDbInstanceResponse;
import software.amazon.awssdk.services.rds.model.RdsException;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class DeleteDBInstance {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                    <dbInstanceIdentifier>\s

                Where:
                    dbInstanceIdentifier - The database instance identifier\s
                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String dbInstanceIdentifier = args[0];
```

```
        Region region = Region.US_WEST_2;
        RdsClient rdsClient = RdsClient.builder()
                .region(region)
                .build();

        deleteDatabaseInstance(rdsClient, dbInstanceIdentifier);
        rdsClient.close();
    }

    public static void deleteDatabaseInstance(RdsClient rdsClient, String
 dbInstanceIdentifier) {
        try {
            DeleteDbInstanceRequest deleteDbInstanceRequest =
 DeleteDbInstanceRequest.builder()
                    .dbInstanceIdentifier(dbInstanceIdentifier)
                    .deleteAutomatedBackups(true)
                    .skipFinalSnapshot(true)
                    .build();

            DeleteDbInstanceResponse response =
 rdsClient.deleteDBInstance(deleteDbInstanceRequest);
            System.out.print("The status of the database is " +
 response.dbInstance().dbInstanceStatus());

        } catch (RdsException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }
    }
}
```

- Per informazioni dettagliate sull'API, consulta DeleteDBInstance nella Documentazione di riferimento delle API di AWS SDK for Java 2.x .

Kotlin

SDK per Kotlin

> **Note**
>
> C'è dell'altro GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
suspend fun deleteDatabaseInstance(dbInstanceIdentifierVal: String?) {

    val deleteDbInstanceRequest = DeleteDbInstanceRequest {
        dbInstanceIdentifier = dbInstanceIdentifierVal
        deleteAutomatedBackups = true
        skipFinalSnapshot = true
    }

    RdsClient { region = "us-west-2" }.use { rdsClient ->
        val response = rdsClient.deleteDbInstance(deleteDbInstanceRequest)
        print("The status of the database is
 ${response.dbInstance?.dbInstanceStatus}")
    }
}
```

- Per informazioni dettagliate sull'API, consulta DeleteDBInstance nella Documentazione di riferimento delle API di AWS SDK per Kotlin.

PHP

SDK per PHP

> **Note**
>
> C'è dell'altro GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
require __DIR__ . '/vendor/autoload.php';

use Aws\Exception\AwsException;


//Create an RDSClient
$rdsClient = new Aws\Rds\RdsClient([
    'region' => 'us-east-1'
]);

$dbIdentifier = '<<{{db-identifier}}>>';

try {
    $result = $rdsClient->deleteDBInstance([
        'DBInstanceIdentifier' => $dbIdentifier,
    ]);
    var_dump($result);
} catch (AwsException $e) {
    echo $e->getMessage();
    echo "\n";
}
```

- Per informazioni dettagliate sull'API, consulta DeleteDBInstance nella Documentazione di riferimento delle API di AWS SDK for PHP .

Python

SDK per Python (Boto3)

> **Note**
>
> C'è dell'altro GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
class InstanceWrapper:
    """Encapsulates Amazon RDS DB instance actions."""

    def __init__(self, rds_client):
```

```
        """
        :param rds_client: A Boto3 Amazon RDS client.
        """
        self.rds_client = rds_client

    @classmethod
    def from_client(cls):
        """
        Instantiates this class from a Boto3 client.
        """
        rds_client = boto3.client("rds")
        return cls(rds_client)


    def delete_db_instance(self, instance_id):
        """
        Deletes a DB instance.

        :param instance_id: The ID of the DB instance to delete.
        :return: Data about the deleted DB instance.
        """
        try:
            response = self.rds_client.delete_db_instance(
                DBInstanceIdentifier=instance_id,
                SkipFinalSnapshot=True,
                DeleteAutomatedBackups=True,
            )
            db_inst = response["DBInstance"]
        except ClientError as err:
            logger.error(
                "Couldn't delete DB instance %s. Here's why: %s: %s",
                instance_id,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return db_inst
```

- Per informazioni dettagliate sull'API, consulta DeleteDBInstance nella documentazione di riferimento dell'API di AWS SDK per Python (Boto3).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`DeleteDBParameterGroup`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`DeleteDBParameterGroup`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Nozioni di base sulle istanze DB

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
   /// <summary>
   /// Delete a DB parameter group. The group cannot be a default DB parameter
group
   /// or be associated with any DB instances.
   /// </summary>
   /// <param name="name">Name of the DB parameter group.</param>
   /// <returns>True if successful.</returns>
   public async Task<bool> DeleteDBParameterGroup(string name)
   {
       var response = await _amazonRDS.DeleteDBParameterGroupAsync(
           new DeleteDBParameterGroupRequest()
           {
               DBParameterGroupName = name,
           });
       return response.HttpStatusCode == HttpStatusCode.OK;
   }
```

- Per i dettagli sull'API, consulta DeleteDB ParameterGroup in AWS SDK for .NET API Reference.

C++

SDK per C++

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
        Aws::Client::ClientConfiguration clientConfig;
        // Optional: Set to the AWS Region (overrides config file).
        // clientConfig.region = "us-east-1";

    Aws::RDS::RDSClient client(clientConfig);

        Aws::RDS::Model::DeleteDBParameterGroupRequest request;
        request.SetDBParameterGroupName(parameterGroupName);

        Aws::RDS::Model::DeleteDBParameterGroupOutcome outcome =
                client.DeleteDBParameterGroup(request);

        if (outcome.IsSuccess()) {
            std::cout << "The DB parameter group was successfully deleted."
                      << std::endl;
        }
        else {
            std::cerr << "Error with RDS::DeleteDBParameterGroup. "
                      << outcome.GetError().GetMessage()
                      << std::endl;
            result = false;
        }
```

- Per i dettagli sull'API, consulta DeleteDB ParameterGroup in AWS SDK for C++ API Reference.

CLI

AWS CLI

Per eliminare un gruppo di parametri DB

L'commandesempio seguente elimina un gruppo di parametri DB.

```
aws rds delete-db-parameter-group \
    --db-parameter-group-name mydbparametergroup
```

Questo comando non produce alcun output.

Per ulteriori informazioni, consulta Utilizzo di gruppi di parametri di database nella Guida per l'utente di Amazon RDS.

- Per i dettagli sull'API, consulta DeleteDB ParameterGroup in AWS CLI Command Reference.

Go

SDK per Go V2

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
type DbInstances struct {
 RdsClient *rds.Client
}



// DeleteParameterGroup deletes the named DB parameter group.
func (instances *DbInstances) DeleteParameterGroup(parameterGroupName string)
 error {
 _, err := instances.RdsClient.DeleteDBParameterGroup(context.TODO(),
  &rds.DeleteDBParameterGroupInput{
   DBParameterGroupName: aws.String(parameterGroupName),
```

```
  })
 if err != nil {
  log.Printf("Couldn't delete parameter group %v: %v\n", parameterGroupName, err)
  return err
 } else {
  return nil
 }
}
```

- Per i dettagli sull'API, consulta DeleteDB ParameterGroup in AWS SDK for Go API Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
   // Delete the parameter group after database has been deleted.
   // An exception is thrown if you attempt to delete the para group while
database
   // exists.
   public static void deleteParaGroup(RdsClient rdsClient, String dbGroupName,
String dbARN)
           throws InterruptedException {
       try {
           boolean isDataDel = false;
           boolean didFind;
           String instanceARN;

           // Make sure that the database has been deleted.
           while (!isDataDel) {
               DescribeDbInstancesResponse response =
rdsClient.describeDBInstances();
               List<DBInstance> instanceList = response.dbInstances();
```

```
                int listSize = instanceList.size();
                didFind = false;
                int index = 1;
                for (DBInstance instance : instanceList) {
                    instanceARN = instance.dbInstanceArn();
                    if (instanceARN.compareTo(dbARN) == 0) {
                        System.out.println(dbARN + " still exists");
                        didFind = true;
                    }
                    if ((index == listSize) && (!didFind)) {
                        // Went through the entire list and did not find the
database ARN.
                        isDataDel = true;
                    }
                    Thread.sleep(sleepTime * 1000);
                    index++;
                }
            }

            // Delete the para group.
            DeleteDbParameterGroupRequest parameterGroupRequest =
DeleteDbParameterGroupRequest.builder()
                    .dbParameterGroupName(dbGroupName)
                    .build();

            rdsClient.deleteDBParameterGroup(parameterGroupRequest);
            System.out.println(dbGroupName + " was deleted.");

        } catch (RdsException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }
    }
```

- Per i dettagli sull'API, consulta DeleteDB ParameterGroup in AWS SDK for Java 2.x API Reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
class InstanceWrapper:
    """Encapsulates Amazon RDS DB instance actions."""

    def __init__(self, rds_client):
        """
        :param rds_client: A Boto3 Amazon RDS client.
        """
        self.rds_client = rds_client

    @classmethod
    def from_client(cls):
        """
        Instantiates this class from a Boto3 client.
        """
        rds_client = boto3.client("rds")
        return cls(rds_client)


    def delete_parameter_group(self, parameter_group_name):
        """
        Deletes a DB parameter group.

        :param parameter_group_name: The name of the parameter group to delete.
        :return: Data about the parameter group.
        """
        try:
            self.rds_client.delete_db_parameter_group(
                DBParameterGroupName=parameter_group_name
            )
        except ClientError as err:
            logger.error(
                "Couldn't delete parameter group %s. Here's why: %s: %s",
```

```
                parameter_group_name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
```

- Per i dettagli sull'API, consulta DeleteDB ParameterGroup in AWS SDK for Python (Boto3) API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`DescribeAccountAttributes`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`DescribeAccountAttributes`.

CLI

AWS CLI

Per descrivere gli attributi dell'account

L'`describe-account-attributes`esempio seguente recupera gli attributi per l' AWS account corrente.

```
aws rds describe-account-attributes
```

Output:

```
{
    "AccountQuotas": [
        {
            "Max": 40,
            "Used": 4,
            "AccountQuotaName": "DBInstances"
        },
        {
            "Max": 40,
            "Used": 0,
```

```
            "AccountQuotaName": "ReservedDBInstances"
        },
        {
            "Max": 100000,
            "Used": 40,
            "AccountQuotaName": "AllocatedStorage"
        },
        {
            "Max": 25,
            "Used": 0,
            "AccountQuotaName": "DBSecurityGroups"
        },
        {
            "Max": 20,
            "Used": 0,
            "AccountQuotaName": "AuthorizationsPerDBSecurityGroup"
        },
        {
            "Max": 50,
            "Used": 1,
            "AccountQuotaName": "DBParameterGroups"
        },
        {
            "Max": 100,
            "Used": 3,
            "AccountQuotaName": "ManualSnapshots"
        },
        {
            "Max": 20,
            "Used": 0,
            "AccountQuotaName": "EventSubscriptions"
        },
        {
            "Max": 50,
            "Used": 1,
            "AccountQuotaName": "DBSubnetGroups"
        },
        {
            "Max": 20,
            "Used": 1,
            "AccountQuotaName": "OptionGroups"
        },
        {
            "Max": 20,
```

```
            "Used": 6,
            "AccountQuotaName": "SubnetsPerDBSubnetGroup"
        },
        {
            "Max": 5,
            "Used": 0,
            "AccountQuotaName": "ReadReplicasPerMaster"
        },
        {
            "Max": 40,
            "Used": 1,
            "AccountQuotaName": "DBClusters"
        },
        {
            "Max": 50,
            "Used": 0,
            "AccountQuotaName": "DBClusterParameterGroups"
        },
        {
            "Max": 5,
            "Used": 0,
            "AccountQuotaName": "DBClusterRoles"
        }
    ]
}
```

- Per i dettagli sull'API, vedere DescribeAccountAttributesin AWS CLI Command Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.rds.RdsClient;
import software.amazon.awssdk.services.rds.model.AccountQuota;
import software.amazon.awssdk.services.rds.model.RdsException;
```

```
import
 software.amazon.awssdk.services.rds.model.DescribeAccountAttributesResponse;
import java.util.List;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class DescribeAccountAttributes {
    public static void main(String[] args) {
        Region region = Region.US_WEST_2;
        RdsClient rdsClient = RdsClient.builder()
                .region(region)
                .build();

        getAccountAttributes(rdsClient);
        rdsClient.close();
    }

    public static void getAccountAttributes(RdsClient rdsClient) {
        try {
            DescribeAccountAttributesResponse response =
 rdsClient.describeAccountAttributes();
            List<AccountQuota> quotasList = response.accountQuotas();
            for (AccountQuota quotas : quotasList) {
                System.out.println("Name is: " + quotas.accountQuotaName());
                System.out.println("Max value is " + quotas.max());
            }

        } catch (RdsException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }
    }
}
```

- Per i dettagli sull'API, DescribeAccountAttributesconsulta AWS SDK for Java 2.x API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
suspend fun getAccountAttributes() {

    RdsClient { region = "us-west-2" }.use { rdsClient ->
        val response =
 rdsClient.describeAccountAttributes(DescribeAccountAttributesRequest {})
        response.accountQuotas?.forEach { quotas ->
            val response = response.accountQuotas
            println("Name is: ${quotas.accountQuotaName}")
            println("Max value is ${quotas.max}")
        }
    }
}
```

- Per i dettagli sull'API, DescribeAccountAttributesconsulta AWS SDK for Kotlin API reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`DescribeDBEngineVersions`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`DescribeDBEngineVersions`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Nozioni di base sulle istanze DB

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    /// <summary>
    /// Get a list of DB engine versions for a particular DB engine.
    /// </summary>
    /// <param name="engine">Name of the engine.</param>
    /// <param name="dbParameterGroupFamily">Optional parameter group family
name.</param>
    /// <returns>List of DBEngineVersions.</returns>
    public async Task<List<DBEngineVersion>> DescribeDBEngineVersions(string
engine,
        string dbParameterGroupFamily = null)
    {
        var response = await _amazonRDS.DescribeDBEngineVersionsAsync(
            new DescribeDBEngineVersionsRequest()
            {
                Engine = engine,
                DBParameterGroupFamily = dbParameterGroupFamily
            });
        return response.DBEngineVersions;
    }
```

- Per i dettagli sull'API, consulta DescribeDB EngineVersions in AWS SDK for .NET API Reference.

C++

SDK per C++

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
        Aws::Client::ClientConfiguration clientConfig;
        // Optional: Set to the AWS Region (overrides config file).
        // clientConfig.region = "us-east-1";

    Aws::RDS::RDSClient client(clientConfig);


//! Routine which gets available DB engine versions for an engine name and
//! an optional parameter group family.
/*!
 \sa getDBEngineVersions()
 \param engineName: A DB engine name.
 \param parameterGroupFamily: A parameter group family name, ignored if empty.
 \param engineVersionsResult: Vector of 'DBEngineVersion' objects returned by the
 routine.
 \param client: 'RDSClient' instance.
 \return bool: Successful completion.
 */
bool AwsDoc::RDS::getDBEngineVersions(const Aws::String &engineName,
                                      const Aws::String &parameterGroupFamily,

 Aws::Vector<Aws::RDS::Model::DBEngineVersion> &engineVersionsResult,
                                      const Aws::RDS::RDSClient &client) {
    Aws::RDS::Model::DescribeDBEngineVersionsRequest request;
    request.SetEngine(engineName);
    if (!parameterGroupFamily.empty()) {
        request.SetDBParameterGroupFamily(parameterGroupFamily);
    }

    engineVersionsResult.clear();
    Aws::String marker; // Used for pagination.
```

```
    do {
        if (!marker.empty()) {
            request.SetMarker(marker);
        }


        Aws::RDS::Model::DescribeDBEngineVersionsOutcome outcome =
                client.DescribeDBEngineVersions(request);

        if (outcome.IsSuccess()) {
            auto &engineVersions = outcome.GetResult().GetDBEngineVersions();
            engineVersionsResult.insert(engineVersionsResult.end(),
 engineVersions.begin(),
                                        engineVersions.end());
            marker = outcome.GetResult().GetMarker();
        }
        else {
            std::cerr << "Error with RDS::DescribeDBEngineVersionsRequest. "
                      << outcome.GetError().GetMessage()
                      << std::endl;
            return false;
        }

    } while (!marker.empty());


    return true;
}
```

- Per i dettagli sull'API, consulta DescribeDB EngineVersions in AWS SDK for C++ API Reference.

CLI

AWS CLI

Per descrivere le versioni del motore DB per il motore MySQL DB

L'`describe-db-engine-versions`esempio seguente mostra i dettagli su ciascuna delle versioni del motore DB per il motore DB specificato.

```
aws rds describe-db-engine-versions \
```

```
    --engine mysql
```

Output:

```
{
    "DBEngineVersions": [
        {
            "Engine": "mysql",
            "EngineVersion": "5.5.46",
            "DBParameterGroupFamily": "mysql5.5",
            "DBEngineDescription": "MySQL Community Edition",
            "DBEngineVersionDescription": "MySQL 5.5.46",
            "ValidUpgradeTarget": [
                {
                    "Engine": "mysql",
                    "EngineVersion": "5.5.53",
                    "Description": "MySQL 5.5.53",
                    "AutoUpgrade": false,
                    "IsMajorVersionUpgrade": false
                },
                {
                    "Engine": "mysql",
                    "EngineVersion": "5.5.54",
                    "Description": "MySQL 5.5.54",
                    "AutoUpgrade": false,
                    "IsMajorVersionUpgrade": false
                },
                {
                    "Engine": "mysql",
                    "EngineVersion": "5.5.57",
                    "Description": "MySQL 5.5.57",
                    "AutoUpgrade": false,
                    "IsMajorVersionUpgrade": false
                },
                ...some output truncated...
            ]
        }
```

Per ulteriori informazioni, consulta What Is Amazon Relational Database Service (Amazon RDS)? nella Amazon RDS User Guide.

- Per i dettagli sull'API, consulta DescribeDB EngineVersions in Command ReferenceAWS CLI .

Go

SDK per Go V2

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
type DbInstances struct {
 RdsClient *rds.Client
}



// GetEngineVersions gets database engine versions that are available for the
 specified engine
// and parameter group family.
func (instances *DbInstances) GetEngineVersions(engine string,
 parameterGroupFamily string) (
 []types.DBEngineVersion, error) {
 output, err := instances.RdsClient.DescribeDBEngineVersions(context.TODO(),
  &rds.DescribeDBEngineVersionsInput{
   Engine:                 aws.String(engine),
   DBParameterGroupFamily: aws.String(parameterGroupFamily),
  })
 if err != nil {
  log.Printf("Couldn't get engine versions for %v: %v\n", engine, err)
  return nil, err
 } else {
  return output.DBEngineVersions, nil
 }
}
```

- Per i dettagli sull'API, consulta DescribeDB EngineVersions in AWS SDK for Go API Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
   public static void describeDBEngines(RdsClient rdsClient) {
       try {
           DescribeDbEngineVersionsRequest engineVersionsRequest =
DescribeDbEngineVersionsRequest.builder()
                   .defaultOnly(true)
                   .engine("mysql")
                   .maxRecords(20)
                   .build();

           DescribeDbEngineVersionsResponse response =
rdsClient.describeDBEngineVersions(engineVersionsRequest);
           List<DBEngineVersion> engines = response.dbEngineVersions();

           // Get all DBEngineVersion objects.
           for (DBEngineVersion engineOb : engines) {
               System.out.println("The name of the DB parameter group family for
the database engine is "
                       + engineOb.dbParameterGroupFamily());
               System.out.println("The name of the database engine " +
engineOb.engine());
               System.out.println("The version number of the database engine " +
engineOb.engineVersion());
           }

       } catch (RdsException e) {
           System.out.println(e.getLocalizedMessage());
           System.exit(1);
       }
   }
```

- Per i dettagli sull'API, consulta DescribeDB EngineVersions in AWS SDK for Java 2.x API Reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
class InstanceWrapper:
    """Encapsulates Amazon RDS DB instance actions."""

    def __init__(self, rds_client):
        """
        :param rds_client: A Boto3 Amazon RDS client.
        """
        self.rds_client = rds_client

    @classmethod
    def from_client(cls):
        """
        Instantiates this class from a Boto3 client.
        """
        rds_client = boto3.client("rds")
        return cls(rds_client)


    def get_engine_versions(self, engine, parameter_group_family=None):
        """
        Gets database engine versions that are available for the specified engine
        and parameter group family.

        :param engine: The database engine to look up.
        :param parameter_group_family: When specified, restricts the returned
 list of
                                       engine versions to those that are
 compatible with
```

```
                                    this parameter group family.
        :return: The list of database engine versions.
        """
        try:
            kwargs = {"Engine": engine}
            if parameter_group_family is not None:
                kwargs["DBParameterGroupFamily"] = parameter_group_family
            response = self.rds_client.describe_db_engine_versions(**kwargs)
            versions = response["DBEngineVersions"]
        except ClientError as err:
            logger.error(
                "Couldn't get engine versions for %s. Here's why: %s: %s",
                engine,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return versions
```

- Per i dettagli sull'API, consulta DescribeDB EngineVersions in AWS SDK for Python (Boto3) API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`DescribeDBInstances`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`DescribeDBInstances`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Nozioni di base sulle istanze DB

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
   /// <summary>
   /// Returns a list of DB instances.
   /// </summary>
   /// <param name="dbInstanceIdentifier">Optional name of a specific DB
instance.</param>
   /// <returns>List of DB instances.</returns>
   public async Task<List<DBInstance>> DescribeDBInstances(string
dbInstanceIdentifier = null)
   {
       var results = new List<DBInstance>();
       var instancesPaginator = _amazonRDS.Paginators.DescribeDBInstances(
           new DescribeDBInstancesRequest
           {
               DBInstanceIdentifier = dbInstanceIdentifier
           });
       // Get the entire list using the paginator.
       await foreach (var instances in instancesPaginator.DBInstances)
       {
           results.Add(instances);
       }
       return results;
   }
```

- Per informazioni dettagliate sull'API, consulta DescribeDBInstances nella Documentazione di riferimento delle API di AWS SDK for .NET .

C++

SDK per C++

> **Note**
>
> C'è dell'altro GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
        Aws::Client::ClientConfiguration clientConfig;
        // Optional: Set to the AWS Region (overrides config file).
        // clientConfig.region = "us-east-1";

    Aws::RDS::RDSClient client(clientConfig);


//! Routine which gets a DB instance description.
/*!
 \sa describeDBInstance()
 \param dbInstanceIdentifier: A DB instance identifier.
 \param instanceResult: The 'DBInstance' object containing the description.
 \param client: 'RDSClient' instance.
 \return bool: Successful completion.
 */
bool AwsDoc::RDS::describeDBInstance(const Aws::String &dbInstanceIdentifier,
                                     Aws::RDS::Model::DBInstance &instanceResult,
                                     const Aws::RDS::RDSClient &client) {
    Aws::RDS::Model::DescribeDBInstancesRequest request;
    request.SetDBInstanceIdentifier(dbInstanceIdentifier);

    Aws::RDS::Model::DescribeDBInstancesOutcome outcome =
            client.DescribeDBInstances(request);

    bool result = true;
    if (outcome.IsSuccess()) {
        instanceResult = outcome.GetResult().GetDBInstances()[0];
    }
    else if (outcome.GetError().GetErrorType() !=
             Aws::RDS::RDSErrors::D_B_INSTANCE_NOT_FOUND_FAULT) {
        result = false;
        std::cerr << "Error with RDS::DescribeDBInstances. "
```

```
                  << outcome.GetError().GetMessage()
                  << std::endl;
    }
        // This example does not log an error if the DB instance does not exist.
        // Instead, instanceResult is set to empty.
    else {
        instanceResult = Aws::RDS::Model::DBInstance();
    }

    return result;
}
```

- Per informazioni dettagliate sull'API, consulta DescribeDBInstances nella Documentazione di riferimento delle API di AWS SDK for C++ .

CLI

AWS CLI

Per descrivere un'istanza DB

L'`describe-db-instances`esempio seguente recupera i dettagli sull'istanza DB specificata.

```
aws rds describe-db-instances \
    --db-instance-identifier mydbinstancecf
```

Output:

```
{
    "DBInstances": [
        {
            "DBInstanceIdentifier": "mydbinstancecf",
            "DBInstanceClass": "db.t3.small",
            "Engine": "mysql",
            "DBInstanceStatus": "available",
            "MasterUsername": "masterawsuser",
            "Endpoint": {
                "Address": "mydbinstancecf.abcexample.us-
east-1.rds.amazonaws.com",
                "Port": 3306,
                "HostedZoneId": "Z2R2ITUGPM61AM"
```

```
            },
            ...some output truncated...
        }
    ]
}
```

- Per i dettagli sull'API, consulta DescribedBInstances in Command Reference AWS CLI .

Go

SDK per Go V2

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
type DbInstances struct {
 RdsClient *rds.Client
}



// GetInstance gets data about a DB instance.
func (instances *DbInstances) GetInstance(instanceName string) (
 *types.DBInstance, error) {
 output, err := instances.RdsClient.DescribeDBInstances(context.TODO(),
  &rds.DescribeDBInstancesInput{
   DBInstanceIdentifier: aws.String(instanceName),
  })
 if err != nil {
  var notFoundError *types.DBInstanceNotFoundFault
  if errors.As(err, &notFoundError) {
   log.Printf("DB instance %v does not exist.\n", instanceName)
   err = nil
  } else {
   log.Printf("Couldn't get instance %v: %v\n", instanceName, err)
  }
  return nil, err
 } else {
```

```
  return &output.DBInstances[0], nil
 }
}
```

- Per informazioni dettagliate sull'API, consulta DescribeDBInstances nella Documentazione di riferimento delle API di AWS SDK for Go .

Java

SDK per Java 2.x

> **Note**
>
> C'è dell'altro GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.rds.RdsClient;
import software.amazon.awssdk.services.rds.model.DescribeDbInstancesResponse;
import software.amazon.awssdk.services.rds.model.DBInstance;
import software.amazon.awssdk.services.rds.model.RdsException;
import java.util.List;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class DescribeDBInstances {

    public static void main(String[] args) {
        Region region = Region.US_EAST_1;
        RdsClient rdsClient = RdsClient.builder()
                .region(region)
```

```
                .build();

        describeInstances(rdsClient);
        rdsClient.close();
    }

    public static void describeInstances(RdsClient rdsClient) {
        try {
            DescribeDbInstancesResponse response =
 rdsClient.describeDBInstances();
            List<DBInstance> instanceList = response.dbInstances();
            for (DBInstance instance : instanceList) {
                System.out.println("Instance ARN is: " +
 instance.dbInstanceArn());
                System.out.println("The Engine is " + instance.engine());
                System.out.println("Connection endpoint is" +
 instance.endpoint().address());
            }

        } catch (RdsException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }
    }
}
```

- Per informazioni dettagliate sull'API, consulta DescribeDBInstances nella Documentazione di riferimento delle API di AWS SDK for Java 2.x .

Kotlin

SDK per Kotlin

> **Note**
>
> C'è dell'altro GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
suspend fun describeInstances() {
```

```
    RdsClient { region = "us-west-2" }.use { rdsClient ->
        val response = rdsClient.describeDbInstances(DescribeDbInstancesRequest
 {})
        response.dbInstances?.forEach { instance ->
            println("Instance Identifier is ${instance.dbInstanceIdentifier}")
            println("The Engine is ${instance.engine}")
            println("Connection endpoint is ${instance.endpoint?.address}")
        }
    }
}
```

- Per informazioni dettagliate sull'API, consulta DescribeDBInstances nella Documentazione di riferimento delle API di AWS SDK per Kotlin.

PHP

SDK per PHP

> **Note**
>
> C'è dell'altro GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
require __DIR__ . '/vendor/autoload.php';

use Aws\Exception\AwsException;


//Create an RDSClient
$rdsClient = new Aws\Rds\RdsClient([
    'region' => 'us-east-2'
]);

try {
    $result = $rdsClient->describeDBInstances();
    foreach ($result['DBInstances'] as $instance) {
        print('<p>DB Identifier: ' . $instance['DBInstanceIdentifier']);
```

```
        print('<br />Endpoint: ' . $instance['Endpoint']["Address"]
            . ':' . $instance['Endpoint']["Port"]);
        print('<br />Current Status: ' . $instance["DBInstanceStatus"]);
        print('</p>');
    }
    print(" Raw Result ");
    var_dump($result);
} catch (AwsException $e) {
    echo $e->getMessage();
    echo "\n";
}
```

- Per informazioni dettagliate sull'API, consulta DescribeDBInstances nella Documentazione di riferimento delle API di AWS SDK for PHP .

Python

SDK per Python (Boto3)

> **Note**
>
> C'è dell'altro GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
class InstanceWrapper:
    """Encapsulates Amazon RDS DB instance actions."""

    def __init__(self, rds_client):
        """
        :param rds_client: A Boto3 Amazon RDS client.
        """
        self.rds_client = rds_client

    @classmethod
    def from_client(cls):
        """
        Instantiates this class from a Boto3 client.
        """
```

```
        rds_client = boto3.client("rds")
        return cls(rds_client)


    def get_db_instance(self, instance_id):
        """
        Gets data about a DB instance.

        :param instance_id: The ID of the DB instance to retrieve.
        :return: The retrieved DB instance.
        """
        try:
            response = self.rds_client.describe_db_instances(
                DBInstanceIdentifier=instance_id
            )
            db_inst = response["DBInstances"][0]
        except ClientError as err:
            if err.response["Error"]["Code"] == "DBInstanceNotFound":
                logger.info("Instance %s does not exist.", instance_id)
            else:
                logger.error(
                    "Couldn't get DB instance %s. Here's why: %s: %s",
                    instance_id,
                    err.response["Error"]["Code"],
                    err.response["Error"]["Message"],
                )
                raise
        else:
            return db_inst
```

- Per informazioni dettagliate sull'API, consulta DescribeDBInstances nella Documentazione di riferimento dell'API SDK AWS per Python (Boto3).

Ruby

SDK per Ruby

> **Note**
>
> C'è dell'altro GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
require "aws-sdk-rds"  # v2: require 'aws-sdk'

# List all Amazon Relational Database Service (Amazon RDS) DB instances.
#
# @param rds_resource [Aws::RDS::Resource] An SDK for Ruby Amazon RDS resource.
# @return [Array, nil] List of all DB instances, or nil if error.
def list_instances(rds_resource)
  db_instances = []
  rds_resource.db_instances.each do |i|
    db_instances.append({
                          "name": i.id,
                          "status": i.db_instance_status
                        })
  end
  db_instances
rescue Aws::Errors::ServiceError => e
  puts "Couldn't list instances:\n#{e.message}"
end
```

- Per informazioni dettagliate sull'API, consulta DescribeDBInstances nella Documentazione di riferimento delle API di AWS SDK for Ruby .

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`DescribeDBParameterGroups`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`DescribeDBParameterGroups`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Nozioni di base sulle istanze DB

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    /// <summary>
    /// Get descriptions of DB parameter groups.
    /// </summary>
    /// <param name="name">Optional name of the DB parameter group to describe.</
param>
    /// <returns>The list of DB parameter group descriptions.</returns>
    public async Task<List<DBParameterGroup>> DescribeDBParameterGroups(string
 name = null)
    {
        var response = await _amazonRDS.DescribeDBParameterGroupsAsync(
            new DescribeDBParameterGroupsRequest()
            {
                DBParameterGroupName = name
            });
        return response.DBParameterGroups;
    }
```

- Per i dettagli sull'API, consulta DescribeDB ParameterGroups in AWS SDK for .NET API Reference.

C++

SDK per C++

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
        Aws::Client::ClientConfiguration clientConfig;
        // Optional: Set to the AWS Region (overrides config file).
        // clientConfig.region = "us-east-1";

    Aws::RDS::RDSClient client(clientConfig);

        Aws::RDS::Model::DescribeDBParameterGroupsRequest request;
        request.SetDBParameterGroupName(PARAMETER_GROUP_NAME);

        Aws::RDS::Model::DescribeDBParameterGroupsOutcome outcome =
                client.DescribeDBParameterGroups(request);

        if (outcome.IsSuccess()) {
            std::cout << "DB parameter group named '" <<
                      PARAMETER_GROUP_NAME << "' already exists." << std::endl;
            dbParameterGroupFamily = outcome.GetResult().GetDBParameterGroups()
[0].GetDBParameterGroupFamily();
        }

        else {
            std::cerr << "Error with RDS::DescribeDBParameterGroups. "
                      << outcome.GetError().GetMessage()
                      << std::endl;
            return false;
        }
```

- Per i dettagli sull'API, consulta DescribeDB ParameterGroups in AWS SDK for C++ API Reference.

CLI

AWS CLI

Per descrivere il gruppo di parametri DB

L'`describe-db-parameter-groups`esempio seguente recupera i dettagli sui gruppi di parametri DB.

```
aws rds describe-db-parameter-groups
```

Output:

```
{
    "DBParameterGroups": [
        {
            "DBParameterGroupName": "default.aurora-mysql5.7",
            "DBParameterGroupFamily": "aurora-mysql5.7",
            "Description": "Default parameter group for aurora-mysql5.7",
            "DBParameterGroupArn": "arn:aws:rds:us-
east-1:123456789012:pg:default.aurora-mysql5.7"
        },
        {
            "DBParameterGroupName": "default.aurora-postgresql9.6",
            "DBParameterGroupFamily": "aurora-postgresql9.6",
            "Description": "Default parameter group for aurora-postgresql9.6",
            "DBParameterGroupArn": "arn:aws:rds:us-
east-1:123456789012:pg:default.aurora-postgresql9.6"
        },
        {
            "DBParameterGroupName": "default.aurora5.6",
            "DBParameterGroupFamily": "aurora5.6",
            "Description": "Default parameter group for aurora5.6",
            "DBParameterGroupArn": "arn:aws:rds:us-
east-1:123456789012:pg:default.aurora5.6"
        },
        {
            "DBParameterGroupName": "default.mariadb10.1",
            "DBParameterGroupFamily": "mariadb10.1",
            "Description": "Default parameter group for mariadb10.1",
            "DBParameterGroupArn": "arn:aws:rds:us-
east-1:123456789012:pg:default.mariadb10.1"
        },
```

```
        ...some output truncated...
    ]
}
```

Per ulteriori informazioni, consulta Utilizzo di gruppi di parametri di database nella Guida per l'utente di Amazon RDS.

- Per i dettagli sull'API, vedete DescribeDB ParameterGroups in Command ReferenceAWS CLI .

Go

SDK per Go V2

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
type DbInstances struct {
 RdsClient *rds.Client
}



// GetParameterGroup gets a DB parameter group by name.
func (instances *DbInstances) GetParameterGroup(parameterGroupName string) (
 *types.DBParameterGroup, error) {
 output, err := instances.RdsClient.DescribeDBParameterGroups(
  context.TODO(), &rds.DescribeDBParameterGroupsInput{
   DBParameterGroupName: aws.String(parameterGroupName),
  })
 if err != nil {
  var notFoundError *types.DBParameterGroupNotFoundFault
  if errors.As(err, &notFoundError) {
   log.Printf("Parameter group %v does not exist.\n", parameterGroupName)
   err = nil
  } else {
   log.Printf("Error getting parameter group %v: %v\n", parameterGroupName, err)
  }
```

```
  return nil, err
 } else {
  return &output.DBParameterGroups[0], err
 }
}
```

- Per i dettagli sull'API, consulta DescribeDB ParameterGroups in AWS SDK for Go API Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
   public static void describeDbParameterGroups(RdsClient rdsClient, String
dbGroupName) {
       try {
           DescribeDbParameterGroupsRequest groupsRequest =
DescribeDbParameterGroupsRequest.builder()
                   .dbParameterGroupName(dbGroupName)
                   .maxRecords(20)
                   .build();

           DescribeDbParameterGroupsResponse response =
rdsClient.describeDBParameterGroups(groupsRequest);
           List<DBParameterGroup> groups = response.dbParameterGroups();
           for (DBParameterGroup group : groups) {
               System.out.println("The group name is " +
group.dbParameterGroupName());
               System.out.println("The group description is " +
group.description());
           }

       } catch (RdsException e) {
```

```
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }
    }
```

- Per i dettagli sull'API, consulta DescribeDB ParameterGroups in AWS SDK for Java 2.x API Reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
class InstanceWrapper:
    """Encapsulates Amazon RDS DB instance actions."""

    def __init__(self, rds_client):
        """
        :param rds_client: A Boto3 Amazon RDS client.
        """
        self.rds_client = rds_client

    @classmethod
    def from_client(cls):
        """
        Instantiates this class from a Boto3 client.
        """
        rds_client = boto3.client("rds")
        return cls(rds_client)


    def get_parameter_group(self, parameter_group_name):
        """
        Gets a DB parameter group.
```

```
        :param parameter_group_name: The name of the parameter group to retrieve.
        :return: The parameter group.
        """
        try:
            response = self.rds_client.describe_db_parameter_groups(
                DBParameterGroupName=parameter_group_name
            )
            parameter_group = response["DBParameterGroups"][0]
        except ClientError as err:
            if err.response["Error"]["Code"] == "DBParameterGroupNotFound":
                logger.info("Parameter group %s does not exist.",
parameter_group_name)
            else:
                logger.error(
                    "Couldn't get parameter group %s. Here's why: %s: %s",
                    parameter_group_name,
                    err.response["Error"]["Code"],
                    err.response["Error"]["Message"],
                )
                raise
        else:
            return parameter_group
```

- Per i dettagli sull'API, consulta DescribeDB ParameterGroups in AWS SDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C' GitHubè di più su. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
require "aws-sdk-rds"  # v2: require 'aws-sdk'

# List all Amazon Relational Database Service (Amazon RDS) parameter groups.
```

```
#
# @param rds_resource [Aws::RDS::Resource] An SDK for Ruby Amazon RDS resource.
# @return [Array, nil] List of all parameter groups, or nil if error.
def list_parameter_groups(rds_resource)
  parameter_groups = []
  rds_resource.db_parameter_groups.each do |p|
    parameter_groups.append({
                              "name": p.db_parameter_group_name,
                              "description": p.description
                            })
  end
  parameter_groups
rescue Aws::Errors::ServiceError => e
  puts "Couldn't list parameter groups:\n #{e.message}"
end
```

- Per i dettagli sull'API, consulta DescribeDB ParameterGroups in AWS SDK for Ruby API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`DescribeDBParameters`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`DescribeDBParameters`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Nozioni di base sulle istanze DB

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
   /// <summary>
   /// Get a list of DB parameters from a specific parameter group.
   /// </summary>
   /// <param name="dbParameterGroupName">Name of a specific DB parameter
group.</param>
   /// <param name="source">Optional source for selecting parameters.</param>
   /// <returns>List of parameter values.</returns>
   public async Task<List<Parameter>> DescribeDBParameters(string
dbParameterGroupName, string source = null)
   {
       var results = new List<Parameter>();
       var paginateParameters = _amazonRDS.Paginators.DescribeDBParameters(
           new DescribeDBParametersRequest()
           {
               DBParameterGroupName = dbParameterGroupName,
               Source = source
           });
       // Get the entire list using the paginator.
       await foreach (var parameters in paginateParameters.Parameters)
       {
           results.Add(parameters);
       }
       return results;
   }
```

- Per informazioni sull'API, consulta DescribeDBParameters nella Documentazione di riferimento dell'API AWS SDK for .NET .

C++

SDK per C++

> **Note**
>
> C'è dell'altro GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
        Aws::Client::ClientConfiguration clientConfig;
        // Optional: Set to the AWS Region (overrides config file).
        // clientConfig.region = "us-east-1";

    Aws::RDS::RDSClient client(clientConfig);


//! Routine which gets DB parameters using the 'DescribeDBParameters' api.
/*!
 \sa getDBParameters()
 \param parameterGroupName: The name of the parameter group.
 \param namePrefix: Prefix string to filter results by parameter name.
 \param source: A source such as 'user', ignored if empty.
 \param parametersResult: Vector of 'Parameter' objects returned by the routine.
 \param client: 'RDSClient' instance.
 \return bool: Successful completion.
 */
bool AwsDoc::RDS::getDBParameters(const Aws::String &parameterGroupName,
                                  const Aws::String &namePrefix,
                                  const Aws::String &source,
                                  Aws::Vector<Aws::RDS::Model::Parameter>
 &parametersResult,
                                  const Aws::RDS::RDSClient &client) {
    Aws::String marker;
    do {
        Aws::RDS::Model::DescribeDBParametersRequest request;
        request.SetDBParameterGroupName(PARAMETER_GROUP_NAME);
        if (!marker.empty()) {
            request.SetMarker(marker);
        }
        if (!source.empty()) {
            request.SetSource(source);
```

```
        }

        Aws::RDS::Model::DescribeDBParametersOutcome outcome =
                client.DescribeDBParameters(request);

        if (outcome.IsSuccess()) {
            const Aws::Vector<Aws::RDS::Model::Parameter> &parameters =
                    outcome.GetResult().GetParameters();
            for (const Aws::RDS::Model::Parameter &parameter: parameters) {
                if (!namePrefix.empty()) {
                    if (parameter.GetParameterName().find(namePrefix) == 0) {
                        parametersResult.push_back(parameter);
                    }
                }
                else {
                    parametersResult.push_back(parameter);
                }
            }

            marker = outcome.GetResult().GetMarker();
        }
        else {
            std::cerr << "Error with RDS::DescribeDBParameters. "
                      << outcome.GetError().GetMessage()
                      << std::endl;
            return false;
        }
    } while (!marker.empty());

    return true;
}
```

- Per informazioni sull'API, consulta DescribeDBParameters nella Documentazione di riferimento dell'API AWS SDK for C++ .

CLI

AWS CLI

Per descrivere i parametri in un gruppo di parametri DB

L'`describe-db-parameters`esempio seguente recupera i dettagli del gruppo di parametri DB specificato.

```
aws rds describe-db-parameters \
    --db-parameter-group-name mydbpg
```

Output:

```
{
    "Parameters": [
        {
            "ParameterName": "allow-suspicious-udfs",
            "Description": "Controls whether user-defined functions that have
 only an xxx symbol for the main function can be loaded",
            "Source": "engine-default",
            "ApplyType": "static",
            "DataType": "boolean",
            "AllowedValues": "0,1",
            "IsModifiable": false,
            "ApplyMethod": "pending-reboot"
        },
        {
            "ParameterName": "auto_generate_certs",
            "Description": "Controls whether the server autogenerates SSL key and
 certificate files in the data directory, if they do not already exist.",
            "Source": "engine-default",
            "ApplyType": "static",
            "DataType": "boolean",
            "AllowedValues": "0,1",
            "IsModifiable": false,
            "ApplyMethod": "pending-reboot"
        },
        ...some output truncated...
    ]
}
```

Per ulteriori informazioni, consulta Utilizzo di gruppi di parametri di database nella Guida per l'utente di Amazon RDS.

- Per i dettagli sull'API, consulta DescribedBParameters in Command AWS CLI Reference.

Go

SDK per Go V2

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
type DbInstances struct {
 RdsClient *rds.Client
}



// GetParameters gets the parameters that are contained in a DB parameter group.
func (instances *DbInstances) GetParameters(parameterGroupName string, source
 string) (
 []types.Parameter, error) {

 var output *rds.DescribeDBParametersOutput
 var params []types.Parameter
 var err error
 parameterPaginator := rds.NewDescribeDBParametersPaginator(instances.RdsClient,
  &rds.DescribeDBParametersInput{
   DBParameterGroupName: aws.String(parameterGroupName),
   Source:               aws.String(source),
  })
 for parameterPaginator.HasMorePages() {
  output, err = parameterPaginator.NextPage(context.TODO())
  if err != nil {
   log.Printf("Couldn't get parameters for %v: %v\n", parameterGroupName, err)
   break
  } else {
   params = append(params, output.Parameters...)
  }
 }
 return params, err
}
```

- Per informazioni sull'API, consulta DescribeDBParameters nella Documentazione di riferimento dell'API AWS SDK for Go .

Java

SDK per Java 2.x

> **Note**
>
> C'è dell'altro GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
   // Retrieve parameters in the group.
   public static void describeDbParameters(RdsClient rdsClient, String
dbGroupName, int flag) {
       try {
           DescribeDbParametersRequest dbParameterGroupsRequest;
           if (flag == 0) {
               dbParameterGroupsRequest = DescribeDbParametersRequest.builder()
                       .dbParameterGroupName(dbGroupName)
                       .build();
           } else {
               dbParameterGroupsRequest = DescribeDbParametersRequest.builder()
                       .dbParameterGroupName(dbGroupName)
                       .source("user")
                       .build();
           }

           DescribeDbParametersResponse response =
rdsClient.describeDBParameters(dbParameterGroupsRequest);
           List<Parameter> dbParameters = response.parameters();
           String paraName;
           for (Parameter para : dbParameters) {
               // Only print out information about either auto_increment_offset
or
               // auto_increment_increment.
               paraName = para.parameterName();
               if ((paraName.compareTo("auto_increment_offset") == 0)
```

```
                        || (paraName.compareTo("auto_increment_increment ") ==
0)) {
                    System.out.println("*** The parameter name is  " + paraName);
                    System.out.println("*** The parameter value is  " +
para.parameterValue());
                    System.out.println("*** The parameter data type is " +
para.dataType());
                    System.out.println("*** The parameter description is " +
para.description());
                    System.out.println("*** The parameter allowed values  is " +
para.allowedValues());
                }
            }

        } catch (RdsException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }
    }
```

- Per informazioni sull'API, consulta DescribeDBParameters nella Documentazione di riferimento dell'API AWS SDK for Java 2.x .

Python

SDK per Python (Boto3)

> **Note**
>
> C'è dell'altro GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
class InstanceWrapper:
    """Encapsulates Amazon RDS DB instance actions."""

    def __init__(self, rds_client):
        """
        :param rds_client: A Boto3 Amazon RDS client.
        """
```

```
        self.rds_client = rds_client

    @classmethod
    def from_client(cls):
        """
        Instantiates this class from a Boto3 client.
        """
        rds_client = boto3.client("rds")
        return cls(rds_client)


    def get_parameters(self, parameter_group_name, name_prefix="", source=None):
        """
        Gets the parameters that are contained in a DB parameter group.

        :param parameter_group_name: The name of the parameter group to query.
        :param name_prefix: When specified, the retrieved list of parameters is
filtered
                            to contain only parameters that start with this
prefix.
        :param source: When specified, only parameters from this source are
retrieved.
                       For example, a source of 'user' retrieves only parameters
that
                       were set by a user.
        :return: The list of requested parameters.
        """
        try:
            kwargs = {"DBParameterGroupName": parameter_group_name}
            if source is not None:
                kwargs["Source"] = source
            parameters = []
            paginator = self.rds_client.get_paginator("describe_db_parameters")
            for page in paginator.paginate(**kwargs):
                parameters += [
                    p
                    for p in page["Parameters"]
                    if p["ParameterName"].startswith(name_prefix)
                ]
        except ClientError as err:
            logger.error(
                "Couldn't get parameters for %s. Here's why: %s: %s",
                parameter_group_name,
                err.response["Error"]["Code"],
```

```
                err.response["Error"]["Message"],
            )
            raise
        else:
            return parameters
```

- Per informazioni sull'API, consulta DescribeDBParameters nella Documentazione di riferimento per l'API SDK AWS per Python (Boto3).

Ruby

SDK per Ruby

> **Note**
>
> C'è dell'altro GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
require "aws-sdk-rds"  # v2: require 'aws-sdk'

# List all Amazon Relational Database Service (Amazon RDS) parameter groups.
#
# @param rds_resource [Aws::RDS::Resource] An SDK for Ruby Amazon RDS resource.
# @return [Array, nil] List of all parameter groups, or nil if error.
def list_parameter_groups(rds_resource)
  parameter_groups = []
  rds_resource.db_parameter_groups.each do |p|
    parameter_groups.append({
                              "name": p.db_parameter_group_name,
                              "description": p.description
                            })
  end
  parameter_groups
rescue Aws::Errors::ServiceError => e
  puts "Couldn't list parameter groups:\n #{e.message}"
end
```

- Per informazioni sull'API, consulta DescribeDBParameters nella Documentazione di riferimento dell'API AWS SDK for Ruby .

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`DescribeDBSnapshots`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`DescribeDBSnapshots`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Nozioni di base sulle istanze DB

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
   /// <summary>
   /// Return a list of DB snapshots for a particular DB instance.
   /// </summary>
   /// <param name="dbInstanceIdentifier">DB instance identifier.</param>
   /// <returns>List of DB snapshots.</returns>
   public async Task<List<DBSnapshot>> DescribeDBSnapshots(string
dbInstanceIdentifier)
   {
       var results = new List<DBSnapshot>();
       var snapshotsPaginator = _amazonRDS.Paginators.DescribeDBSnapshots(
           new DescribeDBSnapshotsRequest()
           {
```

```
                DBInstanceIdentifier = dbInstanceIdentifier
            });

        // Get the entire list using the paginator.
        await foreach (var snapshots in snapshotsPaginator.DBSnapshots)
        {
            results.Add(snapshots);
        }
        return results;
    }
```

- Per informazioni dettagliate sull'API, consulta DescribeDBSnapshots nella Documentazione di riferimento dell'API AWS SDK for .NET .

C++

SDK per C++

> **Note**
>
> C'è dell'altro GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
        Aws::Client::ClientConfiguration clientConfig;
        // Optional: Set to the AWS Region (overrides config file).
        // clientConfig.region = "us-east-1";

    Aws::RDS::RDSClient client(clientConfig);

            Aws::RDS::Model::DescribeDBSnapshotsRequest request;
            request.SetDBSnapshotIdentifier(snapshotID);

            Aws::RDS::Model::DescribeDBSnapshotsOutcome outcome =
                    client.DescribeDBSnapshots(request);

            if (outcome.IsSuccess()) {
                snapshot = outcome.GetResult().GetDBSnapshots()[0];
            }
```

```
            else {
                std::cerr << "Error with RDS::DescribeDBSnapshots. "
                          << outcome.GetError().GetMessage()
                          << std::endl;
                cleanUpResources(PARAMETER_GROUP_NAME, DB_INSTANCE_IDENTIFIER,
 client);
                return false;
            }
```

- Per informazioni dettagliate sull'API, consulta DescribeDBSnapshots nella Documentazione di riferimento dell'API AWS SDK for C++ .

CLI

AWS CLI

Esempio 1: descrivere uno snapshot DB per un'istanza DB

L'`describe-db-snapshots`esempio seguente recupera i dettagli di uno snapshot DB per un'istanza DB.

```
aws rds describe-db-snapshots \
    --db-snapshot-identifier mydbsnapshot
```

Output:

```
{
    "DBSnapshots": [
        {
            "DBSnapshotIdentifier": "mydbsnapshot",
            "DBInstanceIdentifier": "mysqldb",
            "SnapshotCreateTime": "2018-02-08T22:28:08.598Z",
            "Engine": "mysql",
            "AllocatedStorage": 20,
            "Status": "available",
            "Port": 3306,
            "AvailabilityZone": "us-east-1f",
            "VpcId": "vpc-6594f31c",
            "InstanceCreateTime": "2018-02-08T22:24:55.973Z",
            "MasterUsername": "mysqladmin",
```

```
            "EngineVersion": "5.6.37",
            "LicenseModel": "general-public-license",
            "SnapshotType": "manual",
            "OptionGroupName": "default:mysql-5-6",
            "PercentProgress": 100,
            "StorageType": "gp2",
            "Encrypted": false,
            "DBSnapshotArn": "arn:aws:rds:us-
east-1:123456789012:snapshot:mydbsnapshot",
            "IAMDatabaseAuthenticationEnabled": false,
            "ProcessorFeatures": [],
            "DbiResourceId": "db-AKIAIOSFODNN7EXAMPLE"
        }
    ]
}
```

Per ulteriori informazioni, consulta Creating a DB Snapshot nella Amazon RDS User Guide.

Esempio 2: per trovare il numero di istantanee scattate manualmente

L'`describe-db-snapshots`esempio seguente utilizza l'`length`operatore nell'`--query`opzione per restituire il numero di istantanee manuali che sono state scattate in una particolare AWS regione.

```
aws rds describe-db-snapshots \
    --snapshot-type manual \
    --query "length(*[].{DBSnapshots:SnapshotType})" \
    --region eu-central-1
```

Output:

```
35
```

Per ulteriori informazioni, consulta Creating a DB Snapshot nella Amazon RDS User Guide.

- Per i dettagli sull'API, consulta DescribedBSnapshots in Command Reference AWS CLI .

Go

SDK per Go V2

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
type DbInstances struct {
 RdsClient *rds.Client
}



// GetSnapshot gets a DB instance snapshot.
func (instances *DbInstances) GetSnapshot(snapshotName string)
 (*types.DBSnapshot, error) {
 output, err := instances.RdsClient.DescribeDBSnapshots(context.TODO(),
  &rds.DescribeDBSnapshotsInput{
   DBSnapshotIdentifier: aws.String(snapshotName),
  })
 if err != nil {
  log.Printf("Couldn't get snapshot %v: %v\n", snapshotName, err)
  return nil, err
 } else {
  return &output.DBSnapshots[0], nil
 }
}
```

- Per informazioni dettagliate sull'API, consulta DescribeDBSnapshots nella Documentazione di riferimento dell'API AWS SDK for Go .

Python

SDK per Python (Boto3)

> **Note**
>
> C'è dell'altro GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
class InstanceWrapper:
    """Encapsulates Amazon RDS DB instance actions."""

    def __init__(self, rds_client):
        """
        :param rds_client: A Boto3 Amazon RDS client.
        """
        self.rds_client = rds_client

    @classmethod
    def from_client(cls):
        """
        Instantiates this class from a Boto3 client.
        """
        rds_client = boto3.client("rds")
        return cls(rds_client)


    def get_snapshot(self, snapshot_id):
        """
        Gets a DB instance snapshot.

        :param snapshot_id: The ID of the snapshot to retrieve.
        :return: The retrieved snapshot.
        """
        try:
            response = self.rds_client.describe_db_snapshots(
                DBSnapshotIdentifier=snapshot_id
            )
            snapshot = response["DBSnapshots"][0]
        except ClientError as err:
            logger.error(
```

```
                "Couldn't get snapshot %s. Here's why: %s: %s",
                snapshot_id,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return snapshot
```

- Per informazioni sull'API, consulta DescribeDBSnapshots nella Documentazione di riferimento per l'API SDK AWS per Python (Boto3).

Ruby

SDK per Ruby

> **Note**
>
> C'è dell'altro GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
require "aws-sdk-rds" # v2: require 'aws-sdk'

# List all Amazon Relational Database Service (Amazon RDS) DB instance
# snapshots.
#
# @param rds_resource [Aws::RDS::Resource] An SDK for Ruby Amazon RDS resource.
# @return instance_snapshots [Array, nil] All instance snapshots, or nil if
 error.
def list_instance_snapshots(rds_resource)
  instance_snapshots = []
  rds_resource.db_snapshots.each do |s|
    instance_snapshots.append({
                                "id": s.snapshot_id,
                                "status": s.status
                              })
  end
  instance_snapshots
```

```
rescue Aws::Errors::ServiceError => e
  puts "Couldn't list instance snapshots:\n #{e.message}"
end
```

- Per informazioni dettagliate sull'API, consulta DescribeDBSnapshots nella Documentazione di riferimento dell'API AWS SDK for Ruby .

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`DescribeOrderableDBInstanceOptions`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`DescribeOrderableDBInstanceOptions`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Nozioni di base sulle istanze DB

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    /// <summary>
    /// Get a list of orderable DB instance options for a specific
    /// engine and engine version.
    /// </summary>
    /// <param name="engine">Name of the engine.</param>
```

```
    /// <param name="engineVersion">Version of the engine.</param>
    /// <returns>List of OrderableDBInstanceOptions.</returns>
    public async Task<List<OrderableDBInstanceOption>>
DescribeOrderableDBInstanceOptions(string engine, string engineVersion)
    {
        // Use a paginator to get a list of DB instance options.
        var results = new List<OrderableDBInstanceOption>();
        var paginateInstanceOptions =
_amazonRDS.Paginators.DescribeOrderableDBInstanceOptions(
            new DescribeOrderableDBInstanceOptionsRequest()
            {
                Engine = engine,
                EngineVersion = engineVersion,
            });
        // Get the entire list using the paginator.
        await foreach (var instanceOptions in
paginateInstanceOptions.OrderableDBInstanceOptions)
        {
            results.Add(instanceOptions);
        }
        return results;
    }
```

- Per i dettagli sull'API, consulta DescribeOrderableDB InstanceOptions in AWS SDK for .NET API Reference.

C++

SDK per C++

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
        Aws::Client::ClientConfiguration clientConfig;
        // Optional: Set to the AWS Region (overrides config file).
        // clientConfig.region = "us-east-1";
```

```
    Aws::RDS::RDSClient client(clientConfig);


//! Routine which gets available 'micro' DB instance classes, displays the list
//! to the user, and returns the user selection.
/*!
 \sa chooseMicroDBInstanceClass()
 \param engineName: The DB engine name.
 \param engineVersion: The DB engine version.
 \param dbInstanceClass: String for DB instance class chosen by the user.
 \param client: 'RDSClient' instance.
 \return bool: Successful completion.
 */
bool AwsDoc::RDS::chooseMicroDBInstanceClass(const Aws::String &engine,
                                             const Aws::String &engineVersion,
                                             Aws::String &dbInstanceClass,
                                             const Aws::RDS::RDSClient &client) {
    std::vector<Aws::String> instanceClasses;
    Aws::String marker;
    do {
        Aws::RDS::Model::DescribeOrderableDBInstanceOptionsRequest request;
        request.SetEngine(engine);
        request.SetEngineVersion(engineVersion);
        if (!marker.empty()) {
            request.SetMarker(marker);
        }

        Aws::RDS::Model::DescribeOrderableDBInstanceOptionsOutcome outcome =
                client.DescribeOrderableDBInstanceOptions(request);

        if (outcome.IsSuccess()) {
            const Aws::Vector<Aws::RDS::Model::OrderableDBInstanceOption>
 &options =
                    outcome.GetResult().GetOrderableDBInstanceOptions();
            for (const Aws::RDS::Model::OrderableDBInstanceOption &option:
 options) {
                const Aws::String &instanceClass = option.GetDBInstanceClass();
                if (instanceClass.find("micro") != std::string::npos) {
                    if (std::find(instanceClasses.begin(), instanceClasses.end(),
                                  instanceClass) ==
                        instanceClasses.end()) {
                        instanceClasses.push_back(instanceClass);
                    }
```

```
                }
            }
            marker = outcome.GetResult().GetMarker();
        }
        else {
            std::cerr << "Error with RDS::DescribeOrderableDBInstanceOptions. "
                      << outcome.GetError().GetMessage()
                      << std::endl;
            return false;
        }
    } while (!marker.empty());

    std::cout << "The available micro DB instance classes for your database
 engine are:"
              << std::endl;
    for (int i = 0; i < instanceClasses.size(); ++i) {
        std::cout << "   " << i + 1 << ": " << instanceClasses[i] << std::endl;
    }

    int choice = askQuestionForIntRange(
            "Which micro DB instance class do you want to use? ",
            1, static_cast<int>(instanceClasses.size()));
    dbInstanceClass = instanceClasses[choice - 1];
    return true;
}
```

- Per i dettagli sull'API, consulta DescribeOrderableDB InstanceOptions in AWS SDK for C++ API Reference.

CLI

AWS CLI

Per descrivere le opzioni delle istanze DB ordinabili

L'`describe-orderable-db-instance-options`esempio seguente recupera i dettagli sulle opzioni ordinabili per le istanze DB che eseguono il motore MySQL DB.

```
aws rds describe-orderable-db-instance-options \
    --engine mysql
```

Output:

```
{
    "OrderableDBInstanceOptions": [
        {
            "MinStorageSize": 5,
            "ReadReplicaCapable": true,
            "MaxStorageSize": 6144,
            "AvailabilityZones": [
                {
                    "Name": "us-east-1a"
                },
                {
                    "Name": "us-east-1b"
                },
                {
                    "Name": "us-east-1c"
                },
                {
                    "Name": "us-east-1d"
                }
            ],
            "SupportsIops": false,
            "AvailableProcessorFeatures": [],
            "MultiAZCapable": true,
            "DBInstanceClass": "db.m1.large",
            "Vpc": true,
            "StorageType": "gp2",
            "LicenseModel": "general-public-license",
            "EngineVersion": "5.5.46",
            "SupportsStorageEncryption": false,
            "SupportsEnhancedMonitoring": true,
            "Engine": "mysql",
            "SupportsIAMDatabaseAuthentication": false,
            "SupportsPerformanceInsights": false
        }
    ]
    ...some output truncated...
}
```

- Per i dettagli sull'API, consulta DescribeOrderable DB in Command Reference. InstanceOptions AWS CLI

Go

SDK per Go V2

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
type DbInstances struct {
 RdsClient *rds.Client
}



// GetOrderableInstances uses a paginator to get DB instance options that can be
 used to create DB instances that are
// compatible with a set of specifications.
func (instances *DbInstances) GetOrderableInstances(engine string, engineVersion
 string) (
 []types.OrderableDBInstanceOption, error) {

 var output *rds.DescribeOrderableDBInstanceOptionsOutput
 var instanceOptions []types.OrderableDBInstanceOption
 var err error
 orderablePaginator :=
 rds.NewDescribeOrderableDBInstanceOptionsPaginator(instances.RdsClient,
  &rds.DescribeOrderableDBInstanceOptionsInput{
   Engine:        aws.String(engine),
   EngineVersion: aws.String(engineVersion),
  })
 for orderablePaginator.HasMorePages() {
  output, err = orderablePaginator.NextPage(context.TODO())
  if err != nil {
   log.Printf("Couldn't get orderable DB instance options: %v\n", err)
   break
  } else {
   instanceOptions = append(instanceOptions,
 output.OrderableDBInstanceOptions...)
  }
```

```
 }
 return instanceOptions, err
}
```

- Per i dettagli sull'API, consulta DescribeOrderableDB InstanceOptions in AWS SDK for Go API Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
   // Get a list of allowed engine versions.
   public static void getAllowedEngines(RdsClient rdsClient, String
dbParameterGroupFamily) {
       try {
           DescribeDbEngineVersionsRequest versionsRequest =
DescribeDbEngineVersionsRequest.builder()
                   .dbParameterGroupFamily(dbParameterGroupFamily)
                   .engine("mysql")
                   .build();

           DescribeDbEngineVersionsResponse response =
rdsClient.describeDBEngineVersions(versionsRequest);
           List<DBEngineVersion> dbEngines = response.dbEngineVersions();
           for (DBEngineVersion dbEngine : dbEngines) {
               System.out.println("The engine version is " +
dbEngine.engineVersion());
               System.out.println("The engine description is " +
dbEngine.dbEngineDescription());
           }

       } catch (RdsException e) {
           System.out.println(e.getLocalizedMessage());
```

```
            System.exit(1);
        }
    }
```

- Per i dettagli sull'API, consulta DescribeOrderableDB InstanceOptions in AWS SDK for Java 2.x API Reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
class InstanceWrapper:
    """Encapsulates Amazon RDS DB instance actions."""

    def __init__(self, rds_client):
        """
        :param rds_client: A Boto3 Amazon RDS client.
        """
        self.rds_client = rds_client

    @classmethod
    def from_client(cls):
        """
        Instantiates this class from a Boto3 client.
        """
        rds_client = boto3.client("rds")
        return cls(rds_client)


    def get_orderable_instances(self, db_engine, db_engine_version):
        """
        Gets DB instance options that can be used to create DB instances that are
        compatible with a set of specifications.
```

```
        :param db_engine: The database engine that must be supported by the DB
instance.
        :param db_engine_version: The engine version that must be supported by
the DB instance.
        :return: The list of DB instance options that can be used to create a
compatible DB instance.
        """
        try:
            inst_opts = []
            paginator = self.rds_client.get_paginator(
                "describe_orderable_db_instance_options"
            )
            for page in paginator.paginate(
                Engine=db_engine, EngineVersion=db_engine_version
            ):
                inst_opts += page["OrderableDBInstanceOptions"]
        except ClientError as err:
            logger.error(
                "Couldn't get orderable DB instances. Here's why: %s: %s",
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return inst_opts
```

- Per i dettagli sull'API, consulta DescribeOrderableDB InstanceOptions in AWS SDK for Python (Boto3) API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`GenerateRDSAuthToken`** con un AWS SDK o una CLI

Il seguente esempio di codice mostra come utilizzare`GenerateRDSAuthToken`.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Usa la RdsUtilitiesclasse per generare un token di autenticazione.

```
public class GenerateRDSAuthToken {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                    <dbInstanceIdentifier> <masterUsername>

                Where:
                    dbInstanceIdentifier - The database instance identifier.\s
                    masterUsername - The master user name.\s
                """;

        if (args.length != 2) {
            System.out.println(usage);
            System.exit(1);
        }

        String dbInstanceIdentifier = args[0];
        String masterUsername = args[1];
        Region region = Region.US_WEST_2;
        RdsClient rdsClient = RdsClient.builder()
                .region(region)
                .build();

        String token = getAuthToken(rdsClient, dbInstanceIdentifier,
 masterUsername);
        System.out.println("The token response is " + token);
    }

    public static String getAuthToken(RdsClient rdsClient, String
 dbInstanceIdentifier, String masterUsername) {
```

```
        RdsUtilities utilities = rdsClient.utilities();
        try {
            GenerateAuthenticationTokenRequest tokenRequest =
 GenerateAuthenticationTokenRequest.builder()
                    .credentialsProvider(ProfileCredentialsProvider.create())
                    .username(masterUsername)
                    .port(3306)
                    .hostname(dbInstanceIdentifier)
                    .build();

            return utilities.generateAuthenticationToken(tokenRequest);

        } catch (RdsException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }
        return "";
    }
}
```

- Per i dettagli sull'API, consulta GenerateRDS AuthToken in AWS SDK for Java 2.x API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`ModifyDBInstance`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`ModifyDBInstance`.

CLI

AWS CLI

Esempio 1: modificare un'istanza DB

L'`modify-db-instance`esempio seguente associa un gruppo di opzioni e un gruppo di parametri a un'istanza DB di Microsoft SQL Server compatibile. Il `--apply-immediately`

parametro fa sì che l'opzione e i gruppi di parametri vengano associati immediatamente, anziché attendere la finestra di manutenzione successiva.

```
aws rds modify-db-instance \
    --db-instance-identifier database-2 \
    --option-group-name test-se-2017 \
    --db-parameter-group-name test-sqlserver-se-2017 \
    --apply-immediately
```

Output:

```
{
    "DBInstance": {
        "DBInstanceIdentifier": "database-2",
        "DBInstanceClass": "db.r4.large",
        "Engine": "sqlserver-se",
        "DBInstanceStatus": "available",

        ...output omitted...

        "DBParameterGroups": [
            {
                "DBParameterGroupName": "test-sqlserver-se-2017",
                "ParameterApplyStatus": "applying"
            }
        ],
        "AvailabilityZone": "us-west-2d",

        ...output omitted...

        "MultiAZ": true,
        "EngineVersion": "14.00.3281.6.v1",
        "AutoMinorVersionUpgrade": false,
        "ReadReplicaDBInstanceIdentifiers": [],
        "LicenseModel": "license-included",
        "OptionGroupMemberships": [
            {
                "OptionGroupName": "test-se-2017",
                "Status": "pending-apply"
            }
        ],
        "CharacterSetName": "SQL_Latin1_General_CP1_CI_AS",
        "SecondaryAvailabilityZone": "us-west-2c",
```

```
        "PubliclyAccessible": true,
        "StorageType": "gp2",

        ...output omitted...

        "DeletionProtection": false,
        "AssociatedRoles": [],
        "MaxAllocatedStorage": 1000
    }
}
```

Per ulteriori informazioni, consulta Modificare un'istanza database Amazon RDS nella Amazon RDS User Guide.

Esempio 2: associare il gruppo di sicurezza VPC a un'istanza DB

L'`modify-db-instance`esempio seguente associa un gruppo di sicurezza VPC specifico e rimuove i gruppi di sicurezza DB da un'istanza DB:

```
aws rds modify-db-instance \
    --db-instance-identifier dbName \
    --vpc-security-group-ids sg-ID
```

Output:

```
{
"DBInstance": {
    "DBInstanceIdentifier": "dbName",
    "DBInstanceClass": "db.t3.micro",
    "Engine": "mysql",
    "DBInstanceStatus": "available",
    "MasterUsername": "admin",
    "Endpoint": {
        "Address": "dbName.abcdefghijk.us-west-2.rds.amazonaws.com",
        "Port": 3306,
        "HostedZoneId": "ABCDEFGHIJK1234"
    },
    "AllocatedStorage": 20,
    "InstanceCreateTime": "2024-02-15T00:37:58.793000+00:00",
    "PreferredBackupWindow": "11:57-12:27",
    "BackupRetentionPeriod": 7,
    "DBSecurityGroups": [],
    "VpcSecurityGroups": [
```

```
        {
            "VpcSecurityGroupId": "sg-ID",
            "Status": "active"
        }
    ],
    ... output omitted ...
    "MultiAZ": false,
    "EngineVersion": "8.0.35",
    "AutoMinorVersionUpgrade": true,
    "ReadReplicaDBInstanceIdentifiers": [],
    "LicenseModel": "general-public-license",

    ... output ommited ...
    }
}
```

Per ulteriori informazioni, consulta Controllare l'accesso con gruppi di sicurezza nella Amazon RDS User Guide.

- Per i dettagli sull'API, consulta ModifyDbInstance in Command Reference AWS CLI .

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.rds.RdsClient;
import software.amazon.awssdk.services.rds.model.ModifyDbInstanceRequest;
import software.amazon.awssdk.services.rds.model.ModifyDbInstanceResponse;
import software.amazon.awssdk.services.rds.model.RdsException;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
```

```
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class ModifyDBInstance {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                    <dbInstanceIdentifier> <dbSnapshotIdentifier>\s
                Where:
                    dbInstanceIdentifier - The database instance identifier.\s
                    masterUserPassword - The updated password that corresponds to
 the master user name.\s
                """;

        if (args.length != 2) {
            System.out.println(usage);
            System.exit(1);
        }

        String dbInstanceIdentifier = args[0];
        String masterUserPassword = args[1];
        Region region = Region.US_WEST_2;
        RdsClient rdsClient = RdsClient.builder()
                .region(region)
                .build();

        updateIntance(rdsClient, dbInstanceIdentifier, masterUserPassword);
        rdsClient.close();
    }

    public static void updateIntance(RdsClient rdsClient, String
 dbInstanceIdentifier, String masterUserPassword) {
        try {
            // For a demo - modify the DB instance by modifying the master
 password.
            ModifyDbInstanceRequest modifyDbInstanceRequest =
 ModifyDbInstanceRequest.builder()
                    .dbInstanceIdentifier(dbInstanceIdentifier)
                    .publiclyAccessible(true)
                    .masterUserPassword(masterUserPassword)
                    .build();
```

```
            ModifyDbInstanceResponse instanceResponse =
 rdsClient.modifyDBInstance(modifyDbInstanceRequest);
            System.out.print("The ARN of the modified database is: " +
 instanceResponse.dbInstance().dbInstanceArn());

        } catch (RdsException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }
    }
}
```

- Per informazioni dettagliate sulle API, consulta ModifyDBInstance nella documentazione di riferimento dell'API AWS SDK for Java 2.x .

Kotlin

SDK per Kotlin

> **Note**
>
> C'è dell'altro GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
suspend fun updateIntance(dbInstanceIdentifierVal: String?,
 masterUserPasswordVal: String?) {

    val request = ModifyDbInstanceRequest {
        dbInstanceIdentifier = dbInstanceIdentifierVal
        publiclyAccessible = true
        masterUserPassword = masterUserPasswordVal
    }

    RdsClient { region = "us-west-2" }.use { rdsClient ->
        val instanceResponse = rdsClient.modifyDbInstance(request)
        println("The ARN of the modified database is
 ${instanceResponse.dbInstance?.dbInstanceArn}")
    }
}
```

- Per informazioni dettagliate sull'API, consulta ModifyDBInstance nella Documentazione di riferimento dell'API SDK AWS per Kotlin.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`ModifyDBParameterGroup`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`ModifyDBParameterGroup`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Nozioni di base sulle istanze DB

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    /// <summary>
    /// Update a DB parameter group. Use the action
 DescribeDBParameterGroupsAsync
    /// to determine when the DB parameter group is ready to use.
    /// </summary>
    /// <param name="name">Name of the DB parameter group.</param>
    /// <param name="parameters">List of parameters. Maximum of 20 per request.</
param>
    /// <returns>The updated DB parameter group name.</returns>
    public async Task<string> ModifyDBParameterGroup(
        string name, List<Parameter> parameters)
```

```
    {
        var response = await _amazonRDS.ModifyDBParameterGroupAsync(
            new ModifyDBParameterGroupRequest()
            {
                DBParameterGroupName = name,
                Parameters = parameters,
            });
        return response.DBParameterGroupName;
    }
```

- Per i dettagli sull'API, consulta ModifyDB ParameterGroup in AWS SDK for .NET API Reference.

C++

SDK per C++

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    Aws::Client::ClientConfiguration clientConfig;
    // Optional: Set to the AWS Region (overrides config file).
    // clientConfig.region = "us-east-1";

Aws::RDS::RDSClient client(clientConfig);

    Aws::RDS::Model::ModifyDBParameterGroupRequest request;
    request.SetDBParameterGroupName(PARAMETER_GROUP_NAME);
    request.SetParameters(updateParameters);

    Aws::RDS::Model::ModifyDBParameterGroupOutcome outcome =
            client.ModifyDBParameterGroup(request);

    if (outcome.IsSuccess()) {
        std::cout << "The DB parameter group was successfully modified."
                  << std::endl;
```

```
        }
        else {
            std::cerr << "Error with RDS::ModifyDBParameterGroup. "
                      << outcome.GetError().GetMessage()
                      << std::endl;
        }
```

- Per i dettagli sull'API, consulta ModifyDB ParameterGroup in AWS SDK for C++ API Reference.

CLI

AWS CLI

Per modificare un gruppo di parametri DB

L'`modify-db-parameter-group`esempio seguente modifica il valore del `clr enabled` parametro in un gruppo di parametri DB. Il `--apply-immediately` parametro fa sì che il gruppo di parametri DB venga modificato immediatamente, anziché attendere la finestra di manutenzione successiva.

```
aws rds modify-db-parameter-group \
    --db-parameter-group-name test-sqlserver-se-2017 \
    --parameters "ParameterName='clr
 enabled',ParameterValue=1,ApplyMethod=immediate"
```

Output:

```
{
    "DBParameterGroupName": "test-sqlserver-se-2017"
}
```

Per ulteriori informazioni, consulta Modifying Parameters in a DB Parameter Group nella Amazon RDS User Guide.

- Per i dettagli sull'API, consulta ModifyDB in Command Reference. ParameterGroupAWS CLI

Go

SDK per Go V2

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
type DbInstances struct {
 RdsClient *rds.Client
}



// UpdateParameters updates parameters in a named DB parameter group.
func (instances *DbInstances) UpdateParameters(parameterGroupName string, params
 []types.Parameter) error {
 _, err := instances.RdsClient.ModifyDBParameterGroup(context.TODO(),
  &rds.ModifyDBParameterGroupInput{
   DBParameterGroupName: aws.String(parameterGroupName),
   Parameters:           params,
  })
 if err != nil {
  log.Printf("Couldn't update parameters in %v: %v\n", parameterGroupName, err)
  return err
 } else {
  return nil
 }
}
```

- Per i dettagli sull'API, consulta ModifyDB ParameterGroup in AWS SDK for Go API Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
   // Modify auto_increment_offset and auto_increment_increment parameters.
   public static void modifyDBParas(RdsClient rdsClient, String dbGroupName) {
       try {
           Parameter parameter1 = Parameter.builder()
                   .parameterName("auto_increment_offset")
                   .applyMethod("immediate")
                   .parameterValue("5")
                   .build();

           List<Parameter> paraList = new ArrayList<>();
           paraList.add(parameter1);
           ModifyDbParameterGroupRequest groupRequest =
ModifyDbParameterGroupRequest.builder()
                   .dbParameterGroupName(dbGroupName)
                   .parameters(paraList)
                   .build();

           ModifyDbParameterGroupResponse response =
rdsClient.modifyDBParameterGroup(groupRequest);
           System.out.println("The parameter group " +
response.dbParameterGroupName() + " was successfully modified");

       } catch (RdsException e) {
           System.out.println(e.getLocalizedMessage());
           System.exit(1);
       }
   }
```

- Per i dettagli sull'API, consulta ModifyDB ParameterGroup in AWS SDK for Java 2.x API Reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
class InstanceWrapper:
    """Encapsulates Amazon RDS DB instance actions."""

    def __init__(self, rds_client):
        """
        :param rds_client: A Boto3 Amazon RDS client.
        """
        self.rds_client = rds_client

    @classmethod
    def from_client(cls):
        """
        Instantiates this class from a Boto3 client.
        """
        rds_client = boto3.client("rds")
        return cls(rds_client)


    def update_parameters(self, parameter_group_name, update_parameters):
        """
        Updates parameters in a custom DB parameter group.

        :param parameter_group_name: The name of the parameter group to update.
        :param update_parameters: The parameters to update in the group.
        :return: Data about the modified parameter group.
        """
        try:
            response = self.rds_client.modify_db_parameter_group(
                DBParameterGroupName=parameter_group_name,
 Parameters=update_parameters
            )
        except ClientError as err:
```

```
            logger.error(
                "Couldn't update parameters in %s. Here's why: %s: %s",
                parameter_group_name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return response
```

- Per i dettagli sull'API, consulta ModifyDB ParameterGroup in AWS SDK for Python (Boto3) API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Utilizzo **`RebootDBInstance`** con un AWS SDK o una CLI

I seguenti esempi di codice mostrano come utilizzare`RebootDBInstance`.

CLI

AWS CLI

Per riavviare un'istanza DB

L'`reboot-db-instance`esempio seguente avvia il riavvio dell'istanza DB specificata.

```
aws rds reboot-db-instance \
    --db-instance-identifier test-mysql-instance
```

Output:

```
{
    "DBInstance": {
        "DBInstanceIdentifier": "test-mysql-instance",
        "DBInstanceClass": "db.t3.micro",
        "Engine": "mysql",
        "DBInstanceStatus": "rebooting",
```

```
        "MasterUsername": "admin",
        "Endpoint": {
            "Address": "test-mysql-instance.############.us-
west-2.rds.amazonaws.com",
            "Port": 3306,
            "HostedZoneId": "Z1PVIF0EXAMPLE"
        },

    ... output omitted...

    }
}
```

Per ulteriori informazioni, consulta Rebooting a DB Instance nella Amazon RDS User Guide.

- Per i dettagli sull'API, consulta RebootDBInstance in Command Reference AWS CLI .

Java

SDK per Java 2.x

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.rds.RdsClient;
import software.amazon.awssdk.services.rds.model.RebootDbInstanceRequest;
import software.amazon.awssdk.services.rds.model.RebootDbInstanceResponse;
import software.amazon.awssdk.services.rds.model.RdsException;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
```

```
public class RebootDBInstance {
    public static void main(String[] args) {
        final String usage = """

                Usage:
                    <dbInstanceIdentifier>\s

                Where:
                    dbInstanceIdentifier - The database instance identifier\s
                """;

        if (args.length != 1) {
            System.out.println(usage);
            System.exit(1);
        }

        String dbInstanceIdentifier = args[0];
        Region region = Region.US_WEST_2;
        RdsClient rdsClient = RdsClient.builder()
                .region(region)
                .build();

        rebootInstance(rdsClient, dbInstanceIdentifier);
        rdsClient.close();
    }

    public static void rebootInstance(RdsClient rdsClient, String
 dbInstanceIdentifier) {
        try {
            RebootDbInstanceRequest rebootDbInstanceRequest =
 RebootDbInstanceRequest.builder()
                    .dbInstanceIdentifier(dbInstanceIdentifier)
                    .build();

            RebootDbInstanceResponse instanceResponse =
 rdsClient.rebootDBInstance(rebootDbInstanceRequest);
            System.out.print("The database " +
 instanceResponse.dbInstance().dbInstanceArn() + " was rebooted");

        } catch (RdsException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }
    }
```

```
}
```

- Per informazioni dettagliate sulle API, consulta RebootDBInstance nella documentazione di riferimento dell'API AWS SDK for Java 2.x .

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Scenari per Amazon RDS che utilizzano SDK AWS

I seguenti esempi di codice mostrano come implementare scenari comuni in Amazon RDS con AWS SDK. Questi scenari illustrano come eseguire attività specifiche richiamando più funzioni in Amazon RDS. Ogni scenario include un collegamento a GitHub, dove puoi trovare istruzioni su come configurare ed eseguire il codice.

Esempi

- Inizia a usare le istanze DB di Amazon RDS utilizzando un SDK AWS

Inizia a usare le istanze DB di Amazon RDS utilizzando un SDK AWS

Gli esempi di codice seguenti mostrano come:

- Creare un gruppo di parametri database personalizzati e imposta i relativi valori.
- Creare un'istanza database configurata per utilizzare il gruppo di parametri. L'istanza DB contiene anche un database.
- Acquisire uno snapshot dell'istanza.
- Eliminare l'istanza e il gruppo di parametri.

.NET

AWS SDK for .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Esegui uno scenario interattivo al prompt dei comandi.

```
/// <summary>
/// Scenario for RDS DB instance example.
/// </summary>
public class RDSInstanceScenario
{
    /*
    Before running this .NET code example, set up your development environment,
 including your credentials.

    This .NET example performs the following tasks:
    1.  Returns a list of the available DB engine families using the
 DescribeDBEngineVersionsAsync method.
    2.  Selects an engine family and creates a custom DB parameter group using
 the CreateDBParameterGroupAsync method.
    3.  Gets the parameter groups using the DescribeDBParameterGroupsAsync
 method.
    4.  Gets parameters in the group using the DescribeDBParameters method.
    5.  Parses and displays parameters in the group.
    6.  Modifies both the auto_increment_offset and auto_increment_increment
 parameters
        using the ModifyDBParameterGroupAsync method.
    7.  Gets and displays the updated parameters using the DescribeDBParameters
 method with a source of "user".
    8.  Gets a list of allowed engine versions using the
 DescribeDBEngineVersionsAsync method.
    9.  Displays and selects from a list of micro instance classes available for
 the selected engine and version.
    10. Creates an RDS DB instance that contains a MySql database and uses the
 parameter group
        using the CreateDBInstanceAsync method.
```

```
   11. Waits for DB instance to be ready using the DescribeDBInstancesAsync
method.
   12. Prints out the connection endpoint string for the new DB instance.
   13. Creates a snapshot of the DB instance using the CreateDBSnapshotAsync
method.
   14. Waits for DB snapshot to be ready using the DescribeDBSnapshots method.
   15. Deletes the DB instance using the DeleteDBInstanceAsync method.
   16. Waits for DB instance to be deleted using the DescribeDbInstances method.
   17. Deletes the parameter group using the DeleteDBParameterGroupAsync.
   */

   private static readonly string sepBar = new('-', 80);
   private static RDSWrapper rdsWrapper = null!;
   private static ILogger logger = null!;
   private static readonly string engine = "mysql";
   static async Task Main(string[] args)
   {
       // Set up dependency injection for the Amazon RDS service.
       using var host = Host.CreateDefaultBuilder(args)
           .ConfigureLogging(logging =>
               logging.AddFilter("System", LogLevel.Debug)
                   .AddFilter<DebugLoggerProvider>("Microsoft",
LogLevel.Information)
                   .AddFilter<ConsoleLoggerProvider>("Microsoft",
LogLevel.Trace))
           .ConfigureServices((_, services) =>
               services.AddAWSService<IAmazonRDS>()
                   .AddTransient<RDSWrapper>()
           )
           .Build();

       logger = LoggerFactory.Create(builder =>
       {
           builder.AddConsole();
       }).CreateLogger<RDSInstanceScenario>();

       rdsWrapper = host.Services.GetRequiredService<RDSWrapper>();

       Console.WriteLine(sepBar);
       Console.WriteLine(
           "Welcome to the Amazon Relational Database Service (Amazon RDS) DB
instance scenario example.");
       Console.WriteLine(sepBar);
```

```
        try
        {
            var parameterGroupFamily = await ChooseParameterGroupFamily();

            var parameterGroup = await
CreateDbParameterGroup(parameterGroupFamily);

            var parameters = await
DescribeParametersInGroup(parameterGroup.DBParameterGroupName,
                new List<string> { "auto_increment_offset",
"auto_increment_increment" });

            await ModifyParameters(parameterGroup.DBParameterGroupName,
parameters);

            await
DescribeUserSourceParameters(parameterGroup.DBParameterGroupName);

            var engineVersionChoice = await
ChooseDbEngineVersion(parameterGroupFamily);

            var instanceChoice = await ChooseDbInstanceClass(engine,
engineVersionChoice.EngineVersion);

            var newInstanceIdentifier = "Example-Instance-" + DateTime.Now.Ticks;

            var newInstance = await CreateRdsNewInstance(parameterGroup, engine,
engineVersionChoice.EngineVersion,
                instanceChoice.DBInstanceClass, newInstanceIdentifier);
            if (newInstance != null)
            {
                DisplayConnectionString(newInstance);

                await CreateSnapshot(newInstance);

                await DeleteRdsInstance(newInstance);
            }

            await DeleteParameterGroup(parameterGroup);

            Console.WriteLine("Scenario complete.");
            Console.WriteLine(sepBar);
        }
        catch (Exception ex)
```

```
        {
            logger.LogError(ex, "There was a problem executing the scenario.");
        }
    }

    /// <summary>
    /// Choose the RDS DB parameter group family from a list of available
options.
    /// </summary>
    /// <returns>The selected parameter group family.</returns>
    public static async Task<string> ChooseParameterGroupFamily()
    {
        Console.WriteLine(sepBar);
        // 1. Get a list of available engines.
        var engines = await rdsWrapper.DescribeDBEngineVersions(engine);

        Console.WriteLine("1. The following is a list of available DB parameter
group families:");
        int i = 1;
        var parameterGroupFamilies = engines.GroupBy(e =>
e.DBParameterGroupFamily).ToList();
        foreach (var parameterGroupFamily in parameterGroupFamilies)
        {
            // List the available parameter group families.
            Console.WriteLine(
                $"\t{i}. Family: {parameterGroupFamily.Key}");
            i++;
        }

        var choiceNumber = 0;
        while (choiceNumber < 1 || choiceNumber > parameterGroupFamilies.Count)
        {
            Console.WriteLine("Select an available DB parameter group family by
entering a number from the list above:");
            var choice = Console.ReadLine();
            Int32.TryParse(choice, out choiceNumber);
        }
        var parameterGroupFamilyChoice = parameterGroupFamilies[choiceNumber -
1];
        Console.WriteLine(sepBar);
        return parameterGroupFamilyChoice.Key;
    }

    /// <summary>
```

```
   /// Create and get information on a DB parameter group.
   /// </summary>
   /// <param name="dbParameterGroupFamily">The DBParameterGroupFamily for the
new DB parameter group.</param>
   /// <returns>The new DBParameterGroup.</returns>
   public static async Task<DBParameterGroup> CreateDbParameterGroup(string
dbParameterGroupFamily)
   {
       Console.WriteLine(sepBar);
       Console.WriteLine($"2. Create new DB parameter group with family
{dbParameterGroupFamily}:");

       var parameterGroup = await rdsWrapper.CreateDBParameterGroup(
           "ExampleParameterGroup-" + DateTime.Now.Ticks,
           dbParameterGroupFamily, "New example parameter group");

       var groupInfo =
           await rdsWrapper.DescribeDBParameterGroups(parameterGroup
               .DBParameterGroupName);

       Console.WriteLine(
           $"3. New DB parameter group: \n\t{groupInfo[0].Description}, \n\tARN
{groupInfo[0].DBParameterGroupArn}");
       Console.WriteLine(sepBar);
       return parameterGroup;
   }

   /// <summary>
   /// Get and describe parameters from a DBParameterGroup.
   /// </summary>
   /// <param name="parameterGroupName">Name of the DBParameterGroup.</param>
   /// <param name="parameterNames">Optional specific names of parameters to
describe.</param>
   /// <returns>The list of requested parameters.</returns>
   public static async Task<List<Parameter>> DescribeParametersInGroup(string
parameterGroupName, List<string>? parameterNames = null)
   {
       Console.WriteLine(sepBar);
       Console.WriteLine("4. Get some parameters from the group.");
       Console.WriteLine(sepBar);

       var parameters =
           await rdsWrapper.DescribeDBParameters(parameterGroupName);
```

```
        var matchingParameters =
            parameters.Where(p => parameterNames == null ||
parameterNames.Contains(p.ParameterName)).ToList();

        Console.WriteLine("5. Parameter information:");
        matchingParameters.ForEach(p =>
            Console.WriteLine(
                $"\n\tParameter: {p.ParameterName}." +
                $"\n\tDescription: {p.Description}." +
                $"\n\tAllowed Values: {p.AllowedValues}." +
                $"\n\tValue: {p.ParameterValue}."));

        Console.WriteLine(sepBar);

        return matchingParameters;
    }

    /// <summary>
    /// Modify a parameter from a DBParameterGroup.
    /// </summary>
    /// <param name="parameterGroupName">Name of the DBParameterGroup.</param>
    /// <param name="parameters">The parameters to modify.</param>
    /// <returns>Async task.</returns>
    public static async Task ModifyParameters(string parameterGroupName,
List<Parameter> parameters)
    {
        Console.WriteLine(sepBar);
        Console.WriteLine("6. Modify some parameters in the group.");

        foreach (var p in parameters)
        {
            if (p.IsModifiable && p.DataType == "integer")
            {
                int newValue = 0;
                while (newValue == 0)
                {
                    Console.WriteLine(
                        $"Enter a new value for {p.ParameterName} from the
allowed values {p.AllowedValues} ");

                    var choice = Console.ReadLine();
                    Int32.TryParse(choice, out newValue);
                }
```

```
                p.ParameterValue = newValue.ToString();
            }
        }

        await rdsWrapper.ModifyDBParameterGroup(parameterGroupName, parameters);

        Console.WriteLine(sepBar);
    }

    /// <summary>
    /// Describe the user source parameters in the group.
    /// </summary>
    /// <param name="parameterGroupName">Name of the DBParameterGroup.</param>
    /// <returns>Async task.</returns>
    public static async Task DescribeUserSourceParameters(string
parameterGroupName)
    {
        Console.WriteLine(sepBar);
        Console.WriteLine("7. Describe user source parameters in the group.");

        var parameters =
            await rdsWrapper.DescribeDBParameters(parameterGroupName, "user");


        parameters.ForEach(p =>
            Console.WriteLine(
                $"\n\tParameter: {p.ParameterName}." +
                $"\n\tDescription: {p.Description}." +
                $"\n\tAllowed Values: {p.AllowedValues}." +
                $"\n\tValue: {p.ParameterValue}."));

        Console.WriteLine(sepBar);
    }


    /// <summary>
    /// Choose a DB engine version.
    /// </summary>
    /// <param name="dbParameterGroupFamily">DB parameter group family for engine
choice.</param>
    /// <returns>The selected engine version.</returns>
    public static async Task<DBEngineVersion> ChooseDbEngineVersion(string
dbParameterGroupFamily)
    {
```

```
        Console.WriteLine(sepBar);
        // Get a list of allowed engines.
        var allowedEngines =
            await rdsWrapper.DescribeDBEngineVersions(engine,
 dbParameterGroupFamily);

        Console.WriteLine($"Available DB engine versions for parameter group
 family {dbParameterGroupFamily}:");
        int i = 1;
        foreach (var version in allowedEngines)
        {
            Console.WriteLine(
                $"\t{i}. Engine: {version.Engine} Version
 {version.EngineVersion}.");
            i++;
        }

        var choiceNumber = 0;
        while (choiceNumber < 1 || choiceNumber > allowedEngines.Count)
        {
            Console.WriteLine("8. Select an available DB engine version by
 entering a number from the list above:");
            var choice = Console.ReadLine();
            Int32.TryParse(choice, out choiceNumber);
        }

        var engineChoice = allowedEngines[choiceNumber - 1];
        Console.WriteLine(sepBar);
        return engineChoice;
    }

    /// <summary>
    /// Choose a DB instance class for a particular engine and engine version.
    /// </summary>
    /// <param name="engine">DB engine for DB instance choice.</param>
    /// <param name="engineVersion">DB engine version for DB instance choice.</
param>
    /// <returns>The selected orderable DB instance option.</returns>
    public static async Task<OrderableDBInstanceOption>
 ChooseDbInstanceClass(string engine, string engineVersion)
    {
        Console.WriteLine(sepBar);
        // Get a list of allowed DB instance classes.
        var allowedInstances =
```

```
            await rdsWrapper.DescribeOrderableDBInstanceOptions(engine,
 engineVersion);

        Console.WriteLine($"8. Available micro DB instance classes for engine
 {engine} and version {engineVersion}:");
        int i = 1;

        // Filter to micro instances for this example.
        allowedInstances = allowedInstances
            .Where(i => i.DBInstanceClass.Contains("micro")).ToList();

        foreach (var instance in allowedInstances)
        {
            Console.WriteLine(
                $"\t{i}. Instance class: {instance.DBInstanceClass} (storage type
 {instance.StorageType})");
            i++;
        }

        var choiceNumber = 0;
        while (choiceNumber < 1 || choiceNumber > allowedInstances.Count)
        {
            Console.WriteLine("9. Select an available DB instance class by
 entering a number from the list above:");
            var choice = Console.ReadLine();
            Int32.TryParse(choice, out choiceNumber);
        }

        var instanceChoice = allowedInstances[choiceNumber - 1];
        Console.WriteLine(sepBar);
        return instanceChoice;
    }

    /// <summary>
    /// Create a new RDS DB instance.
    /// </summary>
    /// <param name="parameterGroup">Parameter group to use for the DB
 instance.</param>
    /// <param name="engineName">Engine to use for the DB instance.</param>
    /// <param name="engineVersion">Engine version to use for the DB instance.</
param>
    /// <param name="instanceClass">Instance class to use for the DB instance.</
param>
```

```
   /// <param name="instanceIdentifier">Instance identifier to use for the DB
instance.</param>
   /// <returns>The new DB instance.</returns>
   public static async Task<DBInstance?> CreateRdsNewInstance(DBParameterGroup
parameterGroup,
       string engineName, string engineVersion, string instanceClass, string
instanceIdentifier)
   {
       Console.WriteLine(sepBar);
       Console.WriteLine($"10. Create a new DB instance with identifier
{instanceIdentifier}.");
       bool isInstanceReady = false;
       DBInstance newInstance;
       var instances = await rdsWrapper.DescribeDBInstances();
       isInstanceReady = instances.FirstOrDefault(i =>
           i.DBInstanceIdentifier == instanceIdentifier)?.DBInstanceStatus ==
"available";

       if (isInstanceReady)
       {
           Console.WriteLine("Instance already created.");
           newInstance = instances.First(i => i.DBInstanceIdentifier ==
instanceIdentifier);
       }
       else
       {
           Console.WriteLine("Please enter an admin user name:");
           var username = Console.ReadLine();

           Console.WriteLine("Please enter an admin password:");
           var password = Console.ReadLine();

           newInstance = await rdsWrapper.CreateDBInstance(
               "ExampleInstance",
               instanceIdentifier,
               parameterGroup.DBParameterGroupName,
               engineName,
               engineVersion,
               instanceClass,
               20,
               username,
               password
           );
```

```
            // 11. Wait for the DB instance to be ready.

            Console.WriteLine("11. Waiting for DB instance to be ready...");
            while (!isInstanceReady)
            {
                instances = await
 rdsWrapper.DescribeDBInstances(instanceIdentifier);
                isInstanceReady = instances.FirstOrDefault()?.DBInstanceStatus ==
 "available";
                newInstance = instances.First();
                Thread.Sleep(30000);
            }
        }

        Console.WriteLine(sepBar);
        return newInstance;
    }

    /// <summary>
    /// Display a connection string for an RDS DB instance.
    /// </summary>
    /// <param name="instance">The DB instance to use to get a connection
 string.</param>
    public static void DisplayConnectionString(DBInstance instance)
    {
        Console.WriteLine(sepBar);
        // Display the connection string.
        Console.WriteLine("12. New DB instance connection string: ");
        Console.WriteLine(
            $"\n{engine} -h {instance.Endpoint.Address} -P
 {instance.Endpoint.Port} "
            + $"-u {instance.MasterUsername} -p [YOUR PASSWORD]\n");

        Console.WriteLine(sepBar);
    }

    /// <summary>
    /// Create a snapshot from an RDS DB instance.
    /// </summary>
    /// <param name="instance">DB instance to use when creating a snapshot.</
param>
    /// <returns>The snapshot object.</returns>
    public static async Task<DBSnapshot> CreateSnapshot(DBInstance instance)
    {
```

```
        Console.WriteLine(sepBar);
        // Create a snapshot.
        Console.WriteLine($"13. Creating snapshot from DB instance
{instance.DBInstanceIdentifier}.");
        var snapshot = await
rdsWrapper.CreateDBSnapshot(instance.DBInstanceIdentifier, "ExampleSnapshot-" +
DateTime.Now.Ticks);

        // Wait for the snapshot to be available
        bool isSnapshotReady = false;

        Console.WriteLine($"14. Waiting for snapshot to be ready...");
        while (!isSnapshotReady)
        {
            var snapshots = await
rdsWrapper.DescribeDBSnapshots(instance.DBInstanceIdentifier);
            isSnapshotReady = snapshots.FirstOrDefault()?.Status == "available";
            snapshot = snapshots.First();
            Thread.Sleep(30000);
        }

        Console.WriteLine(
            $"Snapshot {snapshot.DBSnapshotIdentifier} status is
{snapshot.Status}.");
        Console.WriteLine(sepBar);
        return snapshot;
    }

    /// <summary>
    /// Delete an RDS DB instance.
    /// </summary>
    /// <param name="instance">The DB instance to delete.</param>
    /// <returns>Async task.</returns>
    public static async Task DeleteRdsInstance(DBInstance newInstance)
    {
        Console.WriteLine(sepBar);
        // Delete the DB instance.
        Console.WriteLine($"15. Delete the DB instance
{newInstance.DBInstanceIdentifier}.");
        await rdsWrapper.DeleteDBInstance(newInstance.DBInstanceIdentifier);

        // Wait for the DB instance to delete.
        Console.WriteLine($"16. Waiting for the DB instance to delete...");
        bool isInstanceDeleted = false;
```

```
        while (!isInstanceDeleted)
        {
            var instance = await rdsWrapper.DescribeDBInstances();
            isInstanceDeleted = instance.All(i => i.DBInstanceIdentifier !=
 newInstance.DBInstanceIdentifier);
            Thread.Sleep(30000);
        }

        Console.WriteLine("DB instance deleted.");
        Console.WriteLine(sepBar);
    }

    /// <summary>
    /// Delete a DB parameter group.
    /// </summary>
    /// <param name="parameterGroup">The parameter group to delete.</param>
    /// <returns>Async task.</returns>
    public static async Task DeleteParameterGroup(DBParameterGroup
 parameterGroup)
    {
        Console.WriteLine(sepBar);
        // Delete the parameter group.
        Console.WriteLine($"17. Delete the DB parameter group
 {parameterGroup.DBParameterGroupName}.");
        await
 rdsWrapper.DeleteDBParameterGroup(parameterGroup.DBParameterGroupName);

        Console.WriteLine(sepBar);
    }
```

Metodi wrapper utilizzati dallo scenario per operazioni delle istanze DB.

```
/// <summary>
/// Wrapper methods to use Amazon Relational Database Service (Amazon RDS) with
 DB instance operations.
/// </summary>
public partial class RDSWrapper
{
    private readonly IAmazonRDS _amazonRDS;
    public RDSWrapper(IAmazonRDS amazonRDS)
```

```
    {
        _amazonRDS = amazonRDS;
    }


    /// <summary>
    /// Get a list of DB engine versions for a particular DB engine.
    /// </summary>
    /// <param name="engine">Name of the engine.</param>
    /// <param name="dbParameterGroupFamily">Optional parameter group family
name.</param>
    /// <returns>List of DBEngineVersions.</returns>
    public async Task<List<DBEngineVersion>> DescribeDBEngineVersions(string
engine,
        string dbParameterGroupFamily = null)
    {
        var response = await _amazonRDS.DescribeDBEngineVersionsAsync(
            new DescribeDBEngineVersionsRequest()
            {
                Engine = engine,
                DBParameterGroupFamily = dbParameterGroupFamily
            });
        return response.DBEngineVersions;
    }



    /// <summary>
    /// Get a list of orderable DB instance options for a specific
    /// engine and engine version.
    /// </summary>
    /// <param name="engine">Name of the engine.</param>
    /// <param name="engineVersion">Version of the engine.</param>
    /// <returns>List of OrderableDBInstanceOptions.</returns>
    public async Task<List<OrderableDBInstanceOption>>
DescribeOrderableDBInstanceOptions(string engine, string engineVersion)
    {
        // Use a paginator to get a list of DB instance options.
        var results = new List<OrderableDBInstanceOption>();
        var paginateInstanceOptions =
_amazonRDS.Paginators.DescribeOrderableDBInstanceOptions(
            new DescribeOrderableDBInstanceOptionsRequest()
            {
                Engine = engine,
```

```
                EngineVersion = engineVersion,
            });
        // Get the entire list using the paginator.
        await foreach (var instanceOptions in
paginateInstanceOptions.OrderableDBInstanceOptions)
        {
            results.Add(instanceOptions);
        }
        return results;
    }



    /// <summary>
    /// Returns a list of DB instances.
    /// </summary>
    /// <param name="dbInstanceIdentifier">Optional name of a specific DB
instance.</param>
    /// <returns>List of DB instances.</returns>
    public async Task<List<DBInstance>> DescribeDBInstances(string
dbInstanceIdentifier = null)
    {
        var results = new List<DBInstance>();
        var instancesPaginator = _amazonRDS.Paginators.DescribeDBInstances(
            new DescribeDBInstancesRequest
            {
                DBInstanceIdentifier = dbInstanceIdentifier
            });
        // Get the entire list using the paginator.
        await foreach (var instances in instancesPaginator.DBInstances)
        {
            results.Add(instances);
        }
        return results;
    }



    /// <summary>
    /// Create an RDS DB instance with a particular set of properties. Use the
action DescribeDBInstancesAsync
    /// to determine when the DB instance is ready to use.
    /// </summary>
    /// <param name="dbName">Name for the DB instance.</param>
```

```
   /// <param name="dbInstanceIdentifier">DB instance identifier.</param>
   /// <param name="parameterGroupName">DB parameter group to associate with the
instance.</param>
   /// <param name="dbEngine">The engine for the DB instance.</param>
   /// <param name="dbEngineVersion">Version for the DB instance.</param>
   /// <param name="instanceClass">Class for the DB instance.</param>
   /// <param name="allocatedStorage">The amount of storage in gibibytes (GiB)
to allocate to the DB instance.</param>
   /// <param name="adminName">Admin user name.</param>
   /// <param name="adminPassword">Admin user password.</param>
   /// <returns>DB instance object.</returns>
   public async Task<DBInstance> CreateDBInstance(string dbName, string
dbInstanceIdentifier,
       string parameterGroupName, string dbEngine, string dbEngineVersion,
       string instanceClass, int allocatedStorage, string adminName, string
adminPassword)
   {
       var response = await _amazonRDS.CreateDBInstanceAsync(
           new CreateDBInstanceRequest()
           {
               DBName = dbName,
               DBInstanceIdentifier = dbInstanceIdentifier,
               DBParameterGroupName = parameterGroupName,
               Engine = dbEngine,
               EngineVersion = dbEngineVersion,
               DBInstanceClass = instanceClass,
               AllocatedStorage = allocatedStorage,
               MasterUsername = adminName,
               MasterUserPassword = adminPassword
           });

       return response.DBInstance;
   }



   /// <summary>
   /// Delete a particular DB instance.
   /// </summary>
   /// <param name="dbInstanceIdentifier">DB instance identifier.</param>
   /// <returns>DB instance object.</returns>
   public async Task<DBInstance> DeleteDBInstance(string dbInstanceIdentifier)
   {
       var response = await _amazonRDS.DeleteDBInstanceAsync(
```

```
            new DeleteDBInstanceRequest()
            {
                DBInstanceIdentifier = dbInstanceIdentifier,
                SkipFinalSnapshot = true,
                DeleteAutomatedBackups = true
            });

        return response.DBInstance;
    }
```

Metodi wrapper utilizzati dallo scenario per gruppi di parametri database.

```
/// <summary>
/// Wrapper methods to use Amazon Relational Database Service (Amazon RDS) with
 parameter groups.
/// </summary>
public partial class RDSWrapper
{

    /// <summary>
    /// Get descriptions of DB parameter groups.
    /// </summary>
    /// <param name="name">Optional name of the DB parameter group to describe.</
param>
    /// <returns>The list of DB parameter group descriptions.</returns>
    public async Task<List<DBParameterGroup>> DescribeDBParameterGroups(string
 name = null)
    {
        var response = await _amazonRDS.DescribeDBParameterGroupsAsync(
            new DescribeDBParameterGroupsRequest()
            {
                DBParameterGroupName = name
            });
        return response.DBParameterGroups;
    }



    /// <summary>
```

```
    /// Create a new DB parameter group. Use the action
 DescribeDBParameterGroupsAsync
    /// to determine when the DB parameter group is ready to use.
    /// </summary>
    /// <param name="name">Name of the DB parameter group.</param>
    /// <param name="family">Family of the DB parameter group.</param>
    /// <param name="description">Description of the DB parameter group.</param>
    /// <returns>The new DB parameter group.</returns>
    public async Task<DBParameterGroup> CreateDBParameterGroup(
        string name, string family, string description)
    {
        var response = await _amazonRDS.CreateDBParameterGroupAsync(
            new CreateDBParameterGroupRequest()
            {
                DBParameterGroupName = name,
                DBParameterGroupFamily = family,
                Description = description
            });
        return response.DBParameterGroup;
    }



    /// <summary>
    /// Update a DB parameter group. Use the action
 DescribeDBParameterGroupsAsync
    /// to determine when the DB parameter group is ready to use.
    /// </summary>
    /// <param name="name">Name of the DB parameter group.</param>
    /// <param name="parameters">List of parameters. Maximum of 20 per request.</
param>
    /// <returns>The updated DB parameter group name.</returns>
    public async Task<string> ModifyDBParameterGroup(
        string name, List<Parameter> parameters)
    {
        var response = await _amazonRDS.ModifyDBParameterGroupAsync(
            new ModifyDBParameterGroupRequest()
            {
                DBParameterGroupName = name,
                Parameters = parameters,
            });
        return response.DBParameterGroupName;
    }
```

```
    /// <summary>
    /// Delete a DB parameter group. The group cannot be a default DB parameter
group
    /// or be associated with any DB instances.
    /// </summary>
    /// <param name="name">Name of the DB parameter group.</param>
    /// <returns>True if successful.</returns>
    public async Task<bool> DeleteDBParameterGroup(string name)
    {
        var response = await _amazonRDS.DeleteDBParameterGroupAsync(
            new DeleteDBParameterGroupRequest()
            {
                DBParameterGroupName = name,
            });
        return response.HttpStatusCode == HttpStatusCode.OK;
    }



    /// <summary>
    /// Get a list of DB parameters from a specific parameter group.
    /// </summary>
    /// <param name="dbParameterGroupName">Name of a specific DB parameter
group.</param>
    /// <param name="source">Optional source for selecting parameters.</param>
    /// <returns>List of parameter values.</returns>
    public async Task<List<Parameter>> DescribeDBParameters(string
dbParameterGroupName, string source = null)
    {
        var results = new List<Parameter>();
        var paginateParameters = _amazonRDS.Paginators.DescribeDBParameters(
            new DescribeDBParametersRequest()
            {
                DBParameterGroupName = dbParameterGroupName,
                Source = source
            });
        // Get the entire list using the paginator.
        await foreach (var parameters in paginateParameters.Parameters)
        {
            results.Add(parameters);
        }
        return results;
```

```
    }
```

Metodi wrapper utilizzati dallo scenario per operazioni degli snapshot database.

```
/// <summary>
/// Wrapper methods to use Amazon Relational Database Service (Amazon RDS) with
 snapshots.
/// </summary>
public partial class RDSWrapper
{

    /// <summary>
    /// Create a snapshot of a DB instance.
    /// </summary>
    /// <param name="dbInstanceIdentifier">DB instance identifier.</param>
    /// <param name="snapshotIdentifier">Identifier for the snapshot.</param>
    /// <returns>DB snapshot object.</returns>
    public async Task<DBSnapshot> CreateDBSnapshot(string dbInstanceIdentifier,
 string snapshotIdentifier)
    {
        var response = await _amazonRDS.CreateDBSnapshotAsync(
            new CreateDBSnapshotRequest()
            {
                DBSnapshotIdentifier = snapshotIdentifier,
                DBInstanceIdentifier = dbInstanceIdentifier
            });

        return response.DBSnapshot;
    }



    /// <summary>
    /// Return a list of DB snapshots for a particular DB instance.
    /// </summary>
    /// <param name="dbInstanceIdentifier">DB instance identifier.</param>
    /// <returns>List of DB snapshots.</returns>
    public async Task<List<DBSnapshot>> DescribeDBSnapshots(string
 dbInstanceIdentifier)
    {
```

```
        var results = new List<DBSnapshot>();
        var snapshotsPaginator = _amazonRDS.Paginators.DescribeDBSnapshots(
            new DescribeDBSnapshotsRequest()
            {
                DBInstanceIdentifier = dbInstanceIdentifier
            });

        // Get the entire list using the paginator.
        await foreach (var snapshots in snapshotsPaginator.DBSnapshots)
        {
            results.Add(snapshots);
        }
        return results;
    }
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella Documentazione di riferimento delle API AWS SDK for .NET .
  - CreateDBInstance
  - Creato B ParameterGroup
  - CreateDBSnapshot
  - DeleteDBInstance
  - Eliminare DB ParameterGroup
  - Descritto B EngineVersions
  - DescribeDBInstances
  - Descritto B ParameterGroups
  - DescribeDBParameters
  - DescribeDBSnapshots
  - DescribeOrderableDB InstanceOptions
  - Modifica DB ParameterGroup

C++

SDK per C++

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
        Aws::Client::ClientConfiguration clientConfig;
        // Optional: Set to the AWS Region (overrides config file).
        // clientConfig.region = "us-east-1";

//! Routine which creates an Amazon RDS instance and demonstrates several
 operations
//! on that instance.
/*!
 \sa gettingStartedWithDBInstances()
 \param clientConfiguration: AWS client configuration.
 \return bool: Successful completion.
 */
bool AwsDoc::RDS::gettingStartedWithDBInstances(
        const Aws::Client::ClientConfiguration &clientConfig) {
    Aws::RDS::RDSClient client(clientConfig);

    printAsterisksLine();
    std::cout << "Welcome to the Amazon Relational Database Service (Amazon RDS)"
              << std::endl;
    std::cout << "get started with DB instances demo." << std::endl;
    printAsterisksLine();

    std::cout << "Checking for an existing DB parameter group named '" <<
              PARAMETER_GROUP_NAME << "'." << std::endl;
    Aws::String dbParameterGroupFamily("Undefined");
    bool parameterGroupFound = true;
    {
        // 1. Check if the DB parameter group already exists.
        Aws::RDS::Model::DescribeDBParameterGroupsRequest request;
        request.SetDBParameterGroupName(PARAMETER_GROUP_NAME);

        Aws::RDS::Model::DescribeDBParameterGroupsOutcome outcome =
```

```
                client.DescribeDBParameterGroups(request);

        if (outcome.IsSuccess()) {
            std::cout << "DB parameter group named '" <<
                      PARAMETER_GROUP_NAME << "' already exists." << std::endl;
            dbParameterGroupFamily = outcome.GetResult().GetDBParameterGroups()
[0].GetDBParameterGroupFamily();
        }
        else if (outcome.GetError().GetErrorType() ==
                 Aws::RDS::RDSErrors::D_B_PARAMETER_GROUP_NOT_FOUND_FAULT) {
            std::cout << "DB parameter group named '" <<
                      PARAMETER_GROUP_NAME << "' does not exist." << std::endl;
            parameterGroupFound = false;
        }
        else {
            std::cerr << "Error with RDS::DescribeDBParameterGroups. "
                      << outcome.GetError().GetMessage()
                      << std::endl;
            return false;
        }
    }

    if (!parameterGroupFound) {
        Aws::Vector<Aws::RDS::Model::DBEngineVersion> engineVersions;

        // 2. Get available engine versions for the specified engine.
        if (!getDBEngineVersions(DB_ENGINE, NO_PARAMETER_GROUP_FAMILY,
                                 engineVersions, client)) {
            return false;
        }

        std::cout << "Getting available database engine versions for " <<
 DB_ENGINE
                  << "."
                  << std::endl;
        std::vector<Aws::String> families;
        for (const Aws::RDS::Model::DBEngineVersion &version: engineVersions) {
            Aws::String family = version.GetDBParameterGroupFamily();
            if (std::find(families.begin(), families.end(), family) ==
                families.end()) {
                families.push_back(family);
                std::cout << "  " << families.size() << ": " << family <<
 std::endl;
            }
```

```
        }

        int choice = askQuestionForIntRange("Which family do you want to use? ",
1,
                                            static_cast<int>(families.size()));
        dbParameterGroupFamily = families[choice - 1];
    }
    if (!parameterGroupFound) {
        // 3.  Create a DB parameter group.
        Aws::RDS::Model::CreateDBParameterGroupRequest request;
        request.SetDBParameterGroupName(PARAMETER_GROUP_NAME);
        request.SetDBParameterGroupFamily(dbParameterGroupFamily);
        request.SetDescription("Example parameter group.");

        Aws::RDS::Model::CreateDBParameterGroupOutcome outcome =
                client.CreateDBParameterGroup(request);

        if (outcome.IsSuccess()) {
            std::cout << "The DB parameter group was successfully created."
                      << std::endl;
        }
        else {
            std::cerr << "Error with RDS::CreateDBParameterGroup. "
                      << outcome.GetError().GetMessage()
                      << std::endl;
            return false;
        }
    }

    printAsterisksLine();
    std::cout << "Let's set some parameter values in your parameter group."
              << std::endl;

    Aws::String marker;
    Aws::Vector<Aws::RDS::Model::Parameter> autoIncrementParameters;
    // 4.  Get the parameters in the DB parameter group.
    if (!getDBParameters(PARAMETER_GROUP_NAME, AUTO_INCREMENT_PREFIX, NO_SOURCE,
                         autoIncrementParameters,
                         client)) {
        cleanUpResources(PARAMETER_GROUP_NAME, "", client);
        return false;
    }

    Aws::Vector<Aws::RDS::Model::Parameter> updateParameters;
```

```
    for (Aws::RDS::Model::Parameter &autoIncParameter: autoIncrementParameters) {
        if (autoIncParameter.GetIsModifiable() &&
            (autoIncParameter.GetDataType() == "integer")) {
            std::cout << "The " << autoIncParameter.GetParameterName()
                      << " is described as: " <<
                      autoIncParameter.GetDescription() << "." << std::endl;
            if (autoIncParameter.ParameterValueHasBeenSet()) {
                std::cout << "The current value is "
                          << autoIncParameter.GetParameterValue()
                          << "." << std::endl;
            }
            std::vector<int> splitValues = splitToInts(
                    autoIncParameter.GetAllowedValues(), '-');
            if (splitValues.size() == 2) {
                int newValue = askQuestionForIntRange(
                        Aws::String("Enter a new value in the range ") +
                        autoIncParameter.GetAllowedValues() + ": ",
                        splitValues[0], splitValues[1]);
                autoIncParameter.SetParameterValue(std::to_string(newValue));
                updateParameters.push_back(autoIncParameter);

            }
            else {
                std::cerr << "Error parsing " <<
autoIncParameter.GetAllowedValues()
                          << std::endl;
            }
        }
    }

    {
        // 5.  Modify the auto increment parameters in the group.
        Aws::RDS::Model::ModifyDBParameterGroupRequest request;
        request.SetDBParameterGroupName(PARAMETER_GROUP_NAME);
        request.SetParameters(updateParameters);

        Aws::RDS::Model::ModifyDBParameterGroupOutcome outcome =
                client.ModifyDBParameterGroup(request);

        if (outcome.IsSuccess()) {
            std::cout << "The DB parameter group was successfully modified."
                      << std::endl;
        }
```

```
        else {
            std::cerr << "Error with RDS::ModifyDBParameterGroup. "
                      << outcome.GetError().GetMessage()
                      << std::endl;
        }
    }

    std::cout
            << "You can get a list of parameters you've set by specifying a
source of 'user'."
            << std::endl;

    Aws::Vector<Aws::RDS::Model::Parameter> userParameters;
    // 6.  Display the modified parameters in the group.
    if (!getDBParameters(PARAMETER_GROUP_NAME, NO_NAME_PREFIX, "user",
userParameters,
                         client)) {
        cleanUpResources(PARAMETER_GROUP_NAME, "", client);
        return false;
    }

    for (const auto &userParameter: userParameters) {
        std::cout << "  " << userParameter.GetParameterName() << ", " <<
                  userParameter.GetDescription() << ", parameter value - "
                  << userParameter.GetParameterValue() << std::endl;
    }

    printAsterisksLine();
    std::cout << "Checking for an existing DB instance." << std::endl;

    Aws::RDS::Model::DBInstance dbInstance;
    // 7.  Check if the DB instance already exists.
    if (!describeDBInstance(DB_INSTANCE_IDENTIFIER, dbInstance, client)) {
        cleanUpResources(PARAMETER_GROUP_NAME, "", client);
        return false;
    }

    if (dbInstance.DbInstancePortHasBeenSet()) {
        std::cout << "The DB instance already exists." << std::endl;
    }
    else {
        std::cout << "Let's create a DB instance." << std::endl;
        const Aws::String administratorName = askQuestion(
                "Enter an administrator username for the database: ");
```

```
        const Aws::String administratorPassword = askQuestion(
                "Enter a password for the administrator (at least 8 characters):
");
        Aws::Vector<Aws::RDS::Model::DBEngineVersion> engineVersions;

        // 8.  Get a list of available engine versions.
        if (!getDBEngineVersions(DB_ENGINE, dbParameterGroupFamily,
engineVersions,
                                 client)) {
            cleanUpResources(PARAMETER_GROUP_NAME, "", client);
            return false;
        }

        std::cout << "The available engines for your parameter group are:" <<
std::endl;

        int index = 1;
        for (const Aws::RDS::Model::DBEngineVersion &engineVersion:
engineVersions) {
            std::cout << "  " << index << ": " <<
engineVersion.GetEngineVersion()
                      << std::endl;
            ++index;
        }
        int choice = askQuestionForIntRange("Which engine do you want to use? ",
1,

static_cast<int>(engineVersions.size()));
        const Aws::RDS::Model::DBEngineVersion engineVersion =
engineVersions[choice -
                                                                              1];

        Aws::String dbInstanceClass;
        // 9.  Get a list of micro instance classes.
        if (!chooseMicroDBInstanceClass(engineVersion.GetEngine(),
                                        engineVersion.GetEngineVersion(),
                                        dbInstanceClass,
                                        client)) {
            cleanUpResources(PARAMETER_GROUP_NAME, "", client);
            return false;
        }

        std::cout << "Creating a DB instance named '" << DB_INSTANCE_IDENTIFIER
                  << "' and database '" << DB_NAME << "'.\n"
```

```
                << "The DB instance is configured to use your custom parameter
group '"
                << PARAMETER_GROUP_NAME << "',\n"
                << "selected engine version " <<
engineVersion.GetEngineVersion()
                << ",\n"
                << "selected DB instance class '" << dbInstanceClass << "',"
                << " and " << DB_ALLOCATED_STORAGE << " GiB of " <<
DB_STORAGE_TYPE
                << " storage.\nThis typically takes several minutes." <<
std::endl;

       Aws::RDS::Model::CreateDBInstanceRequest request;
       request.SetDBName(DB_NAME);
       request.SetDBInstanceIdentifier(DB_INSTANCE_IDENTIFIER);
       request.SetDBParameterGroupName(PARAMETER_GROUP_NAME);
       request.SetEngine(engineVersion.GetEngine());
       request.SetEngineVersion(engineVersion.GetEngineVersion());
       request.SetDBInstanceClass(dbInstanceClass);
       request.SetStorageType(DB_STORAGE_TYPE);
       request.SetAllocatedStorage(DB_ALLOCATED_STORAGE);
       request.SetMasterUsername(administratorName);
       request.SetMasterUserPassword(administratorPassword);

       Aws::RDS::Model::CreateDBInstanceOutcome outcome =
               client.CreateDBInstance(request);

       if (outcome.IsSuccess()) {
           std::cout << "The DB instance creation has started."
                     << std::endl;
       }
       else {
           std::cerr << "Error with RDS::CreateDBInstance. "
                     << outcome.GetError().GetMessage()
                     << std::endl;
           cleanUpResources(PARAMETER_GROUP_NAME, "", client);
           return false;
       }
   }

   std::cout << "Waiting for the DB instance to become available." << std::endl;

   int counter = 0;
   // 11. Wait for the DB instance to become available.
```

```
   do {
       std::this_thread::sleep_for(std::chrono::seconds(1));
       ++counter;
       if (counter > 900) {
           std::cerr << "Wait for instance to become available timed out ofter "
                     << counter
                     << " seconds." << std::endl;
           cleanUpResources(PARAMETER_GROUP_NAME, DB_INSTANCE_IDENTIFIER,
client);
           return false;
       }

       dbInstance = Aws::RDS::Model::DBInstance();
       if (!describeDBInstance(DB_INSTANCE_IDENTIFIER, dbInstance, client)) {
           cleanUpResources(PARAMETER_GROUP_NAME, DB_INSTANCE_IDENTIFIER,
client);
           return false;
       }

       if ((counter % 20) == 0) {
           std::cout << "Current DB instance status is '"
                     << dbInstance.GetDBInstanceStatus()
                     << "' after " << counter << " seconds." << std::endl;
       }
   } while (dbInstance.GetDBInstanceStatus() != "available");

   if (dbInstance.GetDBInstanceStatus() == "available") {
       std::cout << "The DB instance has been created." << std::endl;
   }

   printAsterisksLine();

   // 12. Display the connection string that can be used to connect a 'mysql'
shell to the database.
   displayConnection(dbInstance);

   printAsterisksLine();

   if (askYesNoQuestion(
           "Do you want to create a snapshot of your DB instance (y/n)? ")) {
       Aws::String snapshotID(DB_INSTANCE_IDENTIFIER + "-" +
                              Aws::String(Aws::Utils::UUID::RandomUUID()));
       {
```

```
            std::cout << "Creating a snapshot named " << snapshotID << "." <<
std::endl;
            std::cout << "This typically takes a few minutes." << std::endl;

            // 13. Create a snapshot of the DB instance.
            Aws::RDS::Model::CreateDBSnapshotRequest request;
            request.SetDBInstanceIdentifier(DB_INSTANCE_IDENTIFIER);
            request.SetDBSnapshotIdentifier(snapshotID);

            Aws::RDS::Model::CreateDBSnapshotOutcome outcome =
                    client.CreateDBSnapshot(request);

            if (outcome.IsSuccess()) {
                std::cout << "Snapshot creation has started."
                          << std::endl;
            }
            else {
                std::cerr << "Error with RDS::CreateDBSnapshot. "
                          << outcome.GetError().GetMessage()
                          << std::endl;
                cleanUpResources(PARAMETER_GROUP_NAME, DB_INSTANCE_IDENTIFIER,
client);
                return false;
            }
        }

        std::cout << "Waiting for snapshot to become available." << std::endl;

        Aws::RDS::Model::DBSnapshot snapshot;
        counter = 0;
        do {
            std::this_thread::sleep_for(std::chrono::seconds(1));
            ++counter;
            if (counter > 600) {
                std::cerr << "Wait for snapshot to be available timed out ofter "
                          << counter
                          << " seconds." << std::endl;
                cleanUpResources(PARAMETER_GROUP_NAME, DB_INSTANCE_IDENTIFIER,
client);
                return false;
            }

            // 14. Wait for the snapshot to become available.
            Aws::RDS::Model::DescribeDBSnapshotsRequest request;
```

```
            request.SetDBSnapshotIdentifier(snapshotID);

            Aws::RDS::Model::DescribeDBSnapshotsOutcome outcome =
                    client.DescribeDBSnapshots(request);

            if (outcome.IsSuccess()) {
                snapshot = outcome.GetResult().GetDBSnapshots()[0];
            }
            else {
                std::cerr << "Error with RDS::DescribeDBSnapshots. "
                          << outcome.GetError().GetMessage()
                          << std::endl;
                cleanUpResources(PARAMETER_GROUP_NAME, DB_INSTANCE_IDENTIFIER,
 client);
                return false;
            }

            if ((counter % 20) == 0) {
                std::cout << "Current snapshot status is '"
                          << snapshot.GetStatus()
                          << "' after " << counter << " seconds." << std::endl;
            }
        } while (snapshot.GetStatus() != "available");

        if (snapshot.GetStatus() != "available") {
            std::cout << "A snapshot has been created." << std::endl;
        }
    }

    printAsterisksLine();

    bool result = true;
    if (askYesNoQuestion(
            "Do you want to delete the DB instance and parameter group (y/n)? "))
 {
        result = cleanUpResources(PARAMETER_GROUP_NAME, DB_INSTANCE_IDENTIFIER,
 client);
    }

    return result;
}


//! Routine which gets DB parameters using the 'DescribeDBParameters' api.
```

```
/*!
 \sa getDBParameters()
 \param parameterGroupName: The name of the parameter group.
 \param namePrefix: Prefix string to filter results by parameter name.
 \param source: A source such as 'user', ignored if empty.
 \param parametersResult: Vector of 'Parameter' objects returned by the routine.
 \param client: 'RDSClient' instance.
 \return bool: Successful completion.
 */
bool AwsDoc::RDS::getDBParameters(const Aws::String &parameterGroupName,
                                  const Aws::String &namePrefix,
                                  const Aws::String &source,
                                  Aws::Vector<Aws::RDS::Model::Parameter>
 &parametersResult,
                                  const Aws::RDS::RDSClient &client) {
    Aws::String marker;
    do {
        Aws::RDS::Model::DescribeDBParametersRequest request;
        request.SetDBParameterGroupName(PARAMETER_GROUP_NAME);
        if (!marker.empty()) {
            request.SetMarker(marker);
        }
        if (!source.empty()) {
            request.SetSource(source);
        }

        Aws::RDS::Model::DescribeDBParametersOutcome outcome =
                client.DescribeDBParameters(request);

        if (outcome.IsSuccess()) {
            const Aws::Vector<Aws::RDS::Model::Parameter> &parameters =
                    outcome.GetResult().GetParameters();
            for (const Aws::RDS::Model::Parameter &parameter: parameters) {
                if (!namePrefix.empty()) {
                    if (parameter.GetParameterName().find(namePrefix) == 0) {
                        parametersResult.push_back(parameter);
                    }
                }
                else {
                    parametersResult.push_back(parameter);
                }
            }

            marker = outcome.GetResult().GetMarker();
```

```
        }
        else {
            std::cerr << "Error with RDS::DescribeDBParameters. "
                      << outcome.GetError().GetMessage()
                      << std::endl;
            return false;
        }
    } while (!marker.empty());

    return true;
}


//! Routine which gets available DB engine versions for an engine name and
//! an optional parameter group family.
/*!
 \sa getDBEngineVersions()
 \param engineName: A DB engine name.
 \param parameterGroupFamily: A parameter group family name, ignored if empty.
 \param engineVersionsResult: Vector of 'DBEngineVersion' objects returned by the
 routine.
 \param client: 'RDSClient' instance.
 \return bool: Successful completion.
 */
bool AwsDoc::RDS::getDBEngineVersions(const Aws::String &engineName,
                                      const Aws::String &parameterGroupFamily,

 Aws::Vector<Aws::RDS::Model::DBEngineVersion> &engineVersionsResult,
                                      const Aws::RDS::RDSClient &client) {
    Aws::RDS::Model::DescribeDBEngineVersionsRequest request;
    request.SetEngine(engineName);
    if (!parameterGroupFamily.empty()) {
        request.SetDBParameterGroupFamily(parameterGroupFamily);
    }

    engineVersionsResult.clear();
    Aws::String marker; // Used for pagination.

    do {
        if (!marker.empty()) {
            request.SetMarker(marker);
        }
```

```
        Aws::RDS::Model::DescribeDBEngineVersionsOutcome outcome =
                client.DescribeDBEngineVersions(request);

        if (outcome.IsSuccess()) {
            auto &engineVersions = outcome.GetResult().GetDBEngineVersions();
            engineVersionsResult.insert(engineVersionsResult.end(),
 engineVersions.begin(),
                                        engineVersions.end());
            marker = outcome.GetResult().GetMarker();
        }
        else {
            std::cerr << "Error with RDS::DescribeDBEngineVersionsRequest. "
                      << outcome.GetError().GetMessage()
                      << std::endl;
            return false;
        }

    } while (!marker.empty());


    return true;
}


//! Routine which gets a DB instance description.
/*!
 \sa describeDBInstance()
 \param dbInstanceIdentifier: A DB instance identifier.
 \param instanceResult: The 'DBInstance' object containing the description.
 \param client: 'RDSClient' instance.
 \return bool: Successful completion.
 */
bool AwsDoc::RDS::describeDBInstance(const Aws::String &dbInstanceIdentifier,
                                     Aws::RDS::Model::DBInstance &instanceResult,
                                     const Aws::RDS::RDSClient &client) {
    Aws::RDS::Model::DescribeDBInstancesRequest request;
    request.SetDBInstanceIdentifier(dbInstanceIdentifier);

    Aws::RDS::Model::DescribeDBInstancesOutcome outcome =
            client.DescribeDBInstances(request);

    bool result = true;
    if (outcome.IsSuccess()) {
        instanceResult = outcome.GetResult().GetDBInstances()[0];
```

```
    }
    else if (outcome.GetError().GetErrorType() !=
             Aws::RDS::RDSErrors::D_B_INSTANCE_NOT_FOUND_FAULT) {
        result = false;
        std::cerr << "Error with RDS::DescribeDBInstances. "
                  << outcome.GetError().GetMessage()
                  << std::endl;
    }
        // This example does not log an error if the DB instance does not exist.
        // Instead, instanceResult is set to empty.
    else {
        instanceResult = Aws::RDS::Model::DBInstance();
    }

    return result;
}


//! Routine which gets available 'micro' DB instance classes, displays the list
//! to the user, and returns the user selection.
/*!
 \sa chooseMicroDBInstanceClass()
 \param engineName: The DB engine name.
 \param engineVersion: The DB engine version.
 \param dbInstanceClass: String for DB instance class chosen by the user.
 \param client: 'RDSClient' instance.
 \return bool: Successful completion.
 */
bool AwsDoc::RDS::chooseMicroDBInstanceClass(const Aws::String &engine,
                                             const Aws::String &engineVersion,
                                             Aws::String &dbInstanceClass,
                                             const Aws::RDS::RDSClient &client) {
    std::vector<Aws::String> instanceClasses;
    Aws::String marker;
    do {
        Aws::RDS::Model::DescribeOrderableDBInstanceOptionsRequest request;
        request.SetEngine(engine);
        request.SetEngineVersion(engineVersion);
        if (!marker.empty()) {
            request.SetMarker(marker);
        }

        Aws::RDS::Model::DescribeOrderableDBInstanceOptionsOutcome outcome =
                client.DescribeOrderableDBInstanceOptions(request);
```

```
        if (outcome.IsSuccess()) {
            const Aws::Vector<Aws::RDS::Model::OrderableDBInstanceOption>
 &options =
                    outcome.GetResult().GetOrderableDBInstanceOptions();
            for (const Aws::RDS::Model::OrderableDBInstanceOption &option:
 options) {
                const Aws::String &instanceClass = option.GetDBInstanceClass();
                if (instanceClass.find("micro") != std::string::npos) {
                    if (std::find(instanceClasses.begin(), instanceClasses.end(),
                                  instanceClass) ==
                        instanceClasses.end()) {
                        instanceClasses.push_back(instanceClass);
                    }
                }
            }
            marker = outcome.GetResult().GetMarker();
        }
        else {
            std::cerr << "Error with RDS::DescribeOrderableDBInstanceOptions. "
                      << outcome.GetError().GetMessage()
                      << std::endl;
            return false;
        }
    } while (!marker.empty());

    std::cout << "The available micro DB instance classes for your database
 engine are:"
              << std::endl;
    for (int i = 0; i < instanceClasses.size(); ++i) {
        std::cout << "   " << i + 1 << ": " << instanceClasses[i] << std::endl;
    }

    int choice = askQuestionForIntRange(
            "Which micro DB instance class do you want to use? ",
            1, static_cast<int>(instanceClasses.size()));
    dbInstanceClass = instanceClasses[choice - 1];
    return true;
}

//! Routine which deletes resources created by the scenario.
/*!
\sa cleanUpResources()
\param parameterGroupName: A parameter group name, this may be empty.
```

```
\param dbInstanceIdentifier: A DB instance identifier, this may be empty.
\param client: 'RDSClient' instance.
\return bool: Successful completion.
*/
bool AwsDoc::RDS::cleanUpResources(const Aws::String &parameterGroupName,
                                   const Aws::String &dbInstanceIdentifier,
                                   const Aws::RDS::RDSClient &client) {
    bool result = true;
    if (!dbInstanceIdentifier.empty()) {
        {
            // 15. Delete the DB instance.
            Aws::RDS::Model::DeleteDBInstanceRequest request;
            request.SetDBInstanceIdentifier(dbInstanceIdentifier);
            request.SetSkipFinalSnapshot(true);
            request.SetDeleteAutomatedBackups(true);

            Aws::RDS::Model::DeleteDBInstanceOutcome outcome =
                    client.DeleteDBInstance(request);

            if (outcome.IsSuccess()) {
                std::cout << "DB instance deletion has started."
                          << std::endl;
            }
            else {
                std::cerr << "Error with RDS::DeleteDBInstance. "
                          << outcome.GetError().GetMessage()
                          << std::endl;
                result = false;
            }
        }

        std::cout
                << "Waiting for DB instance to delete before deleting the
 parameter group."
                << std::endl;
        std::cout << "This may take a while." << std::endl;

        int counter = 0;
        Aws::RDS::Model::DBInstance dbInstance;
        do {
            std::this_thread::sleep_for(std::chrono::seconds(1));
            ++counter;
            if (counter > 800) {
```

```
                std::cerr << "Wait for instance to delete timed out ofter " <<
 counter
                          << " seconds." << std::endl;
                return false;
            }

            dbInstance = Aws::RDS::Model::DBInstance();
            // 16. Wait for the DB instance to be deleted.
            if (!describeDBInstance(dbInstanceIdentifier, dbInstance, client)) {
                return false;
            }

            if (dbInstance.DBInstanceIdentifierHasBeenSet() && (counter % 20) ==
 0) {
                std::cout << "Current DB instance status is '"
                          << dbInstance.GetDBInstanceStatus()
                          << "' after " << counter << " seconds." << std::endl;
            }
        } while (dbInstance.DBInstanceIdentifierHasBeenSet());
    }

    if (!parameterGroupName.empty()) {
        // 17. Delete the parameter group.
        Aws::RDS::Model::DeleteDBParameterGroupRequest request;
        request.SetDBParameterGroupName(parameterGroupName);

        Aws::RDS::Model::DeleteDBParameterGroupOutcome outcome =
                client.DeleteDBParameterGroup(request);

        if (outcome.IsSuccess()) {
            std::cout << "The DB parameter group was successfully deleted."
                      << std::endl;
        }
        else {
            std::cerr << "Error with RDS::DeleteDBParameterGroup. "
                      << outcome.GetError().GetMessage()
                      << std::endl;
            result = false;
        }
    }

    return result;
}
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella Documentazione di riferimento delle API AWS SDK for C++ .
  - CreateDBInstance
  - Creato B ParameterGroup
  - CreateDBSnapshot
  - DeleteDBInstance
  - Eliminare DB ParameterGroup
  - Descritto B EngineVersions
  - DescribeDBInstances
  - Descritto B ParameterGroups
  - DescribeDBParameters
  - DescribeDBSnapshots
  - DescribeOrderableDB InstanceOptions
  - Modifica DB ParameterGroup

Go

SDK per Go V2

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Esegui uno scenario interattivo al prompt dei comandi.

```
// GetStartedInstances is an interactive example that shows you how to use the
 AWS SDK for Go
// with Amazon Relation Database Service (Amazon RDS) to do the following:
//
//  1. Create a custom DB parameter group and set parameter values.
```

```
//  2. Create a DB instance that is configured to use the parameter group. The DB
 instance
//     also contains a database.
//  3. Take a snapshot of the DB instance.
//  4. Delete the DB instance and parameter group.
type GetStartedInstances struct {
 sdkConfig  aws.Config
 instances  actions.DbInstances
 questioner demotools.IQuestioner
 helper     IScenarioHelper
 isTestRun  bool
}

// NewGetStartedInstances constructs a GetStartedInstances instance from a
 configuration.
// It uses the specified config to get an Amazon RDS
// client and create wrappers for the actions used in the scenario.
func NewGetStartedInstances(sdkConfig aws.Config, questioner
 demotools.IQuestioner,
 helper IScenarioHelper) GetStartedInstances {
 rdsClient := rds.NewFromConfig(sdkConfig)
 return GetStartedInstances{
  sdkConfig:  sdkConfig,
  instances:  actions.DbInstances{RdsClient: rdsClient},
  questioner: questioner,
  helper:     helper,
 }
}

// Run runs the interactive scenario.
func (scenario GetStartedInstances) Run(dbEngine string, parameterGroupName
 string,
 instanceName string, dbName string) {
 defer func() {
  if r := recover(); r != nil {
   log.Println("Something went wrong with the demo.")
  }
 }()

 log.Println(strings.Repeat("-", 88))
 log.Println("Welcome to the Amazon Relational Database Service (Amazon RDS) DB
 Instance demo.")
 log.Println(strings.Repeat("-", 88))
```

```
 parameterGroup := scenario.CreateParameterGroup(dbEngine, parameterGroupName)
 scenario.SetUserParameters(parameterGroupName)
 instance := scenario.CreateInstance(instanceName, dbEngine, dbName,
 parameterGroup)
 scenario.DisplayConnection(instance)
 scenario.CreateSnapshot(instance)
 scenario.Cleanup(instance, parameterGroup)

 log.Println(strings.Repeat("-", 88))
 log.Println("Thanks for watching!")
 log.Println(strings.Repeat("-", 88))
}

// CreateParameterGroup shows how to get available engine versions for a
 specified
// database engine and create a DB parameter group that is compatible with a
// selected engine family.
func (scenario GetStartedInstances) CreateParameterGroup(dbEngine string,
 parameterGroupName string) *types.DBParameterGroup {

 log.Printf("Checking for an existing DB parameter group named %v.\n",
  parameterGroupName)
 parameterGroup, err := scenario.instances.GetParameterGroup(parameterGroupName)
 if err != nil {
  panic(err)
 }
 if parameterGroup == nil {
  log.Printf("Getting available database engine versions for %v.\n", dbEngine)
  engineVersions, err := scenario.instances.GetEngineVersions(dbEngine, "")
  if err != nil {
   panic(err)
  }

  familySet := map[string]struct{}{}
  for _, family := range engineVersions {
   familySet[*family.DBParameterGroupFamily] = struct{}{}
  }
  var families []string
  for family := range familySet {
   families = append(families, family)
  }
  sort.Strings(families)
  familyIndex := scenario.questioner.AskChoice("Which family do you want to use?
\n", families)
```

```
  log.Println("Creating a DB parameter group.")
  _, err = scenario.instances.CreateParameterGroup(
   parameterGroupName, families[familyIndex], "Example parameter group.")
  if err != nil {
   panic(err)
  }
  parameterGroup, err = scenario.instances.GetParameterGroup(parameterGroupName)
  if err != nil {
   panic(err)
  }
 }
 log.Printf("Parameter group %v:\n", *parameterGroup.DBParameterGroupFamily)
 log.Printf("\tName: %v\n", *parameterGroup.DBParameterGroupName)
 log.Printf("\tARN: %v\n", *parameterGroup.DBParameterGroupArn)
 log.Printf("\tFamily: %v\n", *parameterGroup.DBParameterGroupFamily)
 log.Printf("\tDescription: %v\n", *parameterGroup.Description)
 log.Println(strings.Repeat("-", 88))
 return parameterGroup
}

// SetUserParameters shows how to get the parameters contained in a custom
 parameter
// group and update some of the parameter values in the group.
func (scenario GetStartedInstances) SetUserParameters(parameterGroupName string)
 {
 log.Println("Let's set some parameter values in your parameter group.")
 dbParameters, err := scenario.instances.GetParameters(parameterGroupName, "")
 if err != nil {
  panic(err)
 }
 var updateParams []types.Parameter
 for _, dbParam := range dbParameters {
  if strings.HasPrefix(*dbParam.ParameterName, "auto_increment") &&
   dbParam.IsModifiable && *dbParam.DataType == "integer" {
   log.Printf("The %v parameter is described as:\n\t%v",
    *dbParam.ParameterName, *dbParam.Description)
   rangeSplit := strings.Split(*dbParam.AllowedValues, "-")
   lower, _ := strconv.Atoi(rangeSplit[0])
   upper, _ := strconv.Atoi(rangeSplit[1])
   newValue := scenario.questioner.AskInt(
    fmt.Sprintf("Enter a value between %v and %v:", lower, upper),
    demotools.InIntRange{Lower: lower, Upper: upper})
   dbParam.ParameterValue = aws.String(strconv.Itoa(newValue))
   updateParams = append(updateParams, dbParam)
```

```
  }
 }
 err = scenario.instances.UpdateParameters(parameterGroupName, updateParams)
 if err != nil {
  panic(err)
 }
 log.Println("To get a list of parameters that you set previously, specify a
 source of 'user'.")
 userParameters, err := scenario.instances.GetParameters(parameterGroupName,
 "user")
 if err != nil {
  panic(err)
 }
 log.Println("Here are the parameters you set:")
 for _, param := range userParameters {
  log.Printf("\t%v: %v\n", *param.ParameterName, *param.ParameterValue)
 }
 log.Println(strings.Repeat("-", 88))
}

// CreateInstance shows how to create a DB instance that contains a database of a
// specified type. The database is also configured to use a custom DB parameter
 group.
func (scenario GetStartedInstances) CreateInstance(instanceName string, dbEngine
 string,
 dbName string, parameterGroup *types.DBParameterGroup) *types.DBInstance {

 log.Println("Checking for an existing DB instance.")
 instance, err := scenario.instances.GetInstance(instanceName)
 if err != nil {
  panic(err)
 }
 if instance == nil {
  adminUsername := scenario.questioner.Ask(
   "Enter an administrator username for the database: ", demotools.NotEmpty{})
  adminPassword := scenario.questioner.AskPassword(
   "Enter a password for the administrator (at least 8 characters): ", 7)
  engineVersions, err := scenario.instances.GetEngineVersions(dbEngine,
   *parameterGroup.DBParameterGroupFamily)
  if err != nil {
   panic(err)
  }
  var engineChoices []string
  for _, engine := range engineVersions {
```

```
  engineChoices = append(engineChoices, *engine.EngineVersion)
 }
 engineIndex := scenario.questioner.AskChoice(
  "The available engines for your parameter group are:\n", engineChoices)
 engineSelection := engineVersions[engineIndex]
 instOpts, err :=
scenario.instances.GetOrderableInstances(*engineSelection.Engine,
  *engineSelection.EngineVersion)
 if err != nil {
  panic(err)
 }
 optSet := map[string]struct{}{}
 for _, opt := range instOpts {
  if strings.Contains(*opt.DBInstanceClass, "micro") {
   optSet[*opt.DBInstanceClass] = struct{}{}
  }
 }
 var optChoices []string
 for opt := range optSet {
  optChoices = append(optChoices, opt)
 }
 sort.Strings(optChoices)
 optIndex := scenario.questioner.AskChoice(
  "The available micro DB instance classes for your database engine are:\n",
optChoices)
 storageType := "standard"
 allocatedStorage := int32(5)
 log.Printf("Creating a DB instance named %v and database %v.\n"+
  "The DB instance is configured to use your custom parameter group %v,\n"+
  "selected engine %v,\n"+
  "selected DB instance class %v,"+
  "and %v GiB of %v storage.\n"+
  "This typically takes several minutes.",
  instanceName, dbName, *parameterGroup.DBParameterGroupName,
*engineSelection.EngineVersion,
  optChoices[optIndex], allocatedStorage, storageType)
 instance, err = scenario.instances.CreateInstance(
  instanceName, dbName, *engineSelection.Engine, *engineSelection.EngineVersion,
  *parameterGroup.DBParameterGroupName, optChoices[optIndex], storageType,
  allocatedStorage, adminUsername, adminPassword)
 if err != nil {
  panic(err)
 }
 for *instance.DBInstanceStatus != "available" {
```

```
   scenario.helper.Pause(30)
   instance, err = scenario.instances.GetInstance(instanceName)
   if err != nil {
    panic(err)
   }
  }
  log.Println("Instance created and available.")
 }
 log.Println("Instance data:")
 log.Printf("\tDBInstanceIdentifier: %v\n", *instance.DBInstanceIdentifier)
 log.Printf("\tARN: %v\n", *instance.DBInstanceArn)
 log.Printf("\tStatus: %v\n", *instance.DBInstanceStatus)
 log.Printf("\tEngine: %v\n", *instance.Engine)
 log.Printf("\tEngine version: %v\n", *instance.EngineVersion)
 log.Println(strings.Repeat("-", 88))
 return instance
}

// DisplayConnection displays connection information about a DB instance and tips
// on how to connect to it.
func (scenario GetStartedInstances) DisplayConnection(instance *types.DBInstance)
 {
 log.Println(
  "You can now connect to your database by using your favorite MySQL client.\n" +
   "One way to connect is by using the 'mysql' shell on an Amazon EC2 instance\n"
 +
   "that is running in the same VPC as your DB instance. Pass the endpoint,\n" +
   "port, and administrator username to 'mysql'. Then, enter your password\n" +
   "when prompted:")
 log.Printf("\n\tmysql -h %v -P %v -u %v -p\n",
  *instance.Endpoint.Address, instance.Endpoint.Port, *instance.MasterUsername)
 log.Println("For more information, see the User Guide for RDS:\n" +
  "\thttps://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/
CHAP_GettingStarted.CreatingConnecting.MySQL.html#CHAP_GettingStarted.Connecting.MySQL")
 log.Println(strings.Repeat("-", 88))
}

// CreateSnapshot shows how to create a DB instance snapshot and wait until it's
 available.
func (scenario GetStartedInstances) CreateSnapshot(instance *types.DBInstance) {
 if scenario.questioner.AskBool(
  "Do you want to create a snapshot of your DB instance (y/n)? ", "y") {
  snapshotId := fmt.Sprintf("%v-%v", *instance.DBInstanceIdentifier,
 scenario.helper.UniqueId())
```

```
  log.Printf("Creating a snapshot named %v. This typically takes a few minutes.
\n", snapshotId)
  snapshot, err :=
 scenario.instances.CreateSnapshot(*instance.DBInstanceIdentifier, snapshotId)
  if err != nil {
   panic(err)
  }
  for *snapshot.Status != "available" {
   scenario.helper.Pause(30)
   snapshot, err = scenario.instances.GetSnapshot(snapshotId)
   if err != nil {
    panic(err)
   }
  }
  log.Println("Snapshot data:")
  log.Printf("\tDBSnapshotIdentifier: %v\n", *snapshot.DBSnapshotIdentifier)
  log.Printf("\tARN: %v\n", *snapshot.DBSnapshotArn)
  log.Printf("\tStatus: %v\n", *snapshot.Status)
  log.Printf("\tEngine: %v\n", *snapshot.Engine)
  log.Printf("\tEngine version: %v\n", *snapshot.EngineVersion)
  log.Printf("\tDBInstanceIdentifier: %v\n", *snapshot.DBInstanceIdentifier)
  log.Printf("\tSnapshotCreateTime: %v\n", *snapshot.SnapshotCreateTime)
  log.Println(strings.Repeat("-", 88))
 }
}

// Cleanup shows how to clean up a DB instance and DB parameter group.
// Before the DB parameter group can be deleted, all associated DB instances must
 first be deleted.
func (scenario GetStartedInstances) Cleanup(
 instance *types.DBInstance, parameterGroup *types.DBParameterGroup) {

 if scenario.questioner.AskBool(
  "\nDo you want to delete the database instance and parameter group (y/n)? ",
 "y") {
  log.Printf("Deleting database instance %v.\n", *instance.DBInstanceIdentifier)
  err := scenario.instances.DeleteInstance(*instance.DBInstanceIdentifier)
  if err != nil {
   panic(err)
  }
  log.Println(
   "Waiting for the DB instance to delete. This typically takes several
 minutes.")
  for instance != nil {
```

```
   scenario.helper.Pause(30)
   instance, err = scenario.instances.GetInstance(*instance.DBInstanceIdentifier)
   if err != nil {
    panic(err)
   }
  }
  log.Printf("Deleting parameter group %v.",
 *parameterGroup.DBParameterGroupName)
  err =
 scenario.instances.DeleteParameterGroup(*parameterGroup.DBParameterGroupName)
  if err != nil {
   panic(err)
  }
 }
}
```

Definisci le funzioni richiamate dallo scenario per gestire le operazioni di Amazon RDS.

```
type DbInstances struct {
 RdsClient *rds.Client
}


// GetParameterGroup gets a DB parameter group by name.
func (instances *DbInstances) GetParameterGroup(parameterGroupName string) (
 *types.DBParameterGroup, error) {
 output, err := instances.RdsClient.DescribeDBParameterGroups(
  context.TODO(), &rds.DescribeDBParameterGroupsInput{
   DBParameterGroupName: aws.String(parameterGroupName),
  })
 if err != nil {
  var notFoundError *types.DBParameterGroupNotFoundFault
  if errors.As(err, &notFoundError) {
   log.Printf("Parameter group %v does not exist.\n", parameterGroupName)
   err = nil
  } else {
   log.Printf("Error getting parameter group %v: %v\n", parameterGroupName, err)
  }
  return nil, err
 } else {
```

```
  return &output.DBParameterGroups[0], err
 }
}



// CreateParameterGroup creates a DB parameter group that is based on the
 specified
// parameter group family.
func (instances *DbInstances) CreateParameterGroup(
 parameterGroupName string, parameterGroupFamily string, description string) (
 *types.DBParameterGroup, error) {

 output, err := instances.RdsClient.CreateDBParameterGroup(context.TODO(),
  &rds.CreateDBParameterGroupInput{
   DBParameterGroupName:   aws.String(parameterGroupName),
   DBParameterGroupFamily: aws.String(parameterGroupFamily),
   Description:            aws.String(description),
  })
 if err != nil {
  log.Printf("Couldn't create parameter group %v: %v\n", parameterGroupName, err)
  return nil, err
 } else {
  return output.DBParameterGroup, err
 }
}



// DeleteParameterGroup deletes the named DB parameter group.
func (instances *DbInstances) DeleteParameterGroup(parameterGroupName string)
 error {
 _, err := instances.RdsClient.DeleteDBParameterGroup(context.TODO(),
  &rds.DeleteDBParameterGroupInput{
   DBParameterGroupName: aws.String(parameterGroupName),
  })
 if err != nil {
  log.Printf("Couldn't delete parameter group %v: %v\n", parameterGroupName, err)
  return err
 } else {
  return nil
 }
}
```

```
// GetParameters gets the parameters that are contained in a DB parameter group.
func (instances *DbInstances) GetParameters(parameterGroupName string, source
 string) (
 []types.Parameter, error) {

 var output *rds.DescribeDBParametersOutput
 var params []types.Parameter
 var err error
 parameterPaginator := rds.NewDescribeDBParametersPaginator(instances.RdsClient,
  &rds.DescribeDBParametersInput{
   DBParameterGroupName: aws.String(parameterGroupName),
   Source:               aws.String(source),
  })
 for parameterPaginator.HasMorePages() {
  output, err = parameterPaginator.NextPage(context.TODO())
  if err != nil {
   log.Printf("Couldn't get parameters for %v: %v\n", parameterGroupName, err)
   break
  } else {
   params = append(params, output.Parameters...)
  }
 }
 return params, err
}



// UpdateParameters updates parameters in a named DB parameter group.
func (instances *DbInstances) UpdateParameters(parameterGroupName string, params
 []types.Parameter) error {
 _, err := instances.RdsClient.ModifyDBParameterGroup(context.TODO(),
  &rds.ModifyDBParameterGroupInput{
   DBParameterGroupName: aws.String(parameterGroupName),
   Parameters:           params,
  })
 if err != nil {
  log.Printf("Couldn't update parameters in %v: %v\n", parameterGroupName, err)
  return err
 } else {
  return nil
 }
}
```

```
// CreateSnapshot creates a snapshot of a DB instance.
func (instances *DbInstances) CreateSnapshot(instanceName string, snapshotName
 string) (
 *types.DBSnapshot, error) {
 output, err := instances.RdsClient.CreateDBSnapshot(context.TODO(),
 &rds.CreateDBSnapshotInput{
  DBInstanceIdentifier: aws.String(instanceName),
  DBSnapshotIdentifier: aws.String(snapshotName),
 })
 if err != nil {
  log.Printf("Couldn't create snapshot %v: %v\n", snapshotName, err)
  return nil, err
 } else {
  return output.DBSnapshot, nil
 }
}



// GetSnapshot gets a DB instance snapshot.
func (instances *DbInstances) GetSnapshot(snapshotName string)
 (*types.DBSnapshot, error) {
 output, err := instances.RdsClient.DescribeDBSnapshots(context.TODO(),
  &rds.DescribeDBSnapshotsInput{
   DBSnapshotIdentifier: aws.String(snapshotName),
  })
 if err != nil {
  log.Printf("Couldn't get snapshot %v: %v\n", snapshotName, err)
  return nil, err
 } else {
  return &output.DBSnapshots[0], nil
 }
}



// CreateInstance creates a DB instance.
func (instances *DbInstances) CreateInstance(instanceName string, dbName string,
 dbEngine string, dbEngineVersion string, parameterGroupName string,
 dbInstanceClass string,
```

```
 storageType string, allocatedStorage int32, adminName string, adminPassword
 string) (
 *types.DBInstance, error) {
 output, err := instances.RdsClient.CreateDBInstance(context.TODO(),
 &rds.CreateDBInstanceInput{
  DBInstanceIdentifier: aws.String(instanceName),
  DBName:               aws.String(dbName),
  DBParameterGroupName: aws.String(parameterGroupName),
  Engine:               aws.String(dbEngine),
  EngineVersion:        aws.String(dbEngineVersion),
  DBInstanceClass:      aws.String(dbInstanceClass),
  StorageType:          aws.String(storageType),
  AllocatedStorage:     aws.Int32(allocatedStorage),
  MasterUsername:       aws.String(adminName),
  MasterUserPassword:   aws.String(adminPassword),
 })
 if err != nil {
  log.Printf("Couldn't create instance %v: %v\n", instanceName, err)
  return nil, err
 } else {
  return output.DBInstance, nil
 }
}



// GetInstance gets data about a DB instance.
func (instances *DbInstances) GetInstance(instanceName string) (
 *types.DBInstance, error) {
 output, err := instances.RdsClient.DescribeDBInstances(context.TODO(),
  &rds.DescribeDBInstancesInput{
   DBInstanceIdentifier: aws.String(instanceName),
  })
 if err != nil {
  var notFoundError *types.DBInstanceNotFoundFault
  if errors.As(err, &notFoundError) {
   log.Printf("DB instance %v does not exist.\n", instanceName)
   err = nil
  } else {
   log.Printf("Couldn't get instance %v: %v\n", instanceName, err)
  }
  return nil, err
 } else {
  return &output.DBInstances[0], nil
```

```
 }
}



// DeleteInstance deletes a DB instance.
func (instances *DbInstances) DeleteInstance(instanceName string) error {
 _, err := instances.RdsClient.DeleteDBInstance(context.TODO(),
 &rds.DeleteDBInstanceInput{
  DBInstanceIdentifier:   aws.String(instanceName),
  SkipFinalSnapshot:      true,
  DeleteAutomatedBackups: aws.Bool(true),
 })
 if err != nil {
  log.Printf("Couldn't delete instance %v: %v\n", instanceName, err)
  return err
 } else {
  return nil
 }
}



// GetEngineVersions gets database engine versions that are available for the
 specified engine
// and parameter group family.
func (instances *DbInstances) GetEngineVersions(engine string,
 parameterGroupFamily string) (
 []types.DBEngineVersion, error) {
 output, err := instances.RdsClient.DescribeDBEngineVersions(context.TODO(),
  &rds.DescribeDBEngineVersionsInput{
   Engine:                 aws.String(engine),
   DBParameterGroupFamily: aws.String(parameterGroupFamily),
  })
 if err != nil {
  log.Printf("Couldn't get engine versions for %v: %v\n", engine, err)
  return nil, err
 } else {
  return output.DBEngineVersions, nil
 }
}
```

```
// GetOrderableInstances uses a paginator to get DB instance options that can be
 used to create DB instances that are
// compatible with a set of specifications.
func (instances *DbInstances) GetOrderableInstances(engine string, engineVersion
 string) (
 []types.OrderableDBInstanceOption, error) {

 var output *rds.DescribeOrderableDBInstanceOptionsOutput
 var instanceOptions []types.OrderableDBInstanceOption
 var err error
 orderablePaginator :=
 rds.NewDescribeOrderableDBInstanceOptionsPaginator(instances.RdsClient,
  &rds.DescribeOrderableDBInstanceOptionsInput{
   Engine:        aws.String(engine),
   EngineVersion: aws.String(engineVersion),
  })
 for orderablePaginator.HasMorePages() {
  output, err = orderablePaginator.NextPage(context.TODO())
  if err != nil {
   log.Printf("Couldn't get orderable DB instance options: %v\n", err)
   break
  } else {
   instanceOptions = append(instanceOptions,
 output.OrderableDBInstanceOptions...)
  }
 }
 return instanceOptions, err
}
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella Documentazione di riferimento delle API AWS SDK for Go .
  - CreateDBInstance
  - Creato B ParameterGroup
  - CreateDBSnapshot
  - DeleteDBInstance
  - Eliminare DB ParameterGroup
  - Descritto B EngineVersions
  - DescribeDBInstances

- Descritto B ParameterGroups
- DescribeDBParameters
- DescribeDBSnapshots
- DescribeOrderableDB InstanceOptions
- Modifica DB ParameterGroup

Java

SDK per Java 2.x

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Esegui più operazioni.

```
import com.google.gson.Gson;
import
 software.amazon.awssdk.auth.credentials.EnvironmentVariableCredentialsProvider;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.rds.RdsClient;
import software.amazon.awssdk.services.rds.model.CreateDbInstanceRequest;
import software.amazon.awssdk.services.rds.model.CreateDbInstanceResponse;
import software.amazon.awssdk.services.rds.model.CreateDbParameterGroupResponse;
import software.amazon.awssdk.services.rds.model.CreateDbSnapshotRequest;
import software.amazon.awssdk.services.rds.model.CreateDbSnapshotResponse;
import software.amazon.awssdk.services.rds.model.DBEngineVersion;
import software.amazon.awssdk.services.rds.model.DBInstance;
import software.amazon.awssdk.services.rds.model.DBParameterGroup;
import software.amazon.awssdk.services.rds.model.DBSnapshot;
import software.amazon.awssdk.services.rds.model.DeleteDbInstanceRequest;
import software.amazon.awssdk.services.rds.model.DeleteDbInstanceResponse;
import software.amazon.awssdk.services.rds.model.DescribeDbEngineVersionsRequest;
import
 software.amazon.awssdk.services.rds.model.DescribeDbEngineVersionsResponse;
import software.amazon.awssdk.services.rds.model.DescribeDbInstancesRequest;
import software.amazon.awssdk.services.rds.model.DescribeDbInstancesResponse;
```

```
import
 software.amazon.awssdk.services.rds.model.DescribeDbParameterGroupsResponse;
import software.amazon.awssdk.services.rds.model.DescribeDbParametersResponse;
import software.amazon.awssdk.services.rds.model.DescribeDbSnapshotsRequest;
import software.amazon.awssdk.services.rds.model.DescribeDbSnapshotsResponse;
import
 software.amazon.awssdk.services.rds.model.DescribeOrderableDbInstanceOptionsResponse;
import software.amazon.awssdk.services.rds.model.ModifyDbParameterGroupResponse;
import software.amazon.awssdk.services.rds.model.OrderableDBInstanceOption;
import software.amazon.awssdk.services.rds.model.Parameter;
import software.amazon.awssdk.services.rds.model.RdsException;
import software.amazon.awssdk.services.rds.model.CreateDbParameterGroupRequest;
import
 software.amazon.awssdk.services.rds.model.DescribeDbParameterGroupsRequest;
import software.amazon.awssdk.services.rds.model.DescribeDbParametersRequest;
import software.amazon.awssdk.services.rds.model.ModifyDbParameterGroupRequest;
import
 software.amazon.awssdk.services.rds.model.DescribeOrderableDbInstanceOptionsRequest;
import software.amazon.awssdk.services.rds.model.DeleteDbParameterGroupRequest;
import software.amazon.awssdk.services.secretsmanager.SecretsManagerClient;
import
 software.amazon.awssdk.services.secretsmanager.model.GetSecretValueRequest;
import
 software.amazon.awssdk.services.secretsmanager.model.GetSecretValueResponse;
import java.util.ArrayList;
import java.util.List;

/**
 * Before running this Java (v2) code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 *
 * This example requires an AWS Secrets Manager secret that contains the
 * database credentials. If you do not create a
 * secret, this example will not work. For details, see:
 *
 * https://docs.aws.amazon.com/secretsmanager/latest/userguide/integrating_how-
services-use-secrets_RS.html
 *
 * This Java example performs these tasks:
```

```
 *
 * 1. Returns a list of the available DB engines.
 * 2. Selects an engine family and create a custom DB parameter group.
 * 3. Gets the parameter groups.
 * 4. Gets parameters in the group.
 * 5. Modifies the auto_increment_offset parameter.
 * 6. Gets and displays the updated parameters.
 * 7. Gets a list of allowed engine versions.
 * 8. Gets a list of micro instance classes available for the selected engine.
 * 9. Creates an RDS database instance that contains a MySql database and uses
 * the parameter group.
 * 10. Waits for the DB instance to be ready and prints out the connection
 * endpoint value.
 * 11. Creates a snapshot of the DB instance.
 * 12. Waits for an RDS DB snapshot to be ready.
 * 13. Deletes the RDS DB instance.
 * 14. Deletes the parameter group.
 */
public class RDSScenario {
    public static long sleepTime = 20;
    public static final String DASHES = new String(new char[80]).replace("\0",
 "-");

    public static void main(String[] args) throws InterruptedException {
        final String usage = """

                Usage:
                    <dbGroupName> <dbParameterGroupFamily> <dbInstanceIdentifier>
 <dbName> <dbSnapshotIdentifier> <secretName>

                Where:
                    dbGroupName - The database group name.\s
                    dbParameterGroupFamily - The database parameter group name
 (for example, mysql8.0).
                    dbInstanceIdentifier - The database instance identifier\s
                    dbName - The database name.\s
                    dbSnapshotIdentifier - The snapshot identifier.\s
                    secretName - The name of the AWS Secrets Manager secret that
 contains the database credentials"
                """;

        if (args.length != 6) {
            System.out.println(usage);
            System.exit(1);
```

```
        }

        String dbGroupName = args[0];
        String dbParameterGroupFamily = args[1];
        String dbInstanceIdentifier = args[2];
        String dbName = args[3];
        String dbSnapshotIdentifier = args[4];
        String secretName = args[5];

        Gson gson = new Gson();
        User user = gson.fromJson(String.valueOf(getSecretValues(secretName)),
User.class);
        String masterUsername = user.getUsername();
        String masterUserPassword = user.getPassword();

        Region region = Region.US_WEST_2;
        RdsClient rdsClient = RdsClient.builder()
                .region(region)
                .build();
        System.out.println(DASHES);
        System.out.println("Welcome to the Amazon RDS example scenario.");
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("1. Return a list of the available DB engines");
        describeDBEngines(rdsClient);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("2. Create a custom parameter group");
        createDBParameterGroup(rdsClient, dbGroupName, dbParameterGroupFamily);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("3. Get the parameter group");
        describeDbParameterGroups(rdsClient, dbGroupName);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("4. Get the parameters in the group");
        describeDbParameters(rdsClient, dbGroupName, 0);
        System.out.println(DASHES);

        System.out.println(DASHES);
```

```
        System.out.println("5. Modify the auto_increment_offset parameter");
        modifyDBParas(rdsClient, dbGroupName);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("6. Display the updated value");
        describeDbParameters(rdsClient, dbGroupName, -1);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("7. Get a list of allowed engine versions");
        getAllowedEngines(rdsClient, dbParameterGroupFamily);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("8. Get a list of micro instance classes available for
the selected engine");
        getMicroInstances(rdsClient);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println(
                "9. Create an RDS database instance that contains a MySql
database and uses the parameter group");
        String dbARN = createDatabaseInstance(rdsClient, dbGroupName,
dbInstanceIdentifier, dbName, masterUsername,
                masterUserPassword);
        System.out.println("The ARN of the new database is " + dbARN);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("10. Wait for DB instance to be ready");
        waitForInstanceReady(rdsClient, dbInstanceIdentifier);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("11. Create a snapshot of the DB instance");
        createSnapshot(rdsClient, dbInstanceIdentifier, dbSnapshotIdentifier);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("12. Wait for DB snapshot to be ready");
        waitForSnapshotReady(rdsClient, dbInstanceIdentifier,
dbSnapshotIdentifier);
```

```
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("13. Delete the DB instance");
        deleteDatabaseInstance(rdsClient, dbInstanceIdentifier);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("14. Delete the parameter group");
        deleteParaGroup(rdsClient, dbGroupName, dbARN);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("The Scenario has successfully completed.");
        System.out.println(DASHES);

        rdsClient.close();
    }

    private static SecretsManagerClient getSecretClient() {
        Region region = Region.US_WEST_2;
        return SecretsManagerClient.builder()
                .region(region)

.credentialsProvider(EnvironmentVariableCredentialsProvider.create())
                .build();
    }

    public static String getSecretValues(String secretName) {
        SecretsManagerClient secretClient = getSecretClient();
        GetSecretValueRequest valueRequest = GetSecretValueRequest.builder()
                .secretId(secretName)
                .build();

        GetSecretValueResponse valueResponse =
secretClient.getSecretValue(valueRequest);
        return valueResponse.secretString();
    }

    // Delete the parameter group after database has been deleted.
    // An exception is thrown if you attempt to delete the para group while
database
    // exists.
```

```
   public static void deleteParaGroup(RdsClient rdsClient, String dbGroupName,
String dbARN)
           throws InterruptedException {
       try {
           boolean isDataDel = false;
           boolean didFind;
           String instanceARN;

           // Make sure that the database has been deleted.
           while (!isDataDel) {
               DescribeDbInstancesResponse response =
rdsClient.describeDBInstances();
               List<DBInstance> instanceList = response.dbInstances();
               int listSize = instanceList.size();
               didFind = false;
               int index = 1;
               for (DBInstance instance : instanceList) {
                   instanceARN = instance.dbInstanceArn();
                   if (instanceARN.compareTo(dbARN) == 0) {
                       System.out.println(dbARN + " still exists");
                       didFind = true;
                   }
                   if ((index == listSize) && (!didFind)) {
                       // Went through the entire list and did not find the
database ARN.
                       isDataDel = true;
                   }
                   Thread.sleep(sleepTime * 1000);
                   index++;
               }
           }

           // Delete the para group.
           DeleteDbParameterGroupRequest parameterGroupRequest =
DeleteDbParameterGroupRequest.builder()
                   .dbParameterGroupName(dbGroupName)
                   .build();

           rdsClient.deleteDBParameterGroup(parameterGroupRequest);
           System.out.println(dbGroupName + " was deleted.");

       } catch (RdsException e) {
           System.out.println(e.getLocalizedMessage());
           System.exit(1);
```

```
        }
    }

    // Delete the DB instance.
    public static void deleteDatabaseInstance(RdsClient rdsClient, String
dbInstanceIdentifier) {
        try {
            DeleteDbInstanceRequest deleteDbInstanceRequest =
DeleteDbInstanceRequest.builder()
                    .dbInstanceIdentifier(dbInstanceIdentifier)
                    .deleteAutomatedBackups(true)
                    .skipFinalSnapshot(true)
                    .build();

            DeleteDbInstanceResponse response =
rdsClient.deleteDBInstance(deleteDbInstanceRequest);
            System.out.print("The status of the database is " +
response.dbInstance().dbInstanceStatus());

        } catch (RdsException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }
    }

    // Waits until the snapshot instance is available.
    public static void waitForSnapshotReady(RdsClient rdsClient, String
dbInstanceIdentifier,
            String dbSnapshotIdentifier) {
        try {
            boolean snapshotReady = false;
            String snapshotReadyStr;
            System.out.println("Waiting for the snapshot to become available.");

            DescribeDbSnapshotsRequest snapshotsRequest =
DescribeDbSnapshotsRequest.builder()
                    .dbSnapshotIdentifier(dbSnapshotIdentifier)
                    .dbInstanceIdentifier(dbInstanceIdentifier)
                    .build();

            while (!snapshotReady) {
                DescribeDbSnapshotsResponse response =
rdsClient.describeDBSnapshots(snapshotsRequest);
                List<DBSnapshot> snapshotList = response.dbSnapshots();
```

```
                for (DBSnapshot snapshot : snapshotList) {
                    snapshotReadyStr = snapshot.status();
                    if (snapshotReadyStr.contains("available")) {
                        snapshotReady = true;
                    } else {
                        System.out.print(".");
                        Thread.sleep(sleepTime * 1000);
                    }
                }
            }

            System.out.println("The Snapshot is available!");
        } catch (RdsException | InterruptedException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }
    }

    // Create an Amazon RDS snapshot.
    public static void createSnapshot(RdsClient rdsClient, String
dbInstanceIdentifier, String dbSnapshotIdentifier) {
        try {
            CreateDbSnapshotRequest snapshotRequest =
CreateDbSnapshotRequest.builder()
                    .dbInstanceIdentifier(dbInstanceIdentifier)
                    .dbSnapshotIdentifier(dbSnapshotIdentifier)
                    .build();

            CreateDbSnapshotResponse response =
rdsClient.createDBSnapshot(snapshotRequest);
            System.out.println("The Snapshot id is " +
response.dbSnapshot().dbiResourceId());

        } catch (RdsException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }
    }

    // Waits until the database instance is available.
    public static void waitForInstanceReady(RdsClient rdsClient, String
dbInstanceIdentifier) {
        boolean instanceReady = false;
        String instanceReadyStr;
```

```
        System.out.println("Waiting for instance to become available.");
        try {
            DescribeDbInstancesRequest instanceRequest =
DescribeDbInstancesRequest.builder()
                    .dbInstanceIdentifier(dbInstanceIdentifier)
                    .build();

            String endpoint = "";
            while (!instanceReady) {
                DescribeDbInstancesResponse response =
rdsClient.describeDBInstances(instanceRequest);
                List<DBInstance> instanceList = response.dbInstances();
                for (DBInstance instance : instanceList) {
                    instanceReadyStr = instance.dbInstanceStatus();
                    if (instanceReadyStr.contains("available")) {
                        endpoint = instance.endpoint().address();
                        instanceReady = true;
                    } else {
                        System.out.print(".");
                        Thread.sleep(sleepTime * 1000);
                    }
                }
            }
            System.out.println("Database instance is available! The connection
endpoint is " + endpoint);

        } catch (RdsException | InterruptedException e) {
            System.err.println(e.getMessage());
            System.exit(1);
        }
    }

    // Create a database instance and return the ARN of the database.
    public static String createDatabaseInstance(RdsClient rdsClient,
            String dbGroupName,
            String dbInstanceIdentifier,
            String dbName,
            String masterUsername,
            String masterUserPassword) {

        try {
            CreateDbInstanceRequest instanceRequest =
CreateDbInstanceRequest.builder()
                    .dbInstanceIdentifier(dbInstanceIdentifier)
```

```
                    .allocatedStorage(100)
                    .dbName(dbName)
                    .dbParameterGroupName(dbGroupName)
                    .engine("mysql")
                    .dbInstanceClass("db.m4.large")
                    .engineVersion("8.0")
                    .storageType("standard")
                    .masterUsername(masterUsername)
                    .masterUserPassword(masterUserPassword)
                    .build();

            CreateDbInstanceResponse response =
rdsClient.createDBInstance(instanceRequest);
            System.out.print("The status is " +
response.dbInstance().dbInstanceStatus());
            return response.dbInstance().dbInstanceArn();

        } catch (RdsException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }

        return "";
    }

    // Get a list of micro instances.
    public static void getMicroInstances(RdsClient rdsClient) {
        try {
            DescribeOrderableDbInstanceOptionsRequest dbInstanceOptionsRequest =
DescribeOrderableDbInstanceOptionsRequest
                    .builder()
                    .engine("mysql")
                    .build();

            DescribeOrderableDbInstanceOptionsResponse response = rdsClient

.describeOrderableDBInstanceOptions(dbInstanceOptionsRequest);
            List<OrderableDBInstanceOption> orderableDBInstances =
response.orderableDBInstanceOptions();
            for (OrderableDBInstanceOption dbInstanceOption :
orderableDBInstances) {
                System.out.println("The engine version is " +
dbInstanceOption.engineVersion());
```

```
                System.out.println("The engine description is " +
dbInstanceOption.engine());
            }

        } catch (RdsException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }
    }

    // Get a list of allowed engine versions.
    public static void getAllowedEngines(RdsClient rdsClient, String
dbParameterGroupFamily) {
        try {
            DescribeDbEngineVersionsRequest versionsRequest =
DescribeDbEngineVersionsRequest.builder()
                    .dbParameterGroupFamily(dbParameterGroupFamily)
                    .engine("mysql")
                    .build();

            DescribeDbEngineVersionsResponse response =
rdsClient.describeDBEngineVersions(versionsRequest);
            List<DBEngineVersion> dbEngines = response.dbEngineVersions();
            for (DBEngineVersion dbEngine : dbEngines) {
                System.out.println("The engine version is " +
dbEngine.engineVersion());
                System.out.println("The engine description is " +
dbEngine.dbEngineDescription());
            }

        } catch (RdsException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }
    }

    // Modify auto_increment_offset and auto_increment_increment parameters.
    public static void modifyDBParas(RdsClient rdsClient, String dbGroupName) {
        try {
            Parameter parameter1 = Parameter.builder()
                    .parameterName("auto_increment_offset")
                    .applyMethod("immediate")
                    .parameterValue("5")
                    .build();
```

```
            List<Parameter> paraList = new ArrayList<>();
            paraList.add(parameter1);
            ModifyDbParameterGroupRequest groupRequest =
ModifyDbParameterGroupRequest.builder()
                    .dbParameterGroupName(dbGroupName)
                    .parameters(paraList)
                    .build();

            ModifyDbParameterGroupResponse response =
rdsClient.modifyDBParameterGroup(groupRequest);
            System.out.println("The parameter group " +
response.dbParameterGroupName() + " was successfully modified");

        } catch (RdsException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }
    }

    // Retrieve parameters in the group.
    public static void describeDbParameters(RdsClient rdsClient, String
dbGroupName, int flag) {
        try {
            DescribeDbParametersRequest dbParameterGroupsRequest;
            if (flag == 0) {
                dbParameterGroupsRequest = DescribeDbParametersRequest.builder()
                        .dbParameterGroupName(dbGroupName)
                        .build();
            } else {
                dbParameterGroupsRequest = DescribeDbParametersRequest.builder()
                        .dbParameterGroupName(dbGroupName)
                        .source("user")
                        .build();
            }

            DescribeDbParametersResponse response =
rdsClient.describeDBParameters(dbParameterGroupsRequest);
            List<Parameter> dbParameters = response.parameters();
            String paraName;
            for (Parameter para : dbParameters) {
                // Only print out information about either auto_increment_offset
or
                // auto_increment_increment.
```

```
                paraName = para.parameterName();
                if ((paraName.compareTo("auto_increment_offset") == 0)
                        || (paraName.compareTo("auto_increment_increment ") ==
0)) {
                    System.out.println("*** The parameter name is  " + paraName);
                    System.out.println("*** The parameter value is  " +
para.parameterValue());
                    System.out.println("*** The parameter data type is " +
para.dataType());
                    System.out.println("*** The parameter description is " +
para.description());
                    System.out.println("*** The parameter allowed values  is " +
para.allowedValues());
                }
            }

        } catch (RdsException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }
    }

    public static void describeDbParameterGroups(RdsClient rdsClient, String
dbGroupName) {
        try {
            DescribeDbParameterGroupsRequest groupsRequest =
DescribeDbParameterGroupsRequest.builder()
                    .dbParameterGroupName(dbGroupName)
                    .maxRecords(20)
                    .build();

            DescribeDbParameterGroupsResponse response =
rdsClient.describeDBParameterGroups(groupsRequest);
            List<DBParameterGroup> groups = response.dbParameterGroups();
            for (DBParameterGroup group : groups) {
                System.out.println("The group name is " +
group.dbParameterGroupName());
                System.out.println("The group description is " +
group.description());
            }

        } catch (RdsException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
```

```
        }
    }

    public static void createDBParameterGroup(RdsClient rdsClient, String
dbGroupName, String dbParameterGroupFamily) {
        try {
            CreateDbParameterGroupRequest groupRequest =
CreateDbParameterGroupRequest.builder()
                    .dbParameterGroupName(dbGroupName)
                    .dbParameterGroupFamily(dbParameterGroupFamily)
                    .description("Created by using the AWS SDK for Java")
                    .build();

            CreateDbParameterGroupResponse response =
rdsClient.createDBParameterGroup(groupRequest);
            System.out.println("The group name is " +
response.dbParameterGroup().dbParameterGroupName());

        } catch (RdsException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }
    }

    public static void describeDBEngines(RdsClient rdsClient) {
        try {
            DescribeDbEngineVersionsRequest engineVersionsRequest =
DescribeDbEngineVersionsRequest.builder()
                    .defaultOnly(true)
                    .engine("mysql")
                    .maxRecords(20)
                    .build();

            DescribeDbEngineVersionsResponse response =
rdsClient.describeDBEngineVersions(engineVersionsRequest);
            List<DBEngineVersion> engines = response.dbEngineVersions();

            // Get all DBEngineVersion objects.
            for (DBEngineVersion engineOb : engines) {
                System.out.println("The name of the DB parameter group family for
the database engine is "
                        + engineOb.dbParameterGroupFamily());
                System.out.println("The name of the database engine " +
engineOb.engine());
```

```
                System.out.println("The version number of the database engine " +
 engineOb.engineVersion());
            }

        } catch (RdsException e) {
            System.out.println(e.getLocalizedMessage());
            System.exit(1);
        }
    }
}
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella Documentazione di riferimento delle API AWS SDK for Java 2.x .
  - CreateDBInstance
  - Creato B ParameterGroup
  - CreateDBSnapshot
  - DeleteDBInstance
  - Eliminare DB ParameterGroup
  - Descritto B EngineVersions
  - DescribeDBInstances
  - Descritto B ParameterGroups
  - DescribeDBParameters
  - DescribeDBSnapshots
  - DescribeOrderableDB InstanceOptions
  - Modifica DB ParameterGroup

Kotlin

SDK per Kotlin

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
/**
Before running this code example, set up your development environment, including
 your credentials.

For more information, see the following documentation topic:

https://docs.aws.amazon.com/sdk-for-kotlin/latest/developer-guide/setup.html

This example requires an AWS Secrets Manager secret that contains the database
 credentials. If you do not create a
secret, this example will not work. For more details, see:

https://docs.aws.amazon.com/secretsmanager/latest/userguide/integrating_how-
services-use-secrets_RS.html

This example performs the following tasks:

1. Returns a list of the available DB engines by invoking the
 DescribeDbEngineVersions method.
2. Selects an engine family and create a custom DB parameter group by invoking
 the createDBParameterGroup method.
3. Gets the parameter groups by invoking the DescribeDbParameterGroups method.
4. Gets parameters in the group by invoking the DescribeDbParameters method.
5. Modifies both the auto_increment_offset and auto_increment_increment
 parameters by invoking the modifyDbParameterGroup method.
6. Gets and displays the updated parameters.
7. Gets a list of allowed engine versions by invoking the
 describeDbEngineVersions method.
8. Gets a list of micro instance classes available for the selected engine.
9. Creates an Amazon Relational Database Service (Amazon RDS) database instance
 that contains a MySQL database and uses the parameter group.
10. Waits for DB instance to be ready and prints out the connection endpoint
 value.
11. Creates a snapshot of the DB instance.
12. Waits for the DB snapshot to be ready.
13. Deletes the DB instance.
14. Deletes the parameter group.
 */

var sleepTime: Long = 20
suspend fun main(args: Array<String>) {
    val usage = """
        Usage:
```

```
            <dbGroupName> <dbParameterGroupFamily> <dbInstanceIdentifier>
<dbName> <dbSnapshotIdentifier><secretName>

        Where:
            dbGroupName - The database group name.
            dbParameterGroupFamily - The database parameter group name.
            dbInstanceIdentifier - The database instance identifier.
            dbName -  The database name.
            dbSnapshotIdentifier - The snapshot identifier.
            secretName - The name of the AWS Secrets Manager secret that contains
the database credentials.
    """

    if (args.size != 6) {
        println(usage)
        exitProcess(1)
    }

    val dbGroupName = args[0]
    val dbParameterGroupFamily = args[1]
    val dbInstanceIdentifier = args[2]
    val dbName = args[3]
    val dbSnapshotIdentifier = args[4]
    val secretName = args[5]

    val gson = Gson()
    val user = gson.fromJson(getSecretValues(secretName).toString(),
User::class.java)
    val username = user.username
    val userPassword = user.password

    println("1. Return a list of the available DB engines")
    describeDBEngines()

    println("2. Create a custom parameter group")
    createDBParameterGroup(dbGroupName, dbParameterGroupFamily)

    println("3. Get the parameter groups")
    describeDbParameterGroups(dbGroupName)

    println("4. Get the parameters in the group")
    describeDbParameters(dbGroupName, 0)

    println("5. Modify the auto_increment_offset parameter")
```

```
    modifyDBParas(dbGroupName)

    println("6. Display the updated value")
    describeDbParameters(dbGroupName, -1)

    println("7. Get a list of allowed engine versions")
    getAllowedEngines(dbParameterGroupFamily)

    println("8. Get a list of micro instance classes available for the selected
 engine")
    getMicroInstances()

    println("9. Create an RDS database instance that contains a MySql database
 and uses the parameter group")
    val dbARN = createDatabaseInstance(dbGroupName, dbInstanceIdentifier, dbName,
 username, userPassword)
    println("The ARN of the new database is $dbARN")

    println("10. Wait for DB instance to be ready")
    waitForDbInstanceReady(dbInstanceIdentifier)

    println("11. Create a snapshot of the DB instance")
    createDbSnapshot(dbInstanceIdentifier, dbSnapshotIdentifier)

    println("12. Wait for DB snapshot to be ready")
    waitForSnapshotReady(dbInstanceIdentifier, dbSnapshotIdentifier)

    println("13. Delete the DB instance")
    deleteDbInstance(dbInstanceIdentifier)

    println("14. Delete the parameter group")
    if (dbARN != null) {
        deleteParaGroup(dbGroupName, dbARN)
    }

    println("The Scenario has successfully completed.")
}

suspend fun deleteParaGroup(dbGroupName: String, dbARN: String) {
    var isDataDel = false
    var didFind: Boolean
    var instanceARN: String

    RdsClient { region = "us-west-2" }.use { rdsClient ->
```

```
        // Make sure that the database has been deleted.
        while (!isDataDel) {
            val response = rdsClient.describeDbInstances()
            val instanceList = response.dbInstances
            val listSize = instanceList?.size
            isDataDel = false // Reset this value.
            didFind = false // Reset this value.
            var index = 1
            if (instanceList != null) {
                for (instance in instanceList) {
                    instanceARN = instance.dbInstanceArn.toString()
                    if (instanceARN.compareTo(dbARN) == 0) {
                        println("$dbARN still exists")
                        didFind = true
                    }
                    if (index == listSize && !didFind) {
                        // Went through the entire list and did not find the
 database name.
                        isDataDel = true
                    }
                    index++
                }
            }
        }

        // Delete the para group.
        val parameterGroupRequest = DeleteDbParameterGroupRequest {
            dbParameterGroupName = dbGroupName
        }
        rdsClient.deleteDbParameterGroup(parameterGroupRequest)
        println("$dbGroupName was deleted.")
    }
}

suspend fun deleteDbInstance(dbInstanceIdentifierVal: String) {
    val deleteDbInstanceRequest = DeleteDbInstanceRequest {
        dbInstanceIdentifier = dbInstanceIdentifierVal
        deleteAutomatedBackups = true
        skipFinalSnapshot = true
    }

    RdsClient { region = "us-west-2" }.use { rdsClient ->
        val response = rdsClient.deleteDbInstance(deleteDbInstanceRequest)
```

```
        print("The status of the database is
 ${response.dbInstance?.dbInstanceStatus}")
    }
}

// Waits until the snapshot instance is available.
suspend fun waitForSnapshotReady(dbInstanceIdentifierVal: String?,
 dbSnapshotIdentifierVal: String?) {
    var snapshotReady = false
    var snapshotReadyStr: String
    println("Waiting for the snapshot to become available.")

    val snapshotsRequest = DescribeDbSnapshotsRequest {
        dbSnapshotIdentifier = dbSnapshotIdentifierVal
        dbInstanceIdentifier = dbInstanceIdentifierVal
    }

    while (!snapshotReady) {
        RdsClient { region = "us-west-2" }.use { rdsClient ->
            val response = rdsClient.describeDbSnapshots(snapshotsRequest)
            val snapshotList: List<DbSnapshot>? = response.dbSnapshots
            if (snapshotList != null) {
                for (snapshot in snapshotList) {
                    snapshotReadyStr = snapshot.status.toString()
                    if (snapshotReadyStr.contains("available")) {
                        snapshotReady = true
                    } else {
                        print(".")
                        delay(sleepTime * 1000)
                    }
                }
            }
        }
    }
    println("The Snapshot is available!")
}

// Create an Amazon RDS snapshot.
suspend fun createDbSnapshot(dbInstanceIdentifierVal: String?,
 dbSnapshotIdentifierVal: String?) {
    val snapshotRequest = CreateDbSnapshotRequest {
        dbInstanceIdentifier = dbInstanceIdentifierVal
        dbSnapshotIdentifier = dbSnapshotIdentifierVal
    }
```

```
    RdsClient { region = "us-west-2" }.use { rdsClient ->
        val response = rdsClient.createDbSnapshot(snapshotRequest)
        print("The Snapshot id is ${response.dbSnapshot?.dbiResourceId}")
    }
}

// Waits until the database instance is available.
suspend fun waitForDbInstanceReady(dbInstanceIdentifierVal: String?) {
    var instanceReady = false
    var instanceReadyStr: String
    println("Waiting for instance to become available.")

    val instanceRequest = DescribeDbInstancesRequest {
        dbInstanceIdentifier = dbInstanceIdentifierVal
    }
    var endpoint = ""
    while (!instanceReady) {
        RdsClient { region = "us-west-2" }.use { rdsClient ->
            val response = rdsClient.describeDbInstances(instanceRequest)
            val instanceList = response.dbInstances
            if (instanceList != null) {
                for (instance in instanceList) {
                    instanceReadyStr = instance.dbInstanceStatus.toString()
                    if (instanceReadyStr.contains("available")) {
                        endpoint = instance.endpoint?.address.toString()
                        instanceReady = true
                    } else {
                        print(".")
                        delay(sleepTime * 1000)
                    }
                }
            }
        }
    }
    println("Database instance is available! The connection endpoint is
 $endpoint")
}

// Create a database instance and return the ARN of the database.
suspend fun createDatabaseInstance(dbGroupNameVal: String?,
 dbInstanceIdentifierVal: String?, dbNameVal: String?, masterUsernameVal:
 String?, masterUserPasswordVal: String?): String? {
    val instanceRequest = CreateDbInstanceRequest {
```

```
        dbInstanceIdentifier = dbInstanceIdentifierVal
        allocatedStorage = 100
        dbName = dbNameVal
        dbParameterGroupName = dbGroupNameVal
        engine = "mysql"
        dbInstanceClass = "db.m4.large"
        engineVersion = "8.0"
        storageType = "standard"
        masterUsername = masterUsernameVal
        masterUserPassword = masterUserPasswordVal
    }

    RdsClient { region = "us-west-2" }.use { rdsClient ->
        val response = rdsClient.createDbInstance(instanceRequest)
        print("The status is ${response.dbInstance?.dbInstanceStatus}")
        return response.dbInstance?.dbInstanceArn
    }
}

// Get a list of micro instances.
suspend fun getMicroInstances() {
    val dbInstanceOptionsRequest = DescribeOrderableDbInstanceOptionsRequest {
        engine = "mysql"
    }
    RdsClient { region = "us-west-2" }.use { rdsClient ->
        val response =
 rdsClient.describeOrderableDbInstanceOptions(dbInstanceOptionsRequest)
        val orderableDBInstances = response.orderableDbInstanceOptions
        if (orderableDBInstances != null) {
            for (dbInstanceOption in orderableDBInstances) {
                println("The engine version is
 ${dbInstanceOption.engineVersion}")
                println("The engine description is ${dbInstanceOption.engine}")
            }
        }
    }
}

// Get a list of allowed engine versions.
suspend fun getAllowedEngines(dbParameterGroupFamilyVal: String?) {
    val versionsRequest = DescribeDbEngineVersionsRequest {
        dbParameterGroupFamily = dbParameterGroupFamilyVal
        engine = "mysql"
    }
```

```
    RdsClient { region = "us-west-2" }.use { rdsClient ->
        val response = rdsClient.describeDbEngineVersions(versionsRequest)
        val dbEngines: List<DbEngineVersion>? = response.dbEngineVersions
        if (dbEngines != null) {
            for (dbEngine in dbEngines) {
                println("The engine version is ${dbEngine.engineVersion}")
                println("The engine description is
 ${dbEngine.dbEngineDescription}")
            }
        }
    }
}

// Modify the auto_increment_offset parameter.
suspend fun modifyDBParas(dbGroupName: String) {
    val parameter1 = Parameter {
        parameterName = "auto_increment_offset"
        applyMethod = ApplyMethod.Immediate
        parameterValue = "5"
    }

    val paraList: ArrayList<Parameter> = ArrayList()
    paraList.add(parameter1)
    val groupRequest = ModifyDbParameterGroupRequest {
        dbParameterGroupName = dbGroupName
        parameters = paraList
    }

    RdsClient { region = "us-west-2" }.use { rdsClient ->
        val response = rdsClient.modifyDbParameterGroup(groupRequest)
        println("The parameter group ${response.dbParameterGroupName} was
 successfully modified")
    }
}

// Retrieve parameters in the group.
suspend fun describeDbParameters(dbGroupName: String?, flag: Int) {
    val dbParameterGroupsRequest: DescribeDbParametersRequest
    dbParameterGroupsRequest = if (flag == 0) {
        DescribeDbParametersRequest {
            dbParameterGroupName = dbGroupName
        }
    } else {
        DescribeDbParametersRequest {
```

```
            dbParameterGroupName = dbGroupName
            source = "user"
        }
    }
    RdsClient { region = "us-west-2" }.use { rdsClient ->
        val response = rdsClient.describeDbParameters(dbParameterGroupsRequest)
        val dbParameters: List<Parameter>? = response.parameters
        var paraName: String
        if (dbParameters != null) {
            for (para in dbParameters) {
                // Only print out information about either auto_increment_offset
 or auto_increment_increment.
                paraName = para.parameterName.toString()
                if (paraName.compareTo("auto_increment_offset") == 0 ||
 paraName.compareTo("auto_increment_increment ") == 0) {
                    println("*** The parameter name is  $paraName")
                    System.out.println("*** The parameter value is
 ${para.parameterValue}")
                    System.out.println("*** The parameter data type is
 ${para.dataType}")
                    System.out.println("*** The parameter description is
 ${para.description}")
                    System.out.println("*** The parameter allowed values  is
 ${para.allowedValues}")
                }
            }
        }
    }
}

suspend fun describeDbParameterGroups(dbGroupName: String?) {
    val groupsRequest = DescribeDbParameterGroupsRequest {
        dbParameterGroupName = dbGroupName
        maxRecords = 20
    }
    RdsClient { region = "us-west-2" }.use { rdsClient ->
        val response = rdsClient.describeDbParameterGroups(groupsRequest)
        val groups = response.dbParameterGroups
        if (groups != null) {
            for (group in groups) {
                println("The group name is ${group.dbParameterGroupName}")
                println("The group description is ${group.description}")
            }
        }
```

```
    }
}

// Create a parameter group.
suspend fun createDBParameterGroup(dbGroupName: String?,
 dbParameterGroupFamilyVal: String?) {
    val groupRequest = CreateDbParameterGroupRequest {
        dbParameterGroupName = dbGroupName
        dbParameterGroupFamily = dbParameterGroupFamilyVal
        description = "Created by using the AWS SDK for Kotlin"
    }

    RdsClient { region = "us-west-2" }.use { rdsClient ->
        val response = rdsClient.createDbParameterGroup(groupRequest)
        println("The group name is
 ${response.dbParameterGroup?.dbParameterGroupName}")
    }
}

// Returns a list of the available DB engines.
suspend fun describeDBEngines() {
    val engineVersionsRequest = DescribeDbEngineVersionsRequest {
        defaultOnly = true
        engine = "mysql"
        maxRecords = 20
    }

    RdsClient { region = "us-west-2" }.use { rdsClient ->
        val response = rdsClient.describeDbEngineVersions(engineVersionsRequest)
        val engines: List<DbEngineVersion>? = response.dbEngineVersions

        // Get all DbEngineVersion objects.
        if (engines != null) {
            for (engineOb in engines) {
                println("The name of the DB parameter group family for the
 database engine is ${engineOb.dbParameterGroupFamily}.")
                println("The name of the database engine ${engineOb.engine}.")
                println("The version number of the database engine
 ${engineOb.engineVersion}")
            }
        }
    }
}
```

```
suspend fun getSecretValues(secretName: String?): String? {
    val valueRequest = GetSecretValueRequest {
        secretId = secretName
    }

    SecretsManagerClient { region = "us-west-2" }.use { secretsClient ->
        val valueResponse = secretsClient.getSecretValue(valueRequest)
        return valueResponse.secretString
    }
}
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella Documentazione di riferimento delle API SDK AWS per Kotlin.
  - CreateDBInstance
  - Creato B ParameterGroup
  - CreateDBSnapshot
  - DeleteDBInstance
  - Eliminare DB ParameterGroup
  - Descritto B EngineVersions
  - DescribeDBInstances
  - Descritto B ParameterGroups
  - DescribeDBParameters
  - DescribeDBSnapshots
  - DescribeOrderableDB InstanceOptions
  - Modifica DB ParameterGroup

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Esegui uno scenario interattivo al prompt dei comandi.

```
class RdsInstanceScenario:
    """Runs a scenario that shows how to get started using Amazon RDS DB
 instances."""

    def __init__(self, instance_wrapper):
        """
        :param instance_wrapper: An object that wraps Amazon RDS DB instance
 actions.
        """
        self.instance_wrapper = instance_wrapper

    def create_parameter_group(self, parameter_group_name, db_engine):
        """
        Shows how to get available engine versions for a specified database
 engine and
        create a DB parameter group that is compatible with a selected engine
 family.

        :param parameter_group_name: The name given to the newly created
 parameter group.
        :param db_engine: The database engine to use as a basis.
        :return: The newly created parameter group.
        """
        print(
            f"Checking for an existing DB instance parameter group named
 {parameter_group_name}."
        )
        parameter_group = self.instance_wrapper.get_parameter_group(
            parameter_group_name
        )
        if parameter_group is None:
            print(f"Getting available database engine versions for {db_engine}.")
            engine_versions =
 self.instance_wrapper.get_engine_versions(db_engine)
            families = list({ver["DBParameterGroupFamily"] for ver in
 engine_versions})
            family_index = q.choose("Which family do you want to use? ",
 families)
            print(f"Creating a parameter group.")
            self.instance_wrapper.create_parameter_group(
                parameter_group_name, families[family_index], "Example parameter
 group."
```

```
            )
            parameter_group = self.instance_wrapper.get_parameter_group(
                parameter_group_name
            )
        print(f"Parameter group {parameter_group['DBParameterGroupName']}:")
        pp(parameter_group)
        print("-" * 88)
        return parameter_group

    def update_parameters(self, parameter_group_name):
        """
        Shows how to get the parameters contained in a custom parameter group and
        update some of the parameter values in the group.

        :param parameter_group_name: The name of the parameter group to query and
modify.
        """
        print("Let's set some parameter values in your parameter group.")
        auto_inc_parameters = self.instance_wrapper.get_parameters(
            parameter_group_name, name_prefix="auto_increment"
        )
        update_params = []
        for auto_inc in auto_inc_parameters:
            if auto_inc["IsModifiable"] and auto_inc["DataType"] == "integer":
                print(f"The {auto_inc['ParameterName']} parameter is described
as:")
                print(f"\t{auto_inc['Description']}")
                param_range = auto_inc["AllowedValues"].split("-")
                auto_inc["ParameterValue"] = str(
                    q.ask(
                        f"Enter a value between {param_range[0]} and
{param_range[1]}: ",
                        q.is_int,
                        q.in_range(int(param_range[0]), int(param_range[1])),
                    )
                )
                update_params.append(auto_inc)
        self.instance_wrapper.update_parameters(parameter_group_name,
update_params)
        print(
            "You can get a list of parameters you've set by specifying a source
of 'user'."
        )
        user_parameters = self.instance_wrapper.get_parameters(
```

```
            parameter_group_name, source="user"
        )
        pp(user_parameters)
        print("-" * 88)

    def create_instance(self, instance_name, db_name, db_engine,
parameter_group):
        """
        Shows how to create a DB instance that contains a database of a specified
        type and is configured to use a custom DB parameter group.

        :param instance_name: The name given to the newly created DB instance.
        :param db_name: The name given to the created database.
        :param db_engine: The engine of the created database.
        :param parameter_group: The parameter group that is associated with the
DB instance.
        :return: The newly created DB instance.
        """
        print("Checking for an existing DB instance.")
        db_inst = self.instance_wrapper.get_db_instance(instance_name)
        if db_inst is None:
            print("Let's create a DB instance.")
            admin_username = q.ask(
                "Enter an administrator user name for the database: ",
q.non_empty
            )
            admin_password = q.ask(
                "Enter a password for the administrator (at least 8 characters):
",
                q.non_empty,
            )
            engine_versions = self.instance_wrapper.get_engine_versions(
                db_engine, parameter_group["DBParameterGroupFamily"]
            )
            engine_choices = [ver["EngineVersion"] for ver in engine_versions]
            print("The available engines for your parameter group are:")
            engine_index = q.choose("Which engine do you want to use? ",
engine_choices)
            engine_selection = engine_versions[engine_index]
            print(
                "The available micro DB instance classes for your database engine
are:"
            )
            inst_opts = self.instance_wrapper.get_orderable_instances(
```

```
                engine_selection["Engine"], engine_selection["EngineVersion"]
            )
            inst_choices = list(
                {
                    opt["DBInstanceClass"]
                    for opt in inst_opts
                    if "micro" in opt["DBInstanceClass"]
                }
            )
            inst_index = q.choose(
                "Which micro DB instance class do you want to use? ",
inst_choices
            )
            group_name = parameter_group["DBParameterGroupName"]
            storage_type = "standard"
            allocated_storage = 5
            print(
                f"Creating a DB instance named {instance_name} and database
{db_name}.\n"
                f"The DB instance is configured to use your custom parameter
group {group_name},\n"
                f"selected engine {engine_selection['EngineVersion']},\n"
                f"selected DB instance class {inst_choices[inst_index]},"
                f"and {allocated_storage} GiB of {storage_type} storage.\n"
                f"This typically takes several minutes."
            )
            db_inst = self.instance_wrapper.create_db_instance(
                db_name,
                instance_name,
                group_name,
                engine_selection["Engine"],
                engine_selection["EngineVersion"],
                inst_choices[inst_index],
                storage_type,
                allocated_storage,
                admin_username,
                admin_password,
            )
            while db_inst.get("DBInstanceStatus") != "available":
                wait(10)
                db_inst = self.instance_wrapper.get_db_instance(instance_name)
        print("Instance data:")
        pp(db_inst)
        print("-" * 88)
```

```
        return db_inst

    @staticmethod
    def display_connection(db_inst):
        """
        Displays connection information about a DB instance and tips on how to
        connect to it.

        :param db_inst: The DB instance to display.
        """
        print(
            "You can now connect to your database using your favorite MySql
 client.\n"
            "One way to connect is by using the 'mysql' shell on an Amazon EC2
 instance\n"
            "that is running in the same VPC as your DB instance. Pass the
 endpoint,\n"
            "port, and administrator user name to 'mysql' and enter your password
\n"
            "when prompted:\n"
        )
        print(
            f"\n\tmysql -h {db_inst['Endpoint']['Address']} -P
 {db_inst['Endpoint']['Port']} "
            f"-u {db_inst['MasterUsername']} -p\n"
        )
        print(
            "For more information, see the User Guide for Amazon RDS:\n"
            "\thttps://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/
CHAP_GettingStarted.CreatingConnecting.MySQL.html#CHAP_GettingStarted.Connecting.MySQL"
        )
        print("-" * 88)

    def create_snapshot(self, instance_name):
        """
        Shows how to create a DB instance snapshot and wait until it's available.

        :param instance_name: The name of a DB instance to snapshot.
        """
        if q.ask(
            "Do you want to create a snapshot of your DB instance (y/n)? ",
 q.is_yesno
        ):
            snapshot_id = f"{instance_name}-{uuid.uuid4()}"
```

```
            print(
                f"Creating a snapshot named {snapshot_id}. This typically takes a
few minutes."
            )
            snapshot = self.instance_wrapper.create_snapshot(snapshot_id,
instance_name)
            while snapshot.get("Status") != "available":
                wait(10)
                snapshot = self.instance_wrapper.get_snapshot(snapshot_id)
            pp(snapshot)
            print("-" * 88)

    def cleanup(self, db_inst, parameter_group_name):
        """
        Shows how to clean up a DB instance and parameter group.
        Before the parameter group can be deleted, all associated DB instances
must first
        be deleted.

        :param db_inst: The DB instance to delete.
        :param parameter_group_name: The DB parameter group to delete.
        """
        if q.ask(
            "\nDo you want to delete the DB instance and parameter group (y/n)?
",
            q.is_yesno,
        ):
            print(f"Deleting DB instance {db_inst['DBInstanceIdentifier']}.")

self.instance_wrapper.delete_db_instance(db_inst["DBInstanceIdentifier"])
            print(
                "Waiting for the DB instance to delete. This typically takes
several minutes."
            )
            while db_inst is not None:
                wait(10)
                db_inst = self.instance_wrapper.get_db_instance(
                    db_inst["DBInstanceIdentifier"]
                )
            print(f"Deleting parameter group {parameter_group_name}.")
            self.instance_wrapper.delete_parameter_group(parameter_group_name)

    def run_scenario(self, db_engine, parameter_group_name, instance_name,
db_name):
```

```
        logging.basicConfig(level=logging.INFO, format="%(levelname)s:
 %(message)s")

        print("-" * 88)
        print(
            "Welcome to the Amazon Relational Database Service (Amazon RDS)\n"
            "get started with DB instances demo."
        )
        print("-" * 88)

        parameter_group = self.create_parameter_group(parameter_group_name,
 db_engine)
        self.update_parameters(parameter_group_name)
        db_inst = self.create_instance(
            instance_name, db_name, db_engine, parameter_group
        )
        self.display_connection(db_inst)
        self.create_snapshot(instance_name)
        self.cleanup(db_inst, parameter_group_name)

        print("\nThanks for watching!")
        print("-" * 88)


if __name__ == "__main__":
    try:
        scenario = RdsInstanceScenario(InstanceWrapper.from_client())
        scenario.run_scenario(
            "mysql",
            "doc-example-parameter-group",
            "doc-example-instance",
            "docexampledb",
        )
    except Exception:
        logging.exception("Something went wrong with the demo.")
```

Definisci le funzioni richiamate dallo scenario per gestire le operazioni di Amazon RDS.

```
class InstanceWrapper:
    """Encapsulates Amazon RDS DB instance actions."""

    def __init__(self, rds_client):
```

```
        """
        :param rds_client: A Boto3 Amazon RDS client.
        """
        self.rds_client = rds_client

    @classmethod
    def from_client(cls):
        """
        Instantiates this class from a Boto3 client.
        """
        rds_client = boto3.client("rds")
        return cls(rds_client)


    def get_parameter_group(self, parameter_group_name):
        """
        Gets a DB parameter group.

        :param parameter_group_name: The name of the parameter group to retrieve.
        :return: The parameter group.
        """
        try:
            response = self.rds_client.describe_db_parameter_groups(
                DBParameterGroupName=parameter_group_name
            )
            parameter_group = response["DBParameterGroups"][0]
        except ClientError as err:
            if err.response["Error"]["Code"] == "DBParameterGroupNotFound":
                logger.info("Parameter group %s does not exist.",
parameter_group_name)
            else:
                logger.error(
                    "Couldn't get parameter group %s. Here's why: %s: %s",
                    parameter_group_name,
                    err.response["Error"]["Code"],
                    err.response["Error"]["Message"],
                )
                raise
        else:
            return parameter_group


    def create_parameter_group(
        self, parameter_group_name, parameter_group_family, description
```

```
    ):
        """
        Creates a DB parameter group that is based on the specified parameter
group
        family.

        :param parameter_group_name: The name of the newly created parameter
group.
        :param parameter_group_family: The family that is used as the basis of
the new
                                       parameter group.
        :param description: A description given to the parameter group.
        :return: Data about the newly created parameter group.
        """
        try:
            response = self.rds_client.create_db_parameter_group(
                DBParameterGroupName=parameter_group_name,
                DBParameterGroupFamily=parameter_group_family,
                Description=description,
            )
        except ClientError as err:
            logger.error(
                "Couldn't create parameter group %s. Here's why: %s: %s",
                parameter_group_name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return response


    def delete_parameter_group(self, parameter_group_name):
        """
        Deletes a DB parameter group.

        :param parameter_group_name: The name of the parameter group to delete.
        :return: Data about the parameter group.
        """
        try:
            self.rds_client.delete_db_parameter_group(
                DBParameterGroupName=parameter_group_name
            )
        except ClientError as err:
```

```
            logger.error(
                "Couldn't delete parameter group %s. Here's why: %s: %s",
                parameter_group_name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise


    def get_parameters(self, parameter_group_name, name_prefix="", source=None):
        """
        Gets the parameters that are contained in a DB parameter group.

        :param parameter_group_name: The name of the parameter group to query.
        :param name_prefix: When specified, the retrieved list of parameters is
filtered
                            to contain only parameters that start with this
prefix.
        :param source: When specified, only parameters from this source are
retrieved.
                       For example, a source of 'user' retrieves only parameters
that
                       were set by a user.
        :return: The list of requested parameters.
        """
        try:
            kwargs = {"DBParameterGroupName": parameter_group_name}
            if source is not None:
                kwargs["Source"] = source
            parameters = []
            paginator = self.rds_client.get_paginator("describe_db_parameters")
            for page in paginator.paginate(**kwargs):
                parameters += [
                    p
                    for p in page["Parameters"]
                    if p["ParameterName"].startswith(name_prefix)
                ]
        except ClientError as err:
            logger.error(
                "Couldn't get parameters for %s. Here's why: %s: %s",
                parameter_group_name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
```

```
            raise
        else:
            return parameters


    def update_parameters(self, parameter_group_name, update_parameters):
        """
        Updates parameters in a custom DB parameter group.

        :param parameter_group_name: The name of the parameter group to update.
        :param update_parameters: The parameters to update in the group.
        :return: Data about the modified parameter group.
        """
        try:
            response = self.rds_client.modify_db_parameter_group(
                DBParameterGroupName=parameter_group_name,
Parameters=update_parameters
            )
        except ClientError as err:
            logger.error(
                "Couldn't update parameters in %s. Here's why: %s: %s",
                parameter_group_name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return response


    def create_snapshot(self, snapshot_id, instance_id):
        """
        Creates a snapshot of a DB instance.

        :param snapshot_id: The ID to give the created snapshot.
        :param instance_id: The ID of the DB instance to snapshot.
        :return: Data about the newly created snapshot.
        """
        try:
            response = self.rds_client.create_db_snapshot(
                DBSnapshotIdentifier=snapshot_id,
DBInstanceIdentifier=instance_id
            )
            snapshot = response["DBSnapshot"]
```

```
        except ClientError as err:
            logger.error(
                "Couldn't create snapshot of %s. Here's why: %s: %s",
                instance_id,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return snapshot


    def get_snapshot(self, snapshot_id):
        """
        Gets a DB instance snapshot.

        :param snapshot_id: The ID of the snapshot to retrieve.
        :return: The retrieved snapshot.
        """
        try:
            response = self.rds_client.describe_db_snapshots(
                DBSnapshotIdentifier=snapshot_id
            )
            snapshot = response["DBSnapshots"][0]
        except ClientError as err:
            logger.error(
                "Couldn't get snapshot %s. Here's why: %s: %s",
                snapshot_id,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return snapshot


    def get_engine_versions(self, engine, parameter_group_family=None):
        """
        Gets database engine versions that are available for the specified engine
        and parameter group family.

        :param engine: The database engine to look up.
        :param parameter_group_family: When specified, restricts the returned
list of
```

```
                                        engine versions to those that are
compatible with
                                        this parameter group family.
        :return: The list of database engine versions.
        """
        try:
            kwargs = {"Engine": engine}
            if parameter_group_family is not None:
                kwargs["DBParameterGroupFamily"] = parameter_group_family
            response = self.rds_client.describe_db_engine_versions(**kwargs)
            versions = response["DBEngineVersions"]
        except ClientError as err:
            logger.error(
                "Couldn't get engine versions for %s. Here's why: %s: %s",
                engine,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return versions


    def get_orderable_instances(self, db_engine, db_engine_version):
        """
        Gets DB instance options that can be used to create DB instances that are
        compatible with a set of specifications.

        :param db_engine: The database engine that must be supported by the DB
instance.
        :param db_engine_version: The engine version that must be supported by
the DB instance.
        :return: The list of DB instance options that can be used to create a
compatible DB instance.
        """
        try:
            inst_opts = []
            paginator = self.rds_client.get_paginator(
                "describe_orderable_db_instance_options"
            )
            for page in paginator.paginate(
                Engine=db_engine, EngineVersion=db_engine_version
            ):
                inst_opts += page["OrderableDBInstanceOptions"]
```

```
        except ClientError as err:
            logger.error(
                "Couldn't get orderable DB instances. Here's why: %s: %s",
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return inst_opts


    def get_db_instance(self, instance_id):
        """
        Gets data about a DB instance.

        :param instance_id: The ID of the DB instance to retrieve.
        :return: The retrieved DB instance.
        """
        try:
            response = self.rds_client.describe_db_instances(
                DBInstanceIdentifier=instance_id
            )
            db_inst = response["DBInstances"][0]
        except ClientError as err:
            if err.response["Error"]["Code"] == "DBInstanceNotFound":
                logger.info("Instance %s does not exist.", instance_id)
            else:
                logger.error(
                    "Couldn't get DB instance %s. Here's why: %s: %s",
                    instance_id,
                    err.response["Error"]["Code"],
                    err.response["Error"]["Message"],
                )
                raise
        else:
            return db_inst


    def create_db_instance(
        self,
        db_name,
        instance_id,
        parameter_group_name,
        db_engine,
```

```
        db_engine_version,
        instance_class,
        storage_type,
        allocated_storage,
        admin_name,
        admin_password,
    ):
        """
        Creates a DB instance.

        :param db_name: The name of the database that is created in the DB
instance.
        :param instance_id: The ID to give the newly created DB instance.
        :param parameter_group_name: A parameter group to associate with the DB
instance.
        :param db_engine: The database engine of a database to create in the DB
instance.
        :param db_engine_version: The engine version for the created database.
        :param instance_class: The DB instance class for the newly created DB
instance.
        :param storage_type: The storage type of the DB instance.
        :param allocated_storage: The amount of storage allocated on the DB
instance, in GiBs.
        :param admin_name: The name of the admin user for the created database.
        :param admin_password: The admin password for the created database.
        :return: Data about the newly created DB instance.
        """
        try:
            response = self.rds_client.create_db_instance(
                DBName=db_name,
                DBInstanceIdentifier=instance_id,
                DBParameterGroupName=parameter_group_name,
                Engine=db_engine,
                EngineVersion=db_engine_version,
                DBInstanceClass=instance_class,
                StorageType=storage_type,
                AllocatedStorage=allocated_storage,
                MasterUsername=admin_name,
                MasterUserPassword=admin_password,
            )
            db_inst = response["DBInstance"]
        except ClientError as err:
            logger.error(
                "Couldn't create DB instance %s. Here's why: %s: %s",
```

```
                instance_id,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return db_inst


    def delete_db_instance(self, instance_id):
        """
        Deletes a DB instance.

        :param instance_id: The ID of the DB instance to delete.
        :return: Data about the deleted DB instance.
        """
        try:
            response = self.rds_client.delete_db_instance(
                DBInstanceIdentifier=instance_id,
                SkipFinalSnapshot=True,
                DeleteAutomatedBackups=True,
            )
            db_inst = response["DBInstance"]
        except ClientError as err:
            logger.error(
                "Couldn't delete DB instance %s. Here's why: %s: %s",
                instance_id,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return db_inst
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella Documentazione di riferimento delle API SDK AWS per Python (Boto3).
  - CreateDBInstance
  - Creato B ParameterGroup

- CreateDBSnapshot
- DeleteDBInstance
- Eliminare DB ParameterGroup
- Descritto B EngineVersions
- DescribeDBInstances
- Descritto B ParameterGroups
- DescribeDBParameters
- DescribeDBSnapshots
- DescribeOrderableDB InstanceOptions
- Modifica DB ParameterGroup

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Esempi serverless per Amazon RDS che utilizzano SDK AWS

I seguenti esempi di codice mostrano come usare Amazon RDS con AWS SDK.

Esempi

- Connessione a un database Amazon RDS in una funzione Lambda

Connessione a un database Amazon RDS in una funzione Lambda

I seguenti esempi di codice mostrano come implementare una funzione Lambda che si connette a un database RDS. La funzione effettua una semplice richiesta al database e restituisce il risultato.

Go

SDK per Go V2

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri come eseguire la configurazione e l'esecuzione nel repository di Esempi serverless.

Connessione a un database Amazon RDS in una funzione Lambda tramite Go.

```
// Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: Apache-2.0
/*
Golang v2 code here.
*/

package main

import (
 "context"
 "database/sql"
 "encoding/json"
 "fmt"

 "github.com/aws/aws-lambda-go/lambda"
 "github.com/aws/aws-sdk-go-v2/config"
 "github.com/aws/aws-sdk-go-v2/feature/rds/auth"
 _ "github.com/go-sql-driver/mysql"
)

type MyEvent struct {
 Name string `json:"name"`
}

func HandleRequest(event *MyEvent) (map[string]interface{}, error) {

 var dbName string = "DatabaseName"
 var dbUser string = "DatabaseUser"
 var dbHost string = "mysqldb.123456789012.us-east-1.rds.amazonaws.com"
 var dbPort int = 3306
 var dbEndpoint string = fmt.Sprintf("%s:%d", dbHost, dbPort)
 var region string = "us-east-1"

 cfg, err := config.LoadDefaultConfig(context.TODO())
 if err != nil {
  panic("configuration error: " + err.Error())
 }

 authenticationToken, err := auth.BuildAuthToken(
  context.TODO(), dbEndpoint, region, dbUser, cfg.Credentials)
 if err != nil {
  panic("failed to create authentication token: " + err.Error())
```

```
 }

 dsn := fmt.Sprintf("%s:%s@tcp(%s)/%s?tls=true&allowCleartextPasswords=true",
  dbUser, authenticationToken, dbEndpoint, dbName,
 )

 db, err := sql.Open("mysql", dsn)
 if err != nil {
  panic(err)
 }

 defer db.Close()

 var sum int
 err = db.QueryRow("SELECT ?+? AS sum", 3, 2).Scan(&sum)
 if err != nil {
  panic(err)
 }
 s := fmt.Sprint(sum)
 message := fmt.Sprintf("The selected sum is: %s", s)

 messageBytes, err := json.Marshal(message)
 if err != nil {
  return nil, err
 }

 messageString := string(messageBytes)
 return map[string]interface{}{
  "statusCode": 200,
  "headers":    map[string]string{"Content-Type": "application/json"},
  "body":       messageString,
 }, nil
}

func main() {
 lambda.Start(HandleRequest)
}
```

JavaScript

SDK per (v2 JavaScript)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri come eseguire la configurazione e l'esecuzione nel repository di Esempi serverless.

Connessione a un database Amazon RDS in una funzione Lambda tramite Javascript.

```
// Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: Apache-2.0
/*
Node.js code here.
*/
// ES6+ example
import { Signer } from "@aws-sdk/rds-signer";
import mysql from 'mysql2/promise';

async function createAuthToken() {
  // Define connection authentication parameters
  const dbinfo = {

    hostname: process.env.ProxyHostName,
    port: process.env.Port,
    username: process.env.DBUserName,
    region: process.env.AWS_REGION,

  }

  // Create RDS Signer object
  const signer = new Signer(dbinfo);

  // Request authorization token from RDS, specifying the username
  const token = await signer.getAuthToken();
  return token;
}

async function dbOps() {

  // Obtain auth token
```

```
  const token = await createAuthToken();
  // Define connection configuration
  let connectionConfig = {
    host: process.env.ProxyHostName,
    user: process.env.DBUserName,
    password: token,
    database: process.env.DBName,
    ssl: 'Amazon RDS'
  }
  // Create the connection to the DB
  const conn = await mysql.createConnection(connectionConfig);
  // Obtain the result of the query
  const [res,] = await conn.execute('select ?+? as sum', [3, 2]);
  return res;

}

export const handler = async (event) => {
  // Execute database flow
  const result = await dbOps();
  // Return result
  return {
    statusCode: 200,
    body: JSON.stringify("The selected sum is: " + result[0].sum)
  }
};
```

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Esempi di servizi multipli per Amazon RDS che utilizzano SDK AWS

Le seguenti applicazioni di esempio utilizzano AWS gli SDK per combinare Amazon RDS con altri. Servizi AWS Ogni esempio include un collegamento a GitHub, dove è possibile trovare istruzioni su come configurare ed eseguire l'applicazione.

Esempi

Creazione di un tracciatore di elementi di lavoro di Aurora Serverless

I seguenti esempi di codice mostrano come creare un'applicazione Web che traccia gli elementi di lavoro in database Amazon Aurora Serverless e utilizza il Servizio di email semplice Amazon (Amazon SES) per inviare report.

.NET

AWS SDK for .NET

Mostra come utilizzare per AWS SDK for .NET creare un'applicazione Web che tenga traccia degli elementi di lavoro in un database Amazon Aurora e invii report tramite e-mail utilizzando Amazon Simple Email Service (Amazon SES). Questo esempio utilizza un front-end creato con React.js per interagire con un backend .NET RESTful.

- Integra un'applicazione web React con AWS i servizi.
- Elenco, aggiunta e aggiornamento di elementi in una tabella Aurora.
- Invia un report per e-mail degli articoli di lavoro filtrati tramite Amazon SES.
- Distribuisci e gestisci risorse di esempio con lo AWS CloudFormation script incluso.

Per il codice sorgente completo e le istruzioni su come configurarlo ed eseguirlo, vedi l'esempio completo su GitHub.

Servizi utilizzati in questo esempio

- Aurora
- Amazon RDS
- Servizi di dati di Amazon RDS
- Amazon SES

C++

SDK per C++

Mostra come creare un'applicazione Web che traccia gli elementi di lavoro archiviati in un database Amazon Aurora Serverless, con i relativi report.

Per il codice sorgente completo e le istruzioni su come configurare un'API REST C++ che interroga dati Amazon Aurora Serverless e per l'utilizzo da parte di un'applicazione React, consulta l'esempio completo su. GitHub

Servizi utilizzati in questo esempio

- Aurora
- Amazon RDS
- Servizi di dati di Amazon RDS
- Amazon SES

Java

SDK per Java 2.x

Mostra come creare un'applicazione Web che traccia e segnala gli elementi di lavoro archiviati in un database Amazon RDS.

Per il codice sorgente completo e le istruzioni su come configurare un'API Spring REST che interroga i dati Serverless di Amazon Aurora e per l'utilizzo da parte di un'applicazione React, consulta l'esempio completo su. GitHub

Per il codice sorgente completo e le istruzioni su come configurare ed eseguire un esempio che utilizza l'API JDBC, consulta l'esempio completo su. GitHub

Servizi utilizzati in questo esempio

- Aurora
- Amazon RDS
- Servizi di dati di Amazon RDS
- Amazon SES

JavaScript

SDK per (v3 JavaScript)

Mostra come utilizzare AWS SDK for JavaScript (v3) per creare un'applicazione Web che tenga traccia degli elementi di lavoro in un database Amazon Aurora e invii report tramite e-mail utilizzando Amazon Simple Email Service (Amazon SES). Questo esempio utilizza un front-end creato con React.js per interagire con un backend Express Node.js.

- Integra un'applicazione web React.js con. Servizi AWS
- Elenca, aggiungi e aggiorna elementi in una tabella Aurora.
- Invia un report per e-mail degli elementi di lavoro filtrati tramite Amazon SES.
- Distribuisci e gestisci risorse di esempio con lo AWS CloudFormation script incluso.

Per il codice sorgente completo e le istruzioni su come configurarlo ed eseguirlo, vedi l'esempio completo su GitHub.

Servizi utilizzati in questo esempio

- Aurora
- Amazon RDS
- Servizi di dati di Amazon RDS
- Amazon SES

Kotlin

SDK per Kotlin

Mostra come creare un'applicazione Web che traccia e segnala gli elementi di lavoro archiviati in un database Amazon RDS.

Per il codice sorgente completo e le istruzioni su come configurare un'API Spring REST che interroga i dati Serverless di Amazon Aurora e per l'utilizzo da parte di un'applicazione React, consulta l'esempio completo su. GitHub

Servizi utilizzati in questo esempio

- Aurora
- Amazon RDS
- Servizi di dati di Amazon RDS
- Amazon SES

PHP

SDK per PHP

Mostra come utilizzare per AWS SDK for PHP creare un'applicazione Web che tenga traccia degli elementi di lavoro in un database Amazon RDS e invii report tramite e-mail utilizzando

Amazon Simple Email Service (Amazon SES). Questo esempio utilizza un front-end creato con React.js per interagire con un backend PHP RESTful.

- Integra un'applicazione web React.js con AWS i servizi.
- Elenca, aggiungi, aggiorna ed elimina gli elementi in una tabella Amazon RDS.
- Invia un report per e-mail degli articoli di lavoro filtrati tramite Amazon SES.
- Distribuisci e gestisci risorse di esempio con lo AWS CloudFormation script incluso.

Per il codice sorgente completo e le istruzioni su come configurarlo ed eseguirlo, vedi l'esempio completo su GitHub.

Servizi utilizzati in questo esempio

- Aurora
- Amazon RDS
- Servizi di dati di Amazon RDS
- Amazon SES

Python

SDK per Python (Boto3)

Mostra come utilizzare per AWS SDK for Python (Boto3) creare un servizio REST che tenga traccia degli elementi di lavoro in un database Amazon Aurora Serverless e invii report tramite e-mail utilizzando Amazon Simple Email Service (Amazon SES). Questo esempio utilizza il framework Web Flask per gestire il routing HTTP e si integra con una pagina Web React per presentare un'applicazione Web completamente funzionale.

- Crea un servizio Flask REST che si integri con. Servizi AWS
- Lettura, scrittura e aggiornamento degli elementi di lavoro archiviati in un database Aurora Serverless.
- Crea un AWS Secrets Manager segreto che contenga le credenziali del database e usalo per autenticare le chiamate al database.
- Utilizzo di Amazon SES per inviare report via e-mail sugli elementi di lavoro.

Per il codice sorgente completo e le istruzioni su come configurarlo ed eseguirlo, guarda l'esempio completo su. GitHub

Servizi utilizzati in questo esempio

- Aurora
- Amazon RDS
- Servizi di dati di Amazon RDS
- Amazon SES

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Sicurezza in Amazon RDS

La sicurezza del cloud in AWS ha la massima priorità. In quanto cliente AWS, puoi trarre vantaggio da un'architettura di data center e di rete progettata per soddisfare i requisiti delle aziende più esigenti a livello di sicurezza.

La sicurezza è una responsabilità condivisa tra AWS e l'utente. Il modello di responsabilità condivisa descrive questo come sicurezza del cloud e sicurezza nel cloud:

- La sicurezza del cloud: AWS è responsabile della protezione dell'infrastruttura che gestisce i servizi AWS nel cloud AWS. AWS fornisce inoltre servizi che puoi utilizzare in sicurezza. I revisori di terze parti testano e verificano regolarmente l'efficacia della sicurezza come parte dei programmi di conformità AWS. Per ulteriori informazioni sui programmi di conformità che si applicano a Amazon RDS, consulta Servizi AWS coperti dal programma di compliance.
- Sicurezza nel cloud: la tua responsabilità è determinata dal servizio AWS che utilizzi. L'utente è anche responsabile per altri fattori, tra cui la riservatezza dei dati, i requisiti dell'azienda, nonché le leggi e le normative applicabili.

Questa documentazione consente di comprendere come applicare il modello di responsabilità condivisa quando si usa Amazon RDS. I seguenti argomenti illustrano come configurare Amazon RDS per soddisfare gli obiettivi di sicurezza e conformità. Scoprirai anche come utilizzare altri servizi di AWS per monitorare e proteggere le risorse Amazon RDS.

È possibile gestire l'accesso alle risorse Amazon RDS e ai database in un cluster database. Il metodo utilizzato per gestire l'accesso dipende dal tipo di attività che l'utente deve eseguire con Amazon RDS:

- Esegui il cluster di database in un cloud privato virtuale (VPC) in base al servizio Amazon VPC per il maggior controllo possibile degli accessi di rete. Per ulteriori informazioni sulla creazione di un di istanza di database in un VPC, consultaVPC di Amazon VPC e Amazon RDS.
- Utilizza le policy AWS Identity and Access Management (IAM) per assegnare le autorizzazioni che stabiliscono chi è autorizzato a gestire le risorse Amazon RDS. Ad esempio, puoi utilizzare IAM per determinare chi è autorizzato a creare, descrivere, modificare ed eliminare cluster di , applicare tag alle risorse oppure modificare i gruppi di sicurezza.
- Utilizza i gruppi di sicurezza per controllare quali indirizzi IP o istanze Amazon EC2 possono collegarsi ai database su un cluster . Quando crei per la prima volta un cluster , il firewall impedisce

qualsiasi accesso al database tranne che attraverso le regole specificate da un gruppo di sicurezza associato.

- Utilizza connessioni Secure Socket Layer (SSL) o Transport Layer Security (TLS) con istanze DB che eseguono i motori di database Db2, MySQL, MariaDB, PostgreSQL, Oracle o Microsoft SQL Server. Per ulteriori informazioni sull'utilizzo di SSL/TLS con un'istanza database, consulta .
- Utilizza la crittografia di Amazon RDS per proteggere le tue istanze database e gli snapshot a riposo. La crittografia Amazon RDS utilizza l'algoritmo di crittografia AES-256 standard del settore per crittografare i dati sul server che ospita l'istanza database. Per ulteriori informazioni, consulta Crittografia delle risorse Amazon RDS.
- Utilizzo della crittografia di rete e della crittografia trasparente dei dati con le istanze database di Oracle; per ulteriori informazioni, consulta Oracle native network encryption e Oracle Transparent Data Encryption
- Utilizzo delle funzionalità di sicurezza del motore database per controllare chi può accedere ai database su un cluster . Queste funzionalità funzionano come se il database si trovasse sulla rete locale.

> **Note**
>
> Devi configurare la sicurezza solo per i tuoi casi d'uso. Non devi configurare l'accesso di sicurezza per i processi gestiti da Amazon RDS. Questo include la creazione di backup, la replica di dati tra una istanza database primaria e una replica di lettura, nonché altri processi.

Per ulteriori informazioni sulla gestione dell'accesso alle risorse Amazon RDS e ai database su un una istanza, database, consulta i seguenti argomenti.

Argomenti

- Autenticazione del database con Amazon RDS
- Gestione delle password con Amazon RDS e AWS Secrets Manager
- Protezione dei dati in Amazon RDS
- Gestione accessi e identità per Amazon RDS
- Registrazione e monitoraggio in Amazon RDS
- Convalida della conformità per Amazon RDS
- Resilienza in Amazon RDS

- Sicurezza dell'infrastruttura in Amazon RDS
- API Amazon RDS ed endpoint VPC dell'interfaccia (AWS PrivateLink)
- Best practice relative alla sicurezza di Amazon RDS
- Controllo dell'accesso con i gruppi di sicurezza
- Privilegi dell'account utente master
- Utilizzo di ruoli collegati ai servizi per Amazon RDS
- VPC di Amazon VPC e Amazon RDS

Autenticazione del database con Amazon RDS

Amazon RDS supporta diversi modi per autenticare gli utenti del database.

L'autenticazione con password, Kerberos e del database IAM utilizzano diversi metodi di autenticazione nel database. Pertanto, un utente specifico può accedere a un database utilizzando un solo metodo di autenticazione.

Per PostgreSQL, utilizza solo una delle seguenti impostazioni di ruolo per un utente di un database specifico:

- Per utilizzare l'autenticazione del database IAM, assegna all'utente il ruolo `rds_iam`.
- Per utilizzare l'autenticazione Kerberos, assegna all'utente il ruolo `rds_ad`.
- Per utilizzare l'autenticazione con password, non assegnare i ruoli `rds_iam` o `rds_ad`.

Non assegnare entrambi i ruoli `rds_iam` e `rds_ad` a un utente di un database PostgreSQL direttamente o indirettamente mediante l'accesso di concessione nidificato. Se all'utente master, viene aggiunto il ruolo `rds_iam`, l'autenticazione IAM ha la precedenza sull'autenticazione con password, quindi l'utente master dovrà accedere come utente IAM.

> **Important**
>
> Si consiglia di non utilizzare l'utente master direttamente nelle applicazioni. Rispetta piuttosto la best practice di utilizzare un utente del database creato con i privilegi minimi richiesti per l'applicazione.

Argomenti

- Autenticazione password
- Autenticazione del database IAM
- Autenticazione Kerberos

Autenticazione password

Con autenticazione con password il database esegue tutta l'amministrazione degli account utente. È possibile creare utenti con istruzioni SQL come `CREATE USER`, con la clausola appropriata richiesta dal motore DB per specificare le password. Ad esempio, in MySQL l'istruzione è `CREATE USER` *`nome`* `IDENTIFIED BY` *`password`*, mentre in PostgreSQL, l'istruzione è `CREATE USER` *`nome`* `WITH PASSWORD` *`password`*.

Con l'autenticazione con password, il database controlla e autentica gli account utente. Se un motore DB dispone di potenti funzionalità di gestione delle password, può migliorare la sicurezza. L'autenticazione del database potrebbe essere più semplice da amministrare utilizzando l'autenticazione con password quando si dispone di comunità di utenti di piccole dimensioni. Poiché in questo caso vengono generate password di testo non crittografato, l'integrazione con AWS Secrets Manager può migliorare la sicurezza.

Per informazioni sull'utilizzo di Secrets Manager con Amazon RDS, consulta Creazione di un segreto di base e Rotazione di segreti per i database Amazon RDS supportati nella Guida per l'utente di AWS Secrets Manager . Per informazioni sul recupero a livello di programmazione dei segreti nelle applicazioni personalizzate, consulta Recupero del valore segreto nella Guida per l'utente di AWS Secrets Manager .

Autenticazione del database IAM

È possibile autenticarsi nel di istanze DB utilizzando l'autenticazione del database AWS Identity and Access Management (IAM). Con questo metodo di autenticazione, non devi utilizzare una password quando esegui la connessione all'istanza database. Utilizzi invece un token di autenticazione.

Per ulteriori informazioni sull'autenticazione del database IAM, incluse le informazioni sulla disponibilità per motori DB specifici, vedere Autenticazione del database IAM per MariaDB, MySQL e PostgreSQL.

Autenticazione Kerberos

Amazon RDS supporta l'autenticazione esterna degli utenti dei database con Kerberos e Microsoft Active Directory. Kerberos è un protocollo di autenticazione di rete che utilizza ticket e crittografia a chiave simmetrica eliminando la necessità di scambiare password sulla rete. È stato integrato in Microsoft Active Directory ed è progettato per autenticare gli utenti su risorse di rete, ad esempio i database.

Il supporto Amazon RDS per Kerberos e Active Directory offre i vantaggi del Single Sign-On e dell'autenticazione centralizzata degli utenti dei database. Puoi mantenere le tue credenziali utente in Active Directory. Active Directory fornisce una posizione centralizzata per archiviare e gestire le credenziali per più istanze database.

Sono disponibili due modi per consentire agli utenti del database di autenticarsi rispetto alle istanze database. Possono utilizzare credenziali archiviate in AWS Directory Service for Microsoft Active Directory o nell'Active Directory locale.

Le istanze database Microsoft SQL Server e PostgreSQL supportano relazioni di trust di foreste unidirezionali e bidirezionali. Le istanze database Oracle supportano relazioni di trust esterne e di foreste unidirezionali e bidirezionali. Per ulteriori informazioni, consulta Quando creare una relazione di trust nella Guida di amministrazione di AWS Directory Service .

Per informazioni sull'autenticazione Kerberos con uno specifico motore del database, consulta quanto segue:

- Utilizzo di Active Directory gestito da AWS con RDS per SQL Server
- Utilizzo dell'autenticazione Kerberos per MySQL
- Configurazione dell'autenticazione Kerberos per Amazon RDS for Oracle
- Utilizzo di Autenticazione Kerberos con Amazon RDS for PostgreSQL

> **Note**
>
> Attualmente, l'autenticazione Kerberos non è supportata per le istanze database MariaDB.

Gestione delle password con Amazon RDS e AWS Secrets Manager

Amazon RDS si integra con Secrets Manager per gestire le password degli utenti master per le istanze database e i cluster database multi-AZ.

Argomenti

Limitazioni per l'integrazione di Secrets Manager con Amazon RDS

La gestione delle password degli utenti master con Secrets Manager non è supportata per le seguenti funzionalità:

- Creazione di una replica di lettura quando il DB o il cluster DB di origine gestisce le credenziali con Secrets Manager. Questo vale per tutti i motori DB ad eccezione di RDS per SQL Server.
- Implementazioni blu/verde di Amazon RDS
- Amazon RDS Custom
- Switchover Oracle Data Guard
- RDS per Oracle con CDB

Panoramica della gestione delle password degli utenti principali con AWS Secrets Manager

Con AWS Secrets Manager, puoi sostituire le credenziali codificate nel codice, incluse le password del database, con una chiamata API a Secrets Manager per recuperare il segreto a livello di codice. Per ulteriori informazioni su Secrets Manager, consultare la Guida per l'utente di AWS Secrets Manager.

Quando memorizzi i segreti del database in Secrets Manager, ti vengono Account AWS addebitati dei costi. Per informazioni sui prezzi, consulta Prezzi di AWS Secrets Manager.

Puoi specificare che RDS gestisca la password dell'utente master in Secrets Manager per un'istanza database Amazon RDS o un cluster database multi-AZ quando esegui una delle seguenti operazioni:

- Creazione dell'istanza database
- Creazione del cluster database multi-AZ
- Modifica dell'istanza database
- Modifica del cluster database multi-AZ
- Ripristino dell'istanza database da Amazon S3

Quando specifichi che RDS gestisce la password dell'utente master in Secrets Manager, RDS genera la password e la memorizza in Secrets Manager. Puoi interagire direttamente con il segreto per recuperare le credenziali dell'utente master. Puoi anche specificare una chiave gestita dal cliente per crittografare il segreto o utilizzare la chiave KMS fornita da Secrets Manager.

RDS gestisce le impostazioni del segreto e lo ruota ogni sette giorni per impostazione predefinita. È possibile modificare alcune impostazioni, ad esempio il programma di rotazione. Se si elimina un'istanza database che gestisce un segreto in Secrets Manager, vengono eliminati anche il segreto e i metadati associati.

Per connetterti a un'istanza database o a un cluster database multi-AZ con le credenziali in un segreto, puoi recuperare il segreto da Secrets Manager. Per ulteriori informazioni, consulta Recupera segreti da AWS Secrets Manager e Connettiti a un database SQL con credenziali in un AWS Secrets Manager segreto nella Guida per l'AWS Secrets Manager utente.

Vantaggi della gestione delle password degli utenti master con Secrets Manager

La gestione delle password degli utenti master RDS con Secrets Manager offre i seguenti vantaggi:

- RDS genera automaticamente le credenziali del database.
- RDS archivia e gestisce automaticamente le credenziali del database in. AWS Secrets Manager
- RDS ruota regolarmente le credenziali del database, senza richiedere modifiche all'applicazione.
- Secrets Manager protegge le credenziali del database dall'accesso umano e dalla visualizzazione in testo normale.
- Secrets Manager consente il recupero delle credenziali del database nei segreti per le connessioni al database.
- Secrets Manager consente un controllo dettagliato dell'accesso alle credenziali del database nei segreti utilizzando IAM.
- Facoltativamente, puoi separare la crittografia del database dalla crittografia delle credenziali con chiavi KMS diverse.
- Puoi eliminare la gestione manuale e la rotazione delle credenziali del database.
- Puoi monitorare facilmente le credenziali del database con AWS CloudTrail Amazon CloudWatch.

Per ulteriori informazioni sui vantaggi di Secrets Manager, consulta la Guida per l'utente di AWS Secrets Manager.

Autorizzazioni necessarie per l'integrazione di Secrets Manager

Gli utenti devono disporre delle autorizzazioni necessarie per eseguire le operazioni relative all'integrazione di Secrets Manager. Puoi creare le policy IAM che concedono l'autorizzazione per eseguire operazioni API specifiche sulle risorse indicate necessarie. Puoi quindi collegare tali policy ai ruoli o ai set di autorizzazioni IAM che richiedono le autorizzazioni. Per ulteriori informazioni, consulta Gestione accessi e identità per Amazon RDS.

Per le operazioni di creazione, modifica o ripristino, l'utente che specifica che Amazon RDS gestisce la password dell'utente master in Secrets Manager deve disporre delle autorizzazioni per eseguire le seguenti operazioni:

- `kms:DescribeKey`
- `secretsmanager:CreateSecret`

- `secretsmanager:TagResource`

Per le operazioni di creazione, modifica o ripristino, l'utente che specifica la chiave gestita dal cliente per crittografare il segreto in Secrets Manager deve disporre delle autorizzazioni per eseguire le seguenti operazioni:

- `kms:Decrypt`
- `kms:GenerateDataKey`
- `kms:CreateGrant`

Per le operazioni di modifica, l'utente che ruota la password dell'utente master in Secrets Manager deve disporre delle autorizzazioni per eseguire la seguente operazione:

- `secretsmanager:RotateSecret`

Applicazione della gestione RDS della password dell'utente principale in AWS Secrets Manager

È possibile utilizzare le chiavi di condizione IAM per implementare la gestione da parte di RDS della password dell'utente master in AWS Secrets Manager. La seguente policy non consente agli utenti di creare o ripristinare istanze database o cluster database a meno che la password dell'utente master non sia gestita da RDS in Secrets Manager.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": ["rds:CreateDBInstance", "rds:CreateDBCluster",
 "rds:RestoreDBInstanceFromS3", "rds:RestoreDBClusterFromS3"],
            "Resource": "*",
            "Condition": {
                "Bool": {
                    "rds:ManageMasterUserPassword": false
                }
            }
        }
    ]
```

```
}
```

> **Note**
>
> Questa politica applica la gestione delle password al momento della creazione. AWS Secrets Manager Tuttavia, puoi comunque disabilitare l'integrazione di Secrets Manager e impostare manualmente una password master modificando l'istanza.
> Per evitare questa procedura, includi `rds:ModifyDBInstance`, `rds:ModifyDBCluster` nel blocco operazione della policy. Tieni presente che in tal modo impedisci all'utente di applicare ulteriori modifiche alle istanze esistenti in cui non è abilitata l'integrazione di Secrets Manager.

Per ulteriori informazioni sull'utilizzo delle chiavi di condizione nelle policy IAM, consulta Chiavi di condizione delle policy per Amazon RDS e Policy di esempio: Utilizzo di chiavi di condizione.

Gestione della password dell'utente master per un'istanza database con Secrets Manager

È possibile configurare la gestione RDS della password dell'utente master in Secrets Manager eseguendo le seguenti operazioni:

- Creazione di un'istanza database Amazon RDS
- Modifica di un'istanza database Amazon RDS
- Ripristino di un backup in un'istanza database MySQL

È possibile utilizzare la console RDS AWS CLI, o l'API RDS per eseguire queste azioni.

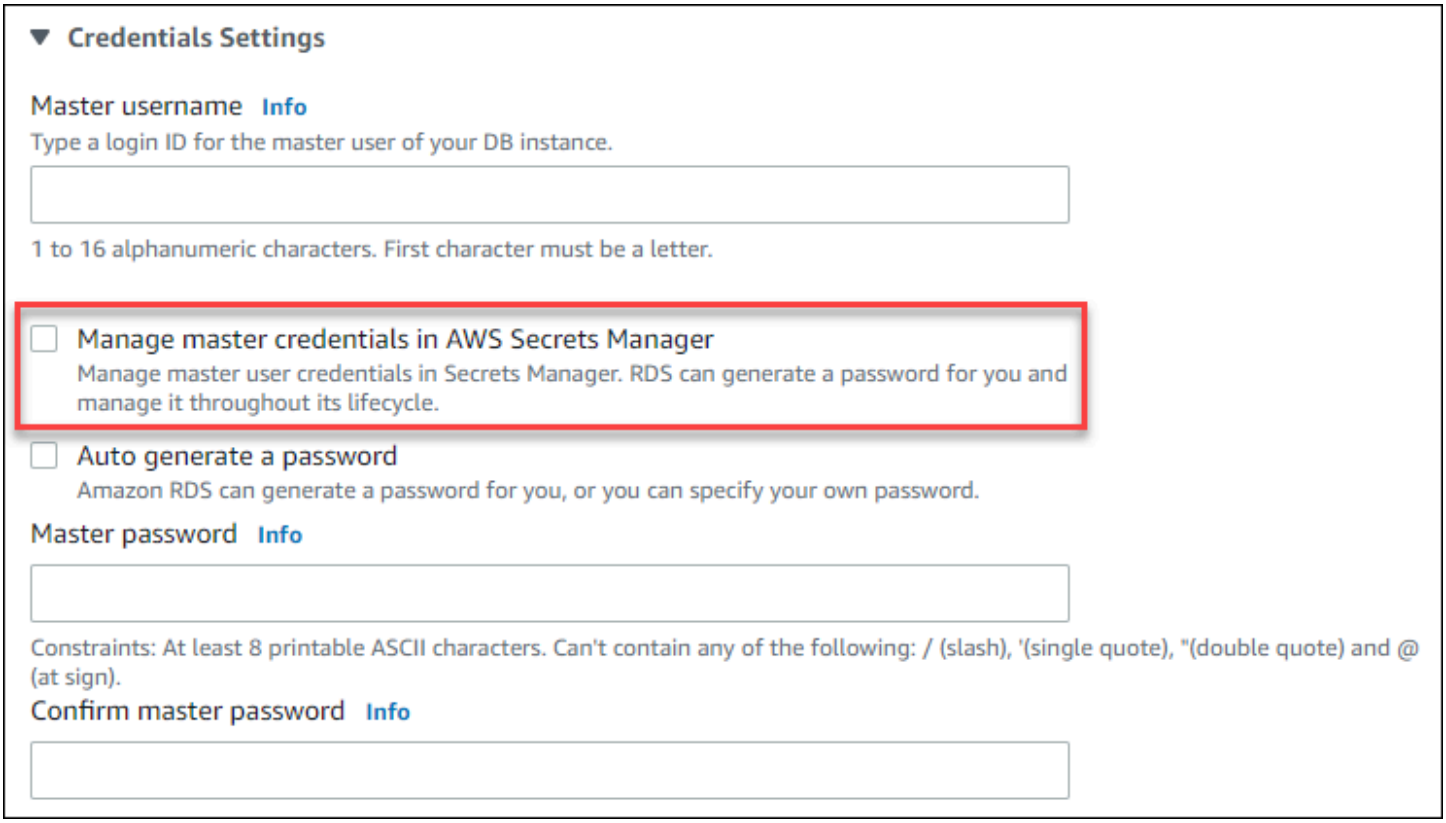
Console

Segui le istruzioni per creare o modificare un'istanza database con la console RDS:

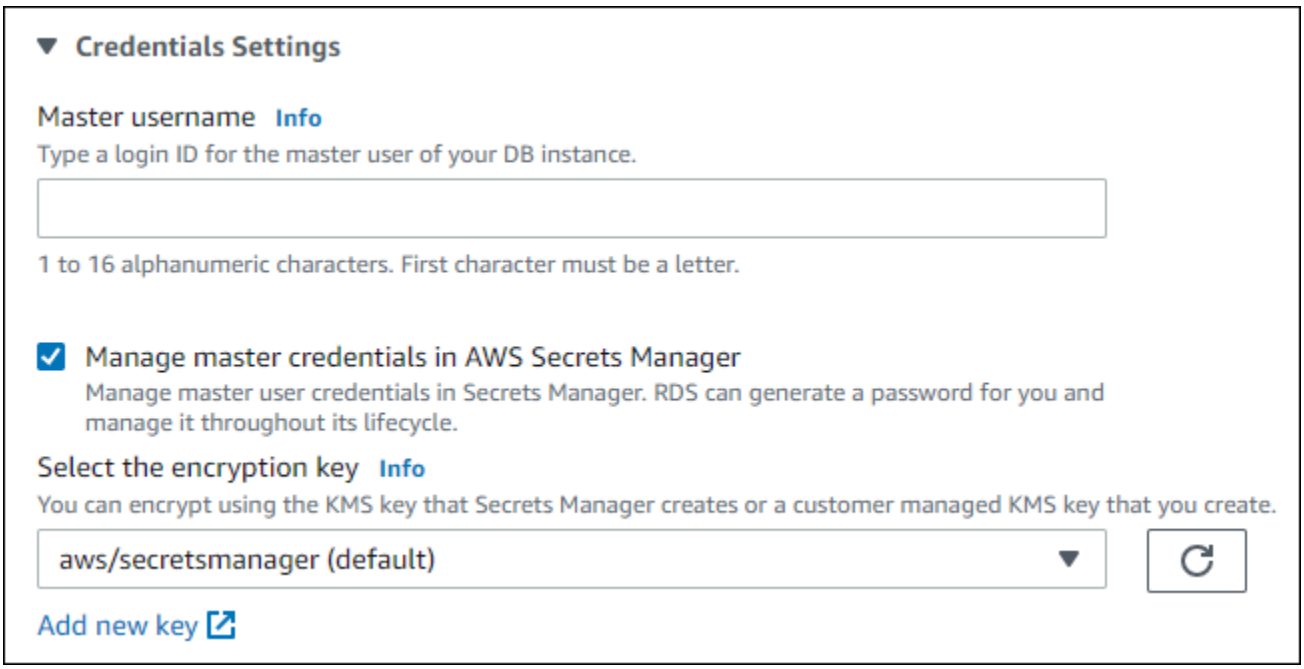
- Creazione di un'istanza database
- Modifica di un'istanza database Amazon RDS
- Importazione di dati da Amazon S3 in una nuova istanza database MySQL

Quando usi la console RDS per eseguire una di queste operazioni, è possibile specificare che la password dell'utente master sia gestita da RDS in Secrets Manager. A tale scopo durante la creazione o il ripristino di un'istanza database, seleziona Manage master credentials in AWS Secrets Manager (Gestione credenziali master in AWS Secrets Manager) in Credential settings (Impostazioni credenziali). Quando modifichi un'istanza database, seleziona Gestisci le credenziali master in AWS Secrets Manager in Impostazioni.

L'immagine seguente è un esempio di impostazione Manage master credentials in AWS Secrets Manager (Gestione credenziali master in AWS Secrets Manager) durante la creazione o il ripristino di un'istanza database.



Quando selezioni questa opzione, RDS genera la password dell'utente master e la gestisce per tutto il suo ciclo di vita in Secrets Manager.



Puoi scegliere di crittografare il segreto con una chiave KMS fornita da Secrets Manager o con una chiave gestita dal cliente creata da te. Dopo che RDS gestisce le credenziali del database per un'istanza database, non puoi modificare la chiave KMS utilizzata per crittografare il segreto.

Puoi scegliere altre impostazioni per soddisfare le tue esigenze. Per ulteriori informazioni sulle impostazioni disponibili per la creazione di un'istanza database, consulta Impostazioni per istanze database. Per ulteriori informazioni sulle impostazioni disponibili per la modifica di un'istanza database, consulta Impostazioni per istanze database.

AWS CLI

Per gestire la password dell'utente principale con RDS in Secrets Manager, specificare l'`--manage-master-user-password`opzione in uno dei seguenti AWS CLI comandi:

- create-db-instance
- modify-db-instance
- restore-db-instance-from-s3

Quando si specifica l'opzione `--manage-master-user-password` in questi comandi, RDS genera la password dell'utente master e la gestisce per tutto il suo ciclo di vita in Secrets Manager.

Per crittografare il segreto, è possibile specificare una chiave gestita dal cliente o utilizzare la chiave KMS predefinita fornita da Secrets Manager. Per specificare la chiave gestita dal cliente usa l'opzione

`--master-user-secret-kms-key-id`. L'identificatore della chiave AWS KMS è l'ARN della chiave, l'ID chiave, l'alias ARN o il nome alias per la chiave KMS. Per utilizzare una chiave KMS in un'altra chiave Account AWS, specifica la chiave ARN o l'alias ARN. Dopo che RDS gestisce le credenziali del database per un'istanza database, non puoi modificare la chiave KMS utilizzata per crittografare il segreto.

Puoi scegliere altre impostazioni per soddisfare le tue esigenze. Per ulteriori informazioni sulle impostazioni disponibili per la creazione di un'istanza database, consulta Impostazioni per istanze database. Per ulteriori informazioni sulle impostazioni disponibili per la modifica di un'istanza database, consulta Impostazioni per istanze database.

Questo esempio crea un'istanza database e specifica che RDS gestisce la password dell'utente master in Secrets Manager. Il segreto viene crittografato utilizzando la chiave KMS fornita da Secrets Manager.

Example

PerLinux, o: macOS Unix

```
aws rds create-db-instance \
    --db-instance-identifier mydbinstance \
    --engine mysql \
    --engine-version 8.0.30 \
    --db-instance-class db.r5b.large \
    --allocated-storage 200 \
    --manage-master-user-password
```

Per Windows:

```
aws rds create-db-instance ^
    --db-instance-identifier mydbinstance ^
    --engine mysql ^
    --engine-version 8.0.30 ^
    --db-instance-class db.r5b.large ^
    --allocated-storage 200 ^
    --manage-master-user-password
```

API RDS

Per specificare che RDS gestisce la password dell'utente master in Secrets Manager, imposta il parametro `ManageMasterUserPassword` su `true` in una delle seguenti operazioni API RDS:

- CreateDBInstance
- ModifyDBInstance
- Ripristina DB S3 InstanceFrom

Quando imposti il parametro `ManageMasterUserPassword` su `true` in una di queste operazioni, RDS genera la password dell'utente master e la gestisce per tutto il suo ciclo di vita in Secrets Manager.

Per crittografare il segreto, è possibile specificare una chiave gestita dal cliente o utilizzare la chiave KMS predefinita fornita da Secrets Manager. Per specificare la chiave gestita dal cliente usa il parametro `MasterUserSecretKmsKeyId`. L'identificatore della chiave AWS KMS è l'ARN della chiave, l'ID chiave, l'alias ARN o il nome alias per la chiave KMS. Per usate una chiave KMS in un Account AWS diverso, specifica l'ARN della chiave o dell'alias. Dopo che RDS gestisce le credenziali del database per un'istanza database, non puoi modificare la chiave KMS utilizzata per crittografare il segreto.

Gestione della password dell'utente master per un cluster database multi-AZ con Secrets Manager

È possibile configurare la gestione RDS della password dell'utente master in Secrets Manager eseguendo le seguenti operazioni:

- Creazione di un cluster di database Multi-AZ
- Modifica di un cluster di database Multi-AZ

È possibile utilizzare la console RDS, l'o l'API RDS per eseguire AWS CLI queste azioni.

Console

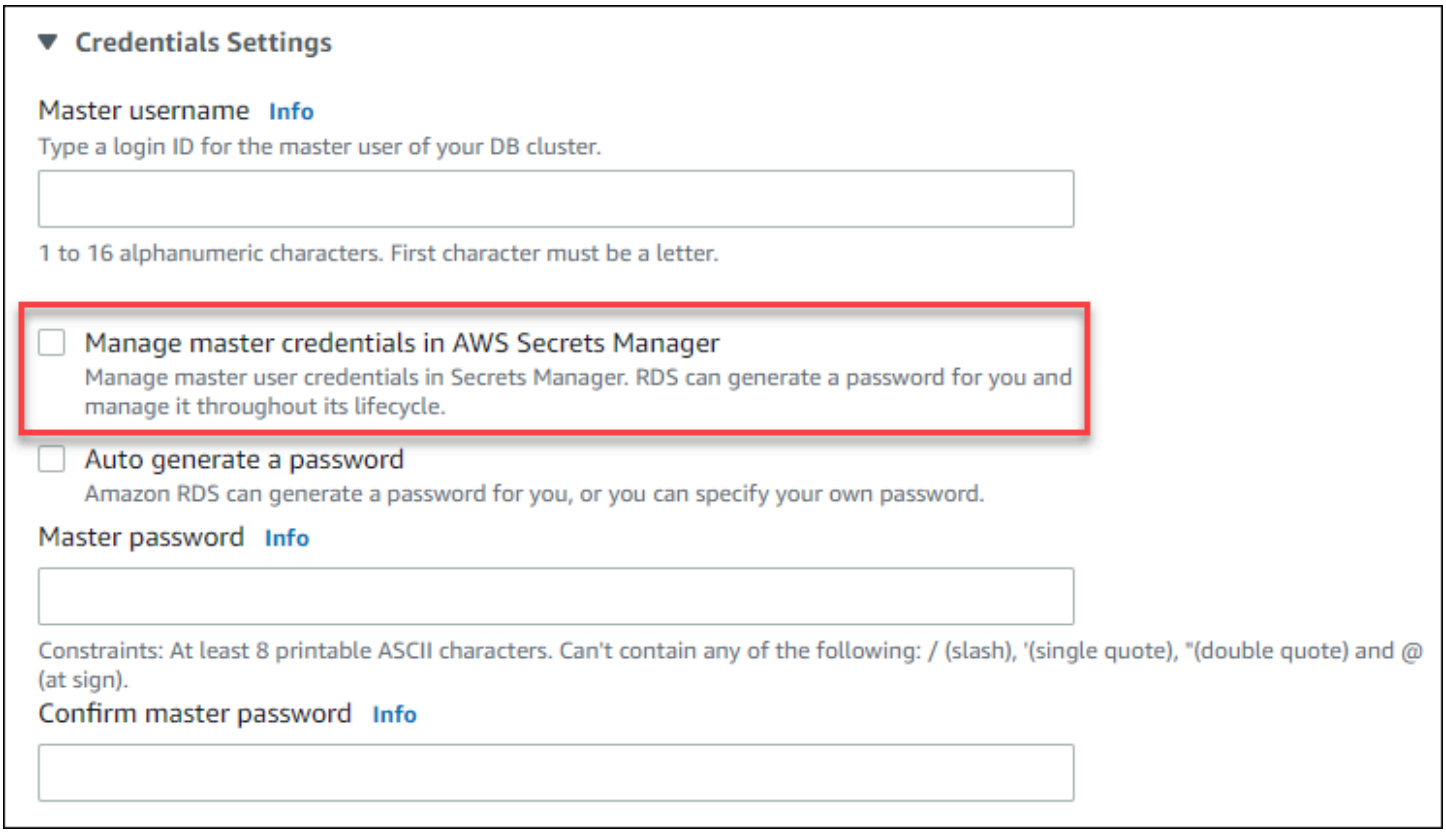
Segui le istruzioni per creare o modificare un cluster database multi-AZ con la console RDS:

- Creazione di un cluster di database
- Modifica di un cluster di database Multi-AZ

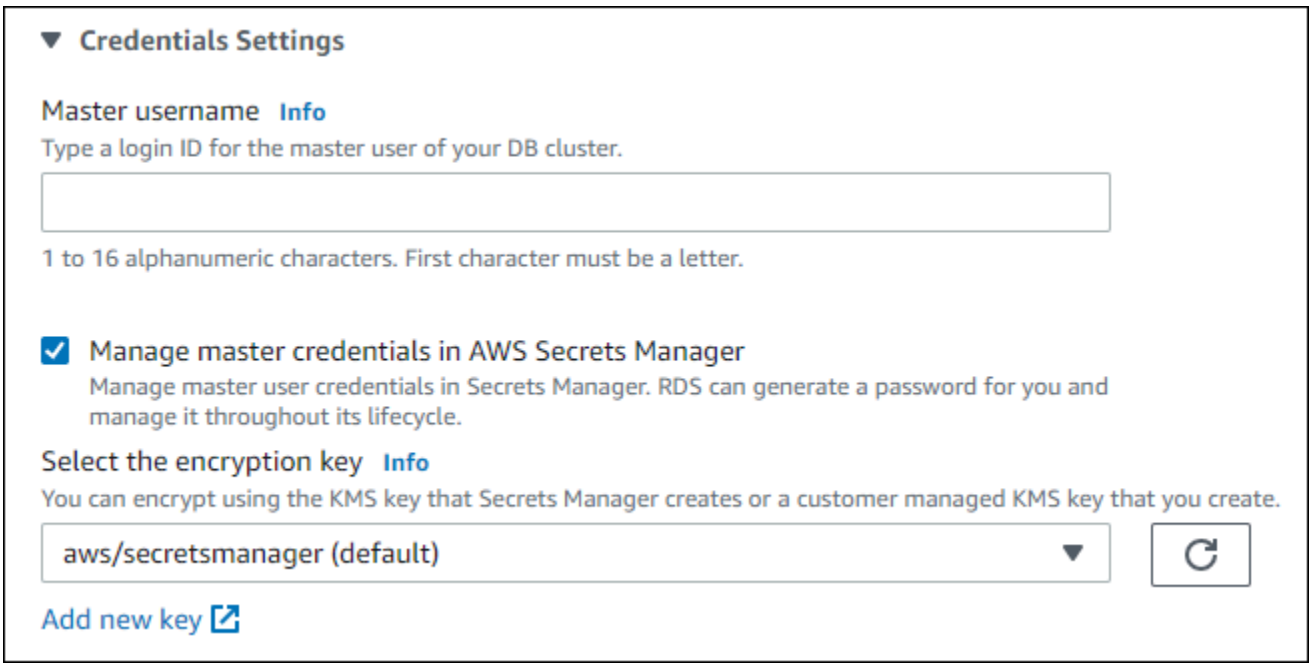
Quando usi la console RDS per eseguire una di queste operazioni, è possibile specificare che la password dell'utente master sia gestita da RDS in Secrets Manager. A tale scopo durante la

creazione di un cluster database, seleziona Manage master credentials in AWS Secrets Manager (Gestione credenziali master in AWS Secrets Manager) in Credential settings (Impostazioni credenziali). Quando modifichi un cluster database, seleziona Manage master credentials in AWS Secrets Manager (Gestione credenziali master in AWS Secrets Manager) in Settings (Impostazioni).

L'immagine seguente è un esempio di impostazione Manage master credentials in AWS Secrets Manager (Gestione credenziali master in AWS Secrets Manager) durante la creazione di un cluster database.



Quando selezioni questa opzione, RDS genera la password dell'utente master e la gestisce per tutto il suo ciclo di vita in Secrets Manager.



Puoi scegliere di crittografare il segreto con una chiave KMS fornita da Secrets Manager o con una chiave gestita dal cliente creata da te. Dopo che RDS gestisce le credenziali del database per un cluster database, non puoi modificare la chiave KMS utilizzata per crittografare il segreto.

Puoi scegliere altre impostazioni per soddisfare le tue esigenze.

Per ulteriori informazioni sulle impostazioni disponibili per la creazione di un cluster database multi-AZ, consulta Impostazioni per la creazione di cluster di database Multi-AZ. Per ulteriori informazioni sulle impostazioni disponibili per la modifica di un cluster database multi-AZ, consulta Impostazioni per la creazione di cluster di database Multi-AZ.

AWS CLI

Per specificare che RDS gestisce la password dell'utente master in Secrets Manager, imposta l'opzione `--manage-master-user-password` in uno dei seguenti comandi:

- create-db-cluster
- modify-db-cluster

Quando si specifica l'opzione `--manage-master-user-password` in questi comandi, RDS genera la password dell'utente master e la gestisce per tutto il suo ciclo di vita in Secrets Manager.

Per crittografare il segreto, è possibile specificare una chiave gestita dal cliente o utilizzare la chiave KMS predefinita fornita da Secrets Manager. Per specificare la chiave gestita dal cliente usa l'opzione

`--master-user-secret-kms-key-id`. L'identificatore della chiave AWS KMS è l'ARN della chiave, l'ID chiave, l'alias ARN o il nome alias per la chiave KMS. Per utilizzare una chiave KMS in un'altra chiave Account AWS, specifica la chiave ARN o l'alias ARN. Dopo che RDS gestisce le credenziali del database per un cluster database, non puoi modificare la chiave KMS utilizzata per crittografare il segreto.

Puoi scegliere altre impostazioni per soddisfare le tue esigenze.

Per ulteriori informazioni sulle impostazioni disponibili per la creazione di un cluster database multi-AZ, consulta Impostazioni per la creazione di cluster di database Multi-AZ. Per ulteriori informazioni sulle impostazioni disponibili per la modifica di un cluster database multi-AZ, consulta Impostazioni per la creazione di cluster di database Multi-AZ.

Questo esempio crea un cluster database multi-AZ e specifica che RDS gestisce la password in Secrets Manager. Il segreto viene crittografato utilizzando la chiave KMS fornita da Secrets Manager.

Example

PerLinux, o: macOS Unix

```
aws rds create-db-cluster \
   --db-cluster-identifier mysql-multi-az-db-cluster \
   --engine mysql \
   --engine-version 8.0.28  \
   --backup-retention-period 1  \
   --allocated-storage 4000 \
   --storage-type io1 \
   --iops 10000 \
   --db-cluster-instance-class db.r6gd.xlarge \
   --manage-master-user-password
```

Per Windows:

```
aws rds create-db-cluster ^
   --db-cluster-identifier mysql-multi-az-db-cluster ^
   --engine mysql ^
   --engine-version 8.0.28 ^
   --backup-retention-period 1 ^
   --allocated-storage 4000 ^
   --storage-type io1 ^
   --iops 10000 ^
   --db-cluster-instance-class db.r6gd.xlarge ^
```

```
    --manage-master-user-password
```

API RDS

Per specificare che RDS gestisce la password dell'utente master in Secrets Manager, imposta il parametro `ManageMasterUserPassword` su `true` in una delle seguenti operazioni:

- CreateDBCluster
- ModifyDBCluster

Quando imposti il parametro `ManageMasterUserPassword` su `true` in una di queste operazioni, RDS genera la password dell'utente master e la gestisce per tutto il suo ciclo di vita in Secrets Manager.

Per crittografare il segreto, è possibile specificare una chiave gestita dal cliente o utilizzare la chiave KMS predefinita fornita da Secrets Manager. Per specificare la chiave gestita dal cliente usa il parametro `MasterUserSecretKmsKeyId`. L'identificatore della chiave AWS KMS è l'ARN della chiave, l'ID chiave, l'alias ARN o il nome alias per la chiave KMS. Per usate una chiave KMS in un Account AWS diverso, specifica l'ARN della chiave o dell'alias. Dopo che RDS gestisce le credenziali del database per un cluster database, non puoi modificare la chiave KMS utilizzata per crittografare il segreto.

Rotazione del segreto della password dell'utente master per un'istanza database

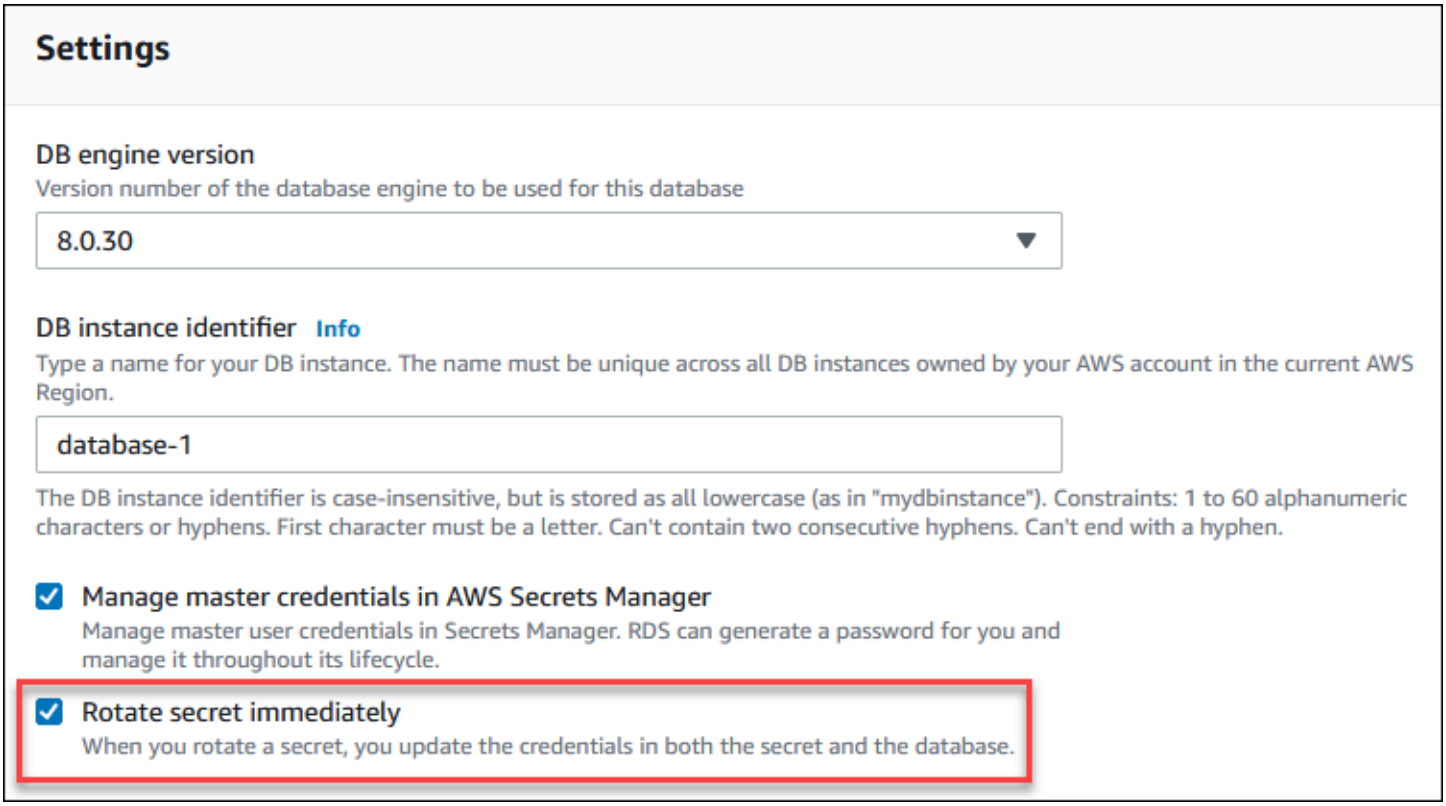
Quando RDS ruota il segreto della password di un utente master, Secrets Manager genera una nuova versione del segreto esistente. La nuova versione del segreto contiene la nuova password dell'utente master. Amazon RDS modifica la password dell'utente master per l'istanza database in modo che corrisponda alla password per la nuova versione del segreto.

Puoi ruotare un segreto immediatamente invece di aspettare la rotazione programmata. Per ruotare il segreto della password dell'utente master in Secrets Manager, modifica l'istanza database. Per ulteriori informazioni sulla modifica di un'istanza database, consulta Modifica di un'istanza database Amazon RDS.

È possibile ruotare immediatamente la password segreta di un utente principale con la console RDS, l'API RDS o l'API RDS. AWS CLI La nuova password è sempre lunga 28 caratteri e contiene almeno un carattere maiuscolo e minuscolo, un numero e una punteggiatura.

Console

Per ruotare il segreto della password dell'utente master utilizzando la console RDS, modifica l'istanza database e seleziona Rotate secret immediately (Ruota il segreto immediatamente) in Settings (Impostazioni).



Per modificare un'istanza database con la console RDS segui le istruzioni presenti in Modifica di un'istanza database Amazon RDS. È necessario scegliere Apply immediately (Applica immediatamente) nella pagina di conferma.

AWS CLI

Per ruotare la password segreta di un utente principale utilizzando il AWS CLI, usa il comando e specifica l'modify-db-instanceopzione. `--rotate-master-user-password` È necessario specificare l'opzione `--apply-immediately` quando si ruota la password master.

Questo esempio ruota il segreto della password dell'utente master.

Example

Per LinuxmacOS, oUnix:

```
aws rds modify-db-instance \
```

```
    --db-instance-identifier mydbinstance \
    --rotate-master-user-password \
    --apply-immediately
```

Per Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --rotate-master-user-password ^
    --apply-immediately
```

API RDS

È possibile ruotare il segreto della password dell'utente master utilizzando l'operazione ModifyDBInstance e impostando il parametro `RotateMasterUserPassword` su `true`. È necessario impostare il parametro `ApplyImmediately` su `true` quando si ruota la password master.

Rotazione del segreto della password dell'utente master per un cluster database multi-AZ

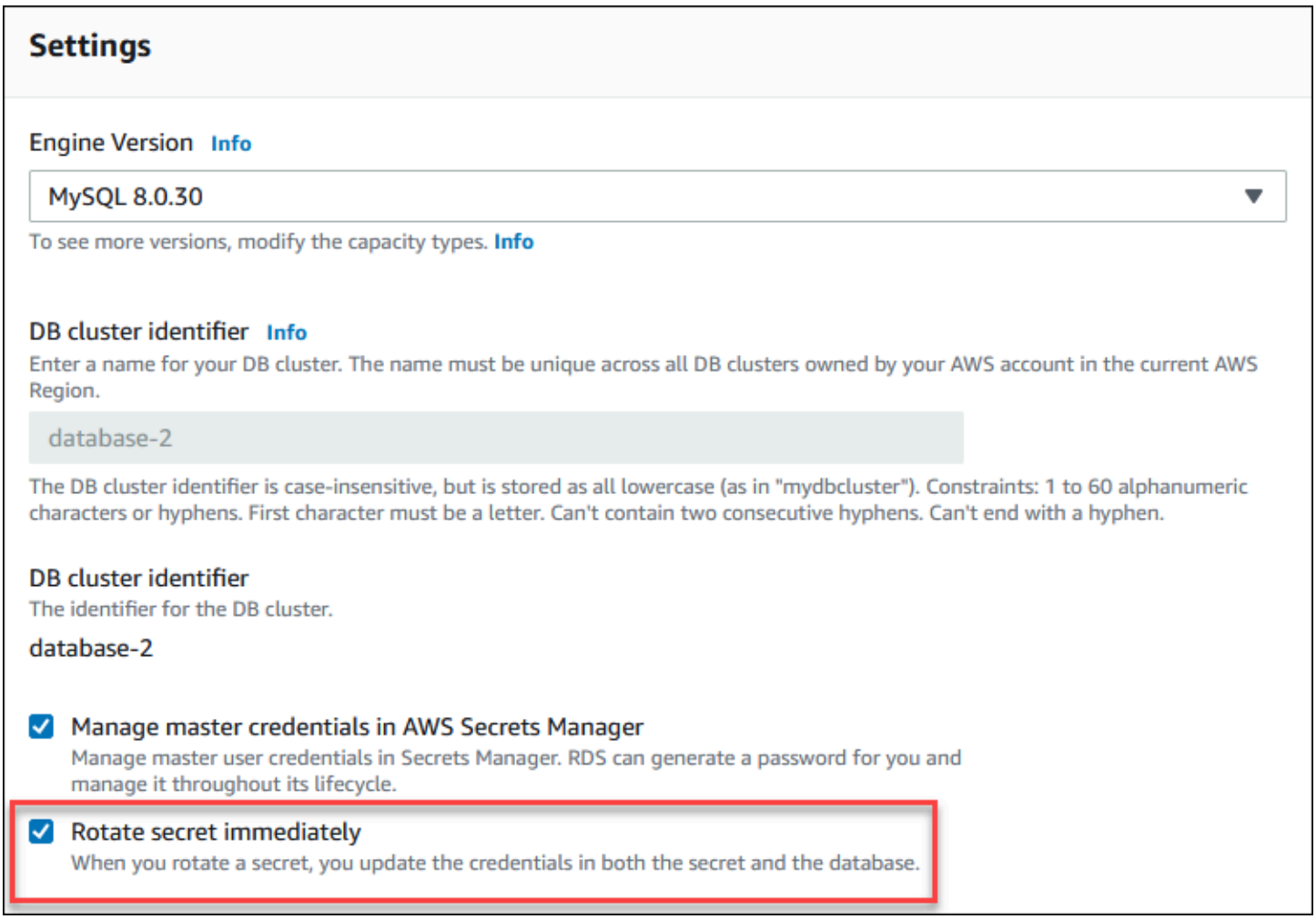
Quando RDS ruota il segreto della password di un utente master, Secrets Manager genera una nuova versione del segreto esistente. La nuova versione del segreto contiene la nuova password dell'utente master. Amazon RDS modifica la password dell'utente master per il cluster database multi-AZ in modo che corrisponda alla password per la nuova versione del segreto.

Puoi ruotare un segreto immediatamente invece di aspettare la rotazione programmata. Per ruotare il segreto della password dell'utente master in Secrets Manager, modifica il cluster database multi-AZ. Per informazioni sulla modifica di un cluster database multi-AZ, consulta Modifica di un cluster di database Multi-AZ.

È possibile ruotare immediatamente la password segreta di un utente principale con la console RDS AWS CLI, o l'API RDS. La nuova password è sempre lunga 28 caratteri e contiene almeno un carattere maiuscolo e minuscolo, un numero e una punteggiatura.

Console

Per ruotare il segreto della password dell'utente master utilizzando la console RDS, modifica il cluster database multi-AZ e seleziona Rotate secret immediately (Ruota il segreto immediatamente) in Settings (Impostazioni).



Per modificare un cluster database multi-AZ con la console RDS segui le istruzioni presenti in Modifica di un cluster di database Multi-AZ. È necessario scegliere Apply immediately (Applica immediatamente) nella pagina di conferma.

AWS CLI

Per ruotare la password segreta di un utente principale utilizzando il AWS CLI, usa il comando e specifica l'modify-db-clusteropzione. `--rotate-master-user-password` È necessario specificare l'opzione `--apply-immediately` quando si ruota la password master.

Questo esempio ruota il segreto della password dell'utente master.

Example

Per LinuxmacOS, oUnix:

```
aws rds modify-db-cluster \
```

```
    --db-cluster-identifier mydbcluster \
    --rotate-master-user-password \
    --apply-immediately
```

Per Windows:

```
aws rds modify-db-cluster ^
    --db-cluster-identifier mydbcluster ^
    --rotate-master-user-password ^
    --apply-immediately
```

API RDS

È possibile ruotare il segreto della password dell'utente master utilizzando l'operazione ModifyDBCluster e impostando il parametro `RotateMasterUserPassword` su `true`. È necessario impostare il parametro `ApplyImmediately` su `true` quando si ruota la password master.

Visualizzazione dei dettagli di un segreto per un'istanza database

Puoi recuperare i tuoi segreti utilizzando la console (https://console.aws.amazon.com/secretsmanager/) o il AWS CLI (comando get-secret-valueSecrets Manager).

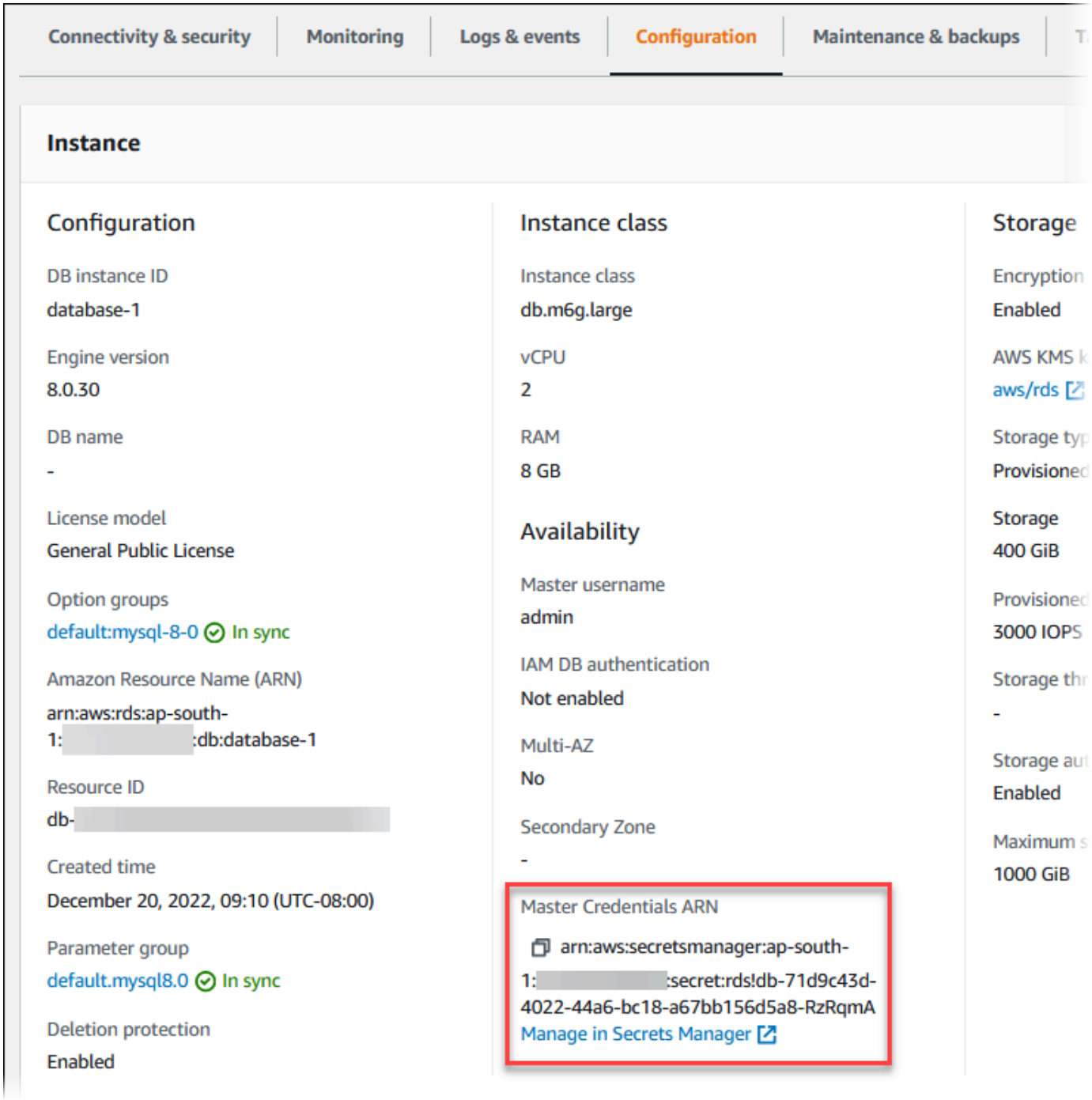
Puoi trovare l'Amazon Resource Name (ARN) di un segreto gestito da RDS in Secrets Manager con la console RDS AWS CLI, o l'API RDS.

Console

Per visualizzare i dettagli di un segreto gestito da RDS in Secrets Manager

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, scegli Databases (Database).
3. Scegliere il nome dell'istanza database per visualizzarne i dettagli.
4. Scegli la scheda Configurazione.

   In Master Credentials ARN (ARN credenziali master), puoi visualizzare l'ARN del segreto.



Puoi selezionare il collegamento Manage in Secrets Manager (Gestisci in Secrets Manager) per visualizzare e gestire il segreto nella console di Secrets Manager.

AWS CLI

È possibile utilizzare il comando describe-db-instancesRDS CLI per trovare le seguenti informazioni su un segreto gestito da RDS in Secrets Manager:

- `SecretArn`: l'ARN del segreto
- `SecretStatus`: lo stato del segreto

  I valori possibili per lo stato sono:

  - `creating`: il segreto è in fase di creazione.
  - `active`: il segreto è disponibile per l'uso normale e la rotazione.
  - `rotating`: il segreto è in fase di rotazione.
  - `impaired`: il segreto può essere utilizzato per accedere alle credenziali del database, ma non può essere ruotato. Un segreto può avere questo stato se, ad esempio, le autorizzazioni vengono modificate in modo che RDS non può più accedere al segreto o alla chiave KMS del segreto.

    Quando un segreto ha questo stato, puoi correggere la condizione che lo ha causato. Se correggi la condizione che ha causato lo stato, lo stato rimane `impaired` fino alla rotazione successiva. In alternativa, è possibile modificare l'istanza database per disattivare la gestione automatica delle credenziali del database e quindi modificare nuovamente l'istanza database per attivare la gestione automatica delle credenziali del database. Per modificare l'istanza DB, utilizzate l'`--manage-master-user-password`opzione nel comando. modify-db-instance
- `KmsKeyId`: l'ARN della chiave KMS utilizzata per crittografare il segreto

Specifica l'opzione `--db-instance-identifier` per mostrare l'output per un'istanza database specifica. Questo esempio mostra l'output di un segreto utilizzato da un'istanza database.

Example

```
aws rds describe-db-instances --db-instance-identifier mydbinstance
```

Di seguito è illustrato l'output di esempio di un segreto:

```
"MasterUserSecret": {
                "SecretArn": "arn:aws:secretsmanager:eu-west-1:123456789012:secret:rds!
db-033d7456-2c96-450d-9d48-f5de3025e51c-xmJRDx",
                "SecretStatus": "active",
                "KmsKeyId": "arn:aws:kms:eu-
west-1:123456789012:key/0987dcba-09fe-87dc-65ba-ab0987654321"
            }
```

Quando si dispone dell'ARN segreto, è possibile visualizzare i dettagli sul segreto utilizzando il comando get-secret-valueSecrets Manager CLI.

Questo esempio mostra i dettagli del segreto nell'output di esempio precedente.

Example

PerLinux, omacOS: Unix

```
aws secretsmanager get-secret-value \
    --secret-id 'arn:aws:secretsmanager:eu-west-1:123456789012:secret:rds!
db-033d7456-2c96-450d-9d48-f5de3025e51c-xmJRDx'
```

Per Windows:

```
aws secretsmanager get-secret-value ^
    --secret-id 'arn:aws:secretsmanager:eu-west-1:123456789012:secret:rds!
db-033d7456-2c96-450d-9d48-f5de3025e51c-xmJRDx'
```

API RDS

È possibile visualizzare l'ARN, lo stato e la chiave KMS di un segreto gestito da RDS in Secrets Manager utilizzando l'operazione DescribeDBInstances e impostando il parametro `DBInstanceIdentifier` su un identificatore di istanza database. I dettagli del segreto sono inclusi nell'output.

Quando si dispone dell'ARN segreto, è possibile visualizzare i dettagli sul segreto utilizzando l'operazione GetSecretValueSecrets Manager.

Visualizzazione dei dettagli di un segreto per un cluster database multi-AZ

Puoi recuperare i tuoi segreti utilizzando la console (https://console.aws.amazon.com/secretsmanager/) o il AWS CLI (comando get-secret-valueSecrets Manager).

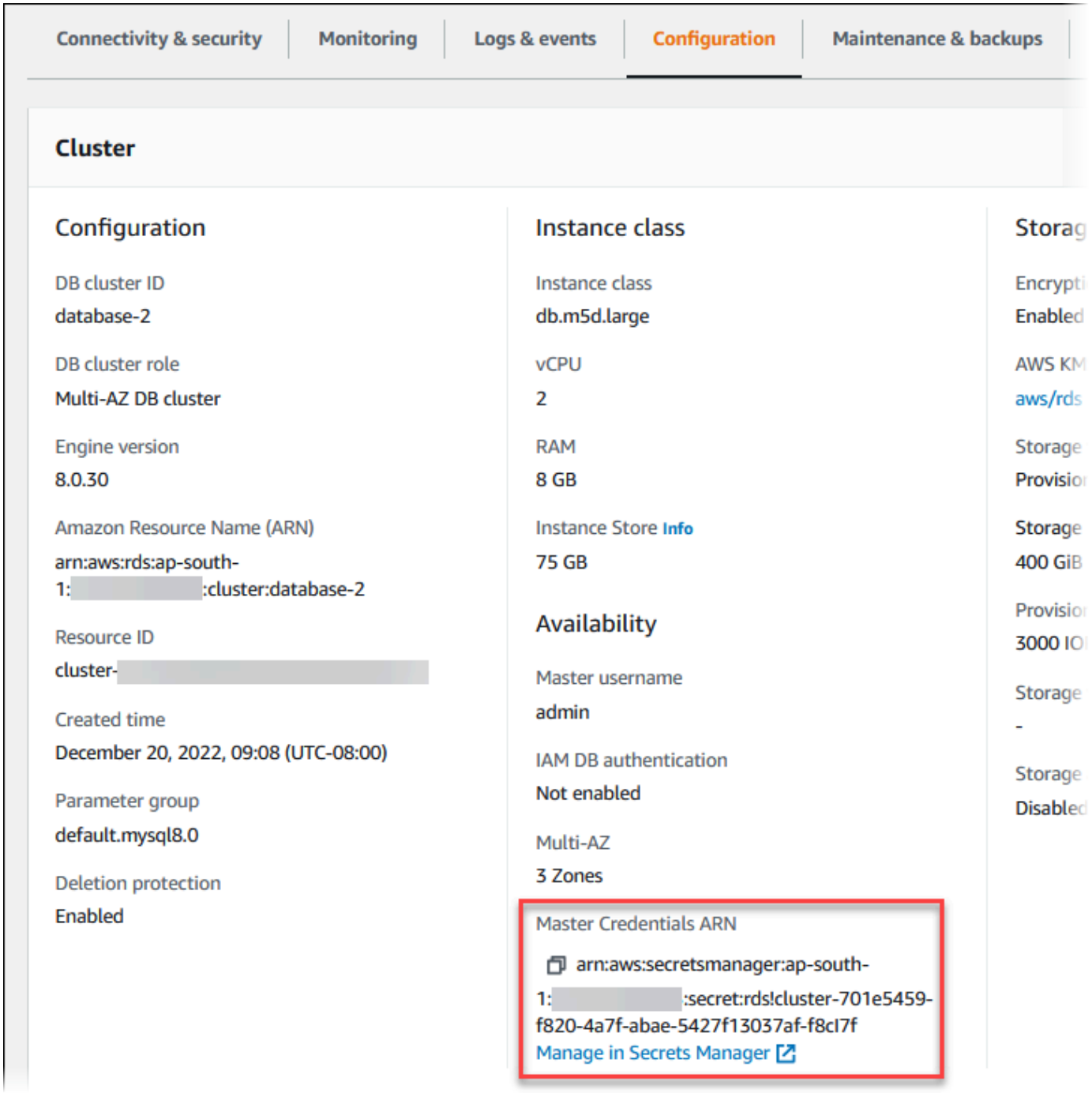
Puoi trovare l'Amazon Resource Name (ARN) di un segreto gestito da RDS in Secrets Manager con la console RDS AWS CLI, l'o l'API RDS.

Console

Per visualizzare i dettagli di un segreto gestito da RDS in Secrets Manager

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, scegli Databases (Database).
3. Scegli il nome del cluster database multi-AZ per visualizzarne i dettagli.
4. Scegli la scheda Configurazione.

   In Master Credentials ARN (ARN credenziali master), puoi visualizzare l'ARN del segreto.



Puoi selezionare il collegamento Manage in Secrets Manager (Gestisci in Secrets Manager) per visualizzare e gestire il segreto nella console di Secrets Manager.

AWS CLI

È possibile utilizzare il AWS CLI [describe-db-clusters](#)comando RDS per trovare le seguenti informazioni su un segreto gestito da RDS Aurora Manager:

- `SecretArn`: l'ARN del segreto
- `SecretStatus`: lo stato del segreto

  I valori possibili per lo stato sono:

  - `creating`: il segreto è in fase di creazione.
  - `active`: il segreto è disponibile per l'uso normale e la rotazione.
  - `rotating`: il segreto è in fase di rotazione.
  - `impaired`: il segreto può essere utilizzato per accedere alle credenziali del database, ma non può essere ruotato. Un segreto può avere questo stato se, ad esempio, le autorizzazioni vengono modificate in modo che RDS non può più accedere al segreto o alla chiave KMS del segreto.

    Quando un segreto ha questo stato, puoi correggere la condizione che lo ha causato. Se correggi la condizione che ha causato lo stato, lo stato rimane `impaired` fino alla rotazione successiva. In alternativa, è possibile modificare il cluster database per disattivare la gestione automatica delle credenziali del database e quindi modificare nuovamente il cluster database per attivare la gestione automatica delle credenziali del database. Per modificare il cluster DB, utilizzare l'`--manage-master-user-password`opzione nel comando. modify-db-cluster
- `KmsKeyId`: l'ARN della chiave KMS utilizzata per crittografare il segreto

Specifica l'opzione `--db-cluster-identifier` per mostrare l'output per un cluster database specifico. Questo esempio mostra l'output di un segreto utilizzato da un cluster database.

Example

```
aws rds describe-db-clusters --db-cluster-identifier mydbcluster
```

L'esempio seguente mostra l'output di un segreto:

```
"MasterUserSecret": {
                "SecretArn": "arn:aws:secretsmanager:eu-west-1:123456789012:secret:rds!
cluster-033d7456-2c96-450d-9d48-f5de3025e51c-xmJRDx",
                "SecretStatus": "active",
                "KmsKeyId": "arn:aws:kms:eu-
west-1:123456789012:key/0987dcba-09fe-87dc-65ba-ab0987654321"
            }
```

Quando si dispone dell'ARN segreto, è possibile visualizzare i dettagli sul segreto utilizzando il comando get-secret-valueSecrets Manager CLI.

Questo esempio mostra i dettagli del segreto nell'output di esempio precedente.

Example

PerLinux, omacOS: Unix

```
aws secretsmanager get-secret-value \
    --secret-id 'arn:aws:secretsmanager:eu-west-1:123456789012:secret:rds!
cluster-033d7456-2c96-450d-9d48-f5de3025e51c-xmJRDx'
```

Per Windows:

```
aws secretsmanager get-secret-value ^
    --secret-id 'arn:aws:secretsmanager:eu-west-1:123456789012:secret:rds!
cluster-033d7456-2c96-450d-9d48-f5de3025e51c-xmJRDx'
```

API RDS

È possibile visualizzare l'ARN, lo stato e la chiave KMS di un segreto gestito da RDS in Secrets Manager utilizzando l'operazione RDS DescribeDBClusters e impostando il parametro `DBClusterIdentifier` su un identificatore di cluster database. I dettagli del segreto sono inclusi nell'output.

Quando si dispone dell'ARN segreto, è possibile visualizzare i dettagli sul segreto utilizzando l'operazione GetSecretValueSecrets Manager.

Disponibilità di regioni e versioni

Il supporto varia a seconda delle versioni specifiche di ciascun motore di database e a seconda delle Regioni AWS. Per ulteriori informazioni sulla disponibilità di versioni e regioni con l'integrazione di Secrets Manager con Amazon RDS, consulta Regioni e motori DB supportati per l'integrazione di Secrets Manager con Amazon RDS.

Protezione dei dati in Amazon RDS

Il modello di responsabilità condivisa di AWS si applica alla protezione dei dati in Amazon Relational Database Service. Come descritto in questo modello, AWS è responsabile della protezione

dell'infrastruttura globale che esegue tutto l'Cloud AWS. L'utente è responsabile del controllo dei contenuti ospitati su questa infrastruttura. Inoltre, sei responsabile della configurazione della protezione e delle attività di gestione per i Servizi AWS che utilizzi. Per ulteriori informazioni sulla privacy dei dati, vedi le Domande frequenti sulla privacy dei dati. Per informazioni sulla protezione dei dati in Europa, consulta il post del blog relativo al Modello di responsabilità condivisa AWS e GDPR nel Blog sulla sicurezza AWS.

Per garantire la protezione dei dati, ti suggeriamo di proteggere le credenziali Account AWS e di configurare singoli utenti con AWS IAM Identity Center o AWS Identity and Access Management (IAM). In tal modo, a ogni utente verranno assegnate solo le autorizzazioni necessarie per svolgere i suoi compiti. Ti suggeriamo, inoltre, di proteggere i dati nei seguenti modi:

- Utilizza l'autenticazione a più fattori (MFA) con ogni account.
- Utilizza SSL/TLS per comunicare con le risorse AWS. È richiesto TLS 1.2 ed è consigliato TLS 1.3.
- Configura l'API e la registrazione delle attività degli utenti con AWS CloudTrail.
- Utilizza le soluzioni di crittografia AWS, insieme a tutti i controlli di sicurezza predefiniti in Servizi AWS.
- Utilizza i servizi di sicurezza gestiti avanzati, come Amazon Macie, che aiutano a individuare e proteggere i dati sensibili archiviati in Amazon S3.
- Se necessiti di moduli crittografici convalidati FIPS 140-2 quando accedi ad AWSattraverso un'interfaccia a riga di comando o un'API, utilizza un endpoint FIPS. Per ulteriori informazioni sugli endpoint FIPS disponibili, consulta il Federal Information Processing Standard (FIPS) 140-2.

Ti consigliamo vivamente di non inserire mai informazioni riservate o sensibili, ad esempio gli indirizzi e-mail dei clienti, nei tag o nei campi di testo in formato libero, ad esempio nel campo Nome. Questo suggerimento è relativo all'utilizzo di Amazon RDS o altri Servizi AWS tramite la console, l'API, AWS CLI o gli SDK AWS. I dati inseriti nei tag o nei campi di testo in formato libero utilizzati per i nomi possono essere utilizzati per i la fatturazione o i log di diagnostica. Quando fornisci un URL a un server esterno, ti suggeriamo vivamente di non includere informazioni sulle credenziali nell'URL per convalidare la tua richiesta al server.

Argomenti

- Protezione dei dati tramite crittografia
- Riservatezza del traffico Internet

Protezione dei dati tramite crittografia

Puoi abilitare la crittografia per le risorse di database. Puoi anche crittografare le connessioni ai cluster .

Argomenti

Crittografia delle risorse Amazon RDS

Amazon RDS può crittografare le istanze database di Amazon RDS. I dati che vengono crittografati quando sono inattivi includono lo storage sottostante per le istanze database, i backup automatici, le repliche di lettura e gli snapshot.

I cluster di Amazon RDS crittografate utilizzano l'algoritmo di crittografia AES-256 standard del settore per crittografare i dati sul server che ospita i cluster di Amazon RDS. Una volta crittografati i dati, Amazon RDS gestisce l'autenticazione dell'accesso e la decrittografia dei dati in modo trasparente con un impatto minimo sulle prestazioni. Non è quindi necessario modificare le applicazioni client di database per utilizzare la crittografia.

> **Note**
>
> Per i dati in transito tra le repliche di origine e quelle di lettura vengono crittografati, anche durante la replica tra regioni. AWS

Argomenti

- Limiti relativi a cluster di database crittografate Amazon RDS

Panoramica della crittografia delle risorse Amazon RDS

Le istanze database Amazon RDS crittografate offrono un livello aggiuntivo di sicurezza dei dati proteggendoli dagli accessi non autorizzati nello storage sottostante. Puoi utilizzare la crittografia Amazon RDS per aumentare la protezione dei dati delle applicazioni che vengono distribuite nel cloud e per soddisfare i requisiti di conformità per la crittografia dei dati inattivi.

Per un'istanza database crittografata con Amazon RDS, vengono crittografati tutti i log, i backup e gli snapshot. Amazon RDS utilizza an AWS KMS key per crittografare queste risorse. Per ulteriori informazioni sulle chiavi KMS, consulta AWS KMS keys nella Guida per sviluppatori di AWS Key Management Service e Gestione di AWS KMS key. Se copi uno snapshot crittografata, puoi utilizzare una chiave KMS diversa per crittografare la snapshot di destinazione rispetto a quella utilizzata per crittografare la snapshot di origine.

Una replica di lettura di un'istanza crittografata Amazon RDS deve essere crittografata utilizzando la stessa chiave KMS dell'istanza DB principale quando entrambe si trovano nella stessa regione. AWS Se l'istanza DB principale e la replica di lettura si trovano in AWS regioni diverse, si crittografa la replica di lettura utilizzando la chiave KMS per quella regione. AWS

È possibile utilizzare una o creare Chiave gestita da AWS chiavi gestite dal cliente. Per gestire le chiavi gestite dal cliente utilizzate per crittografare e decrittografare le risorse Amazon RDS, utilizza AWS Key Management Service (AWS KMS). AWS KMS combina hardware e software sicuri e a disponibilità elevata per offrire un sistema di gestione delle chiavi a misura di cloud. Utilizzando AWS KMS, è possibile creare chiavi gestite dal cliente e definire le politiche che controllano il modo in cui tali chiavi gestite dal cliente possono essere utilizzate. AWS KMS supporta CloudTrail, in modo da poter controllare l'utilizzo delle chiavi KMS per verificare che le chiavi gestite dal cliente vengano utilizzate in modo appropriato. Puoi utilizzare le chiavi gestite dai clienti con Amazon Aurora e AWS servizi supportati come Amazon S3, Amazon EBS e Amazon Redshift. Per un elenco dei servizi integrati con AWS KMS, consulta Service Integration. AWS

Amazon RDS supporta anche la crittografia di un'istanza database di Oracle o SQL Server con Transparent Data Encryption (TDE). TDE può essere utilizzata con la crittografia RDS inattiva, sebbene l'utilizzo simultaneo di TDE e della crittografia RDS inattiva possa influire leggermente sulle prestazioni del database. È necessario gestire chiavi diverse per ogni metodo di crittografia. Per ulteriori informazioni su TDE, consulta Oracle Transparent Data Encryption o Supporto per Transparent Data Encryption in SQL Server.

Crittografia di un'istanza database

Per abilitare la crittografia per una nuova istanza database, scegliere Enable encryption (Abilita crittografia) nella console Amazon RDS. Per ulteriori informazioni sulla creazione di un istanza database, consulta Creazione di un'istanza database Amazon RDS.

Se usi il create-db-instance AWS CLI comando per creare un'istanza DB crittografata, imposta il `--storage-encrypted` parametro. Se utilizzi l'operazione API CreateDBInstance, imposta il parametro `StorageEncrypted` su true.

Quando crei un'istanza database crittografata, puoi scegliere una chiave gestita dal cliente o la Chiave gestita da AWS per Amazon RDS per la crittografia dell'istanza database. Se non specifichi l'identificatore di chiave per una chiave gestita dal cliente, Amazon RDS lo utilizza Chiave gestita da AWS per la tua nuova istanza DB. Amazon RDS crea un Chiave gestita da AWS account per Amazon RDS. AWS Il tuo AWS account ha un account Amazon RDS diverso Chiave gestita da AWS per ogni AWS regione.

Per ulteriori informazioni sulle chiavi KMS, consulta AWS KMS keys nella Guida per gli sviluppatori di AWS Key Management Service .

Una volta creata un'istanza database crittografata, non potrai più modificare la chiave KMS utilizzata da quell'istanza database. Pertanto, assicurati di determinare i requisiti della chiave KMS prima di creare la tua istanza database crittografata.

Se utilizzi il AWS CLI `create-db-instance` comando per creare un'istanza DB crittografata con una chiave gestita dal cliente, imposta il `--kms-key-id` parametro su qualsiasi identificatore di chiave per la chiave KMS. Se utilizzi la funzionalità `CreateDBInstance` dell'API Amazon RDS, imposta il parametro `KmsKeyId` su un qualsiasi identificatore chiave per la chiave KMS. Per utilizzare una chiave gestita dal cliente in un diverso account AWS , specifica l'ARN della chiave o dell'alias.

> **Important**
>
> Amazon RDS può perdere l'accesso alla chiave KMS per un'istanza database. Ad esempio, RDS perde l'accesso quando la chiave KMS non è abilitata o quando l'accesso di RDS a una chiave KMS è stato revocato. In questi casi, l'istanza database crittografata entra nello stato `inaccessible-encryption-credentials-recoverable`. L'istanza database rimane in questo stato per sette giorni. Quando si avvia l'istanza database durante tale periodo, verifica se la chiave KMS è attiva e la recupera in caso affermativo. Riavvia l'istanza DB utilizzando il AWS CLI comando start-db-instanceo. AWS Management Console

Se la chiave KMS non viene recuperata, l'istanza database crittografata entra nello stato terminale `inaccessible-encryption-credentials`. In questo caso, è possibile solo ripristinare l'istanza database da un backup. È consigliabile abilitare sempre i backup per le istanze database crittografate per evitare la perdita di dati crittografati nei database.

Determinare se la crittografia è attivata per un'istanza database

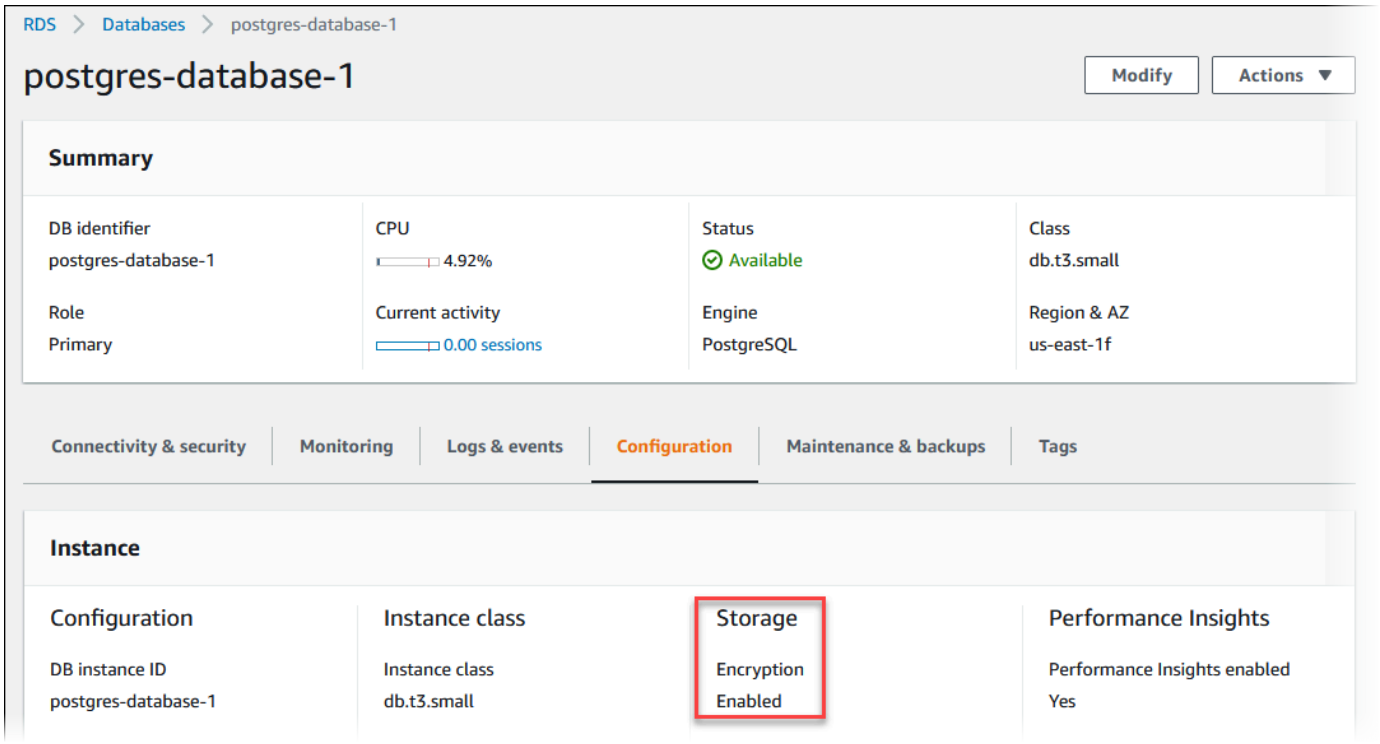
È possibile utilizzare l'API AWS Management Console AWS CLI, o RDS per determinare se la crittografia a riposo è attivata per un'istanza DB.

Console

Per determinare se la crittografia a riposo è attivata per un'istanza database

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel pannello di navigazione, scegliere Databases (Database).
3. Scegliere il nome dell'istanza database a cui si desidera controllare per visualizzarne i dettagli.
4. Seleziona la casella Configurazione, e controlla il valore Crittografia sottoStorage (archiviazione).

   Mostra Enabled (Abilitato) o Non abilitato.



AWS CLI

Per determinare se la crittografia a riposo è attivata per un'istanza DB utilizzando il AWS CLI, chiama il describe-db-instancescomando con la seguente opzione:

- `--db-instance-identifier` – Il nome dell'istanza database.

Nell'esempio seguente viene utilizzata una query per restituire `TRUE` o `FALSE` per quanto riguarda la crittografia inattiva per l'istanza database `mydb`.

Example

```
aws rds describe-db-instances --db-instance-identifier mydb --query "*[].
{StorageEncrypted:StorageEncrypted}" --output text
```

API RDS

Per determinare se la crittografia dei dati inattivi per un'istanza database utilizza l'API Amazon RDS, chiamare l'operazione DescribeDBInstancescon il parametro seguente:

- `DBInstanceIdentifier` – Il nome dell'istanza database.

Disponibilità della crittografia Amazon RDS

La crittografia Amazon RDS è attualmente disponibile per tutti i motori di database e i tipi di archiviazione, eccetto SQL Server Express Edition.

La crittografia Amazon RDS è disponibile per la maggior parte delle classi di istanza database. Nella tabella seguente sono elencate le classi di istanza database che non supportano la crittografia Amazon RDS:

Tipo di istanza	Classe istanza
General purpose (M1)	db.m1.small
	db.m1.medium
	db.m1.large
	db.m1.xlarge

Tipo di istanza	Classe istanza
Memoria ottimizzata (M2)	db.m2.xlarge db.m2.2xlarge db.m2.4xlarge
Burstable (T2)	db.t2.micro

Crittografia in transito

AWS fornisce una connettività sicura e privata tra istanze DB di tutti i tipi. Inoltre, alcuni tipi di istanza utilizzano le funzionalità di offload dell'hardware Nitro System sottostante per crittografare automaticamente il traffico in transito tra le istanze. Questa crittografia utilizza algoritmi AEAD (Authenticated Encryption with Associated Data), con crittografia a 256 bit. Non vi è alcun impatto sulle prestazioni della rete. Per supportare questa crittografia aggiuntiva del traffico in transito tra istanze, è necessario soddisfare i seguenti requisiti:

- Le istanze utilizzano i seguenti tipi di istanza:
  - Scopo generale: M6i, M6id, M6in, M6idn, M7g
  - Memoria ottimizzata: R6i, R6id, R6in, R6idn, R7g, X2idn, X2iEdn, X2iEzn
- Le Regione AWS istanze sono le stesse.
- Le istanze si trovano nello stesso VPC o VPC con peering e il traffico non passa attraverso un dispositivo di rete virtuale, ad esempio un load balancer (load balancer) o un Transit Gateway.

Limiti relativi a cluster di database crittografate Amazon RDS

Esistono le seguenti limitazioni per i cluster di database crittografate Amazon RDS:

- Puoi solo crittografare un'istanza database Amazon RDS quando la crei, non dopo la sua creazione.

  Tuttavia, poiché è possibile crittografare una copia di uno snapshot DB non crittografata, puoi aggiungere in modo efficace la crittografia a un'istanza database non crittografata. Ovvero, è possibile creare uno snapshot dell'istanza database e quindi creare una copia crittografata di quella snapshot. Puoi quindi ripristinare un'istanza database da uno snapshot crittografata e pertanto

disporre di una copia crittografata dell'istanza database originale. Per ulteriori informazioni, consulta Copia di una snapshot DB..

- Non puoi disattivare la crittografia di una istanza database crittografato.
- Non puoi creare uno snapshot crittografata per una istanza database non crittografato.
- Una snapshot di una istanza database crittografato deve essere crittografata utilizzando la stessa chiave KMS dell'istanza database.
- Non è possibile creare una replica di lettura crittografata di un'istanza database non crittografata o una replica di lettura non crittografata di un'istanza database crittografata.
- Le repliche di lettura crittografate devono essere crittografate con la stessa chiave KMS dell'istanza DB di origine quando entrambe si trovano nella stessa regione. AWS
- Non puoi ripristinare un backup o uno snapshot non crittografato in un'istanza database crittografata.
- Per copiare un'istantanea crittografata da una AWS regione all'altra, è necessario specificare la chiave KMS nella regione di destinazione. AWS Questo perché le chiavi KMS sono specifiche della AWS regione in cui vengono create.

  La snapshot di origine resta crittografata nel processo di copia. Amazon RDS utilizza la crittografia envelope per proteggere i dati durante il processo di copia. Per ulteriori informazioni sulla crittografia envelope, consulta Crittografia envelope nella Guida per sviluppatori di AWS Key Management Service .
- Non è possibile decrittografare una istanza database crittografato. Tuttavia, puoi esportare i dati da una istanza database crittografato e importarli in una istanza database non crittografato.

Gestione di AWS KMS key

Amazon RDS si integra automaticamente con AWS Key Management Service (AWS KMS) per la gestione delle chiavi. Amazon RDS utilizza la crittografia envelope. Per ulteriori informazioni sulla crittografia envelope, consulta Crittografia envelope nella Guida per sviluppatori di AWS Key Management Service.

È possibile utilizzare due tipi di chiavi AWS KMS per crittografare le istanze database.

- Per avere il pieno controllo su una chiave KMS, devi creare una chiave gestita dal cliente. Per ulteriori informazioni sulle chiavi gestite dal cliente, consulta Chiavi gestite dal cliente nella Guida per gli sviluppatori di AWS Key Management Service.

Non puoi condividere uno snapshot che è stata crittografata con la Chiave gestita da AWS dell'account AWS che ha condiviso la snapshot.

- Le Chiavi gestite da AWS sono chiavi KMS nel tuo account create, gestite e utilizzate a tuo nome da un servizio AWS che si integra con AWS KMS. Per impostazione predefinita, la Chiave gestita da AWS RDS (`aws/rds`) viene utilizzata per la crittografia. Non è possibile gestire, ruotare o eliminare la Chiave gestita da AWS RDS. Per ulteriori informazioni su Chiavi gestite da AWS, consulta Chiavi gestite da AWS nella Guida per gli sviluppatori di AWS Key Management Service.

Puoi gestire le chiavi KMS utilizzate per le istanze database di Amazon RDS tramite AWS Key Management Service (AWS KMS) nella console AWS KMS, la AWS CLI o l'API AWS KMS. Puoi visualizzare i log di controllo di ogni operazione eseguita con una chiave gestita da AWS o dal cliente utilizzando AWS CloudTrail. Per ulteriori informazioni sulla rotazione delle chiavi, consulta Rotazione delle chiavi AWS KMS.

> **Important**
>
> Se si disattivano o revocano le autorizzazioni per una chiave KMS utilizzata da un database RDS, RDS inserisce il database in uno stato terminale quando è richiesto l'accesso alla chiave KMS. Questa modifica potrebbe essere immediata o differita, a seconda del caso d'uso che richiedeva l'accesso alla chiave KMS. In questo stato, l'istanza database non è più disponibile e lo stato attuale del database non può essere ripristinato. Per ripristinare l'istanza database, devi riabilitare l'accesso alla chiave KMS per RDS e ripristinare l'istanza database dall'ultimo backup disponibile.

Autorizzazione dell'uso di una chiave gestita dal cliente

Quando RDS utilizza una chiave gestita dal cliente in operazioni che coinvolgono la crittografia, funziona per conto dell'utente che crea o modifica la risorsa RDS.

Per creare una risorsa RDS utilizzando una chiave gestita dal cliente, un utente deve disporre delle autorizzazioni per richiamare le seguenti operazioni su tale chiave:

- kms:CreateGrant
- kms:DescribeKey

Puoi specificare queste autorizzazioni necessarie in una policy chiave o in una policy IAM se la policy chiave lo consente.

Esistono diversi modi per rendere la policy IAM più efficace. Ad esempio, se desideri consentire l'uso della chiave gestita dal cliente solo per le richieste che provengono da RDS, puoi utilizzare la chiave di condizione kms:ViaService con il valore `rds.<region>.amazonaws.com`. Puoi inoltre usare le chiavi o i valori nel Contesto di crittografia di Amazon RDS come condizione per utilizzare la chiave gestita dal cliente per la crittografia.

Per ulteriori informazioni, consulta Autorizzazione per gli utenti in altri account di utilizzare una chiave KMS nella Guida per gli sviluppatori di AWS Key Management Service e Policy delle chiavi in AWS KMS.

Contesto di crittografia di Amazon RDS

Quando RDS utilizza la chiave KMS o quando Amazon EBS utilizza la chiave KMS per conto di RDS, il servizio specifica un contesto di crittografia. Il contesto di crittografia rappresenta dati autenticati supplementari (AAD) utilizzati da AWS KMS per garantire l'integrità dei dati. Quando viene specificato un contesto di crittografia per un'operazione di crittografia, il servizio deve specificare lo stesso contesto di crittografia per l'operazione di decrittografia. In caso contrario, la decrittografia ha esito negativo. Il contesto di crittografia viene scritto nei log AWS CloudTrail per aiutarti a comprendere perché è stata utilizzata una determinata chiave KMS. I log CloudTrail potrebbero contenere molte voci che descrivono l'utilizzo di una chiave KMS, ma il contesto di crittografia in ciascuna voce di log può aiutarti a determinare il motivo per quel particolare uso.

Come minimo, Amazon RDS utilizza sempre l'ID dell'istanza database per il contesto di crittografia, come nel seguente esempio in formato JSON:

```
{ "aws:rds:db-id": "db-CQYSMDPBRZ7BPMH7Y3RTDG5QY" }
```

Questo contesto di crittografia può aiutarti a identificare l'istanza database per la quale è stata utilizzata la tua chiave KMS.

Quando la tua chiave KMS viene utilizzata per un'istanza database specifica e un determinato volume Amazon EBS, sia l'ID dell'istanza database e l'ID del volume Amazon EBS vengono utilizzati per il contesto di crittografia, come nel seguente esempio in formato JSON:

```
{
  "aws:rds:db-id": "db-BRG7VYS3SVIFQW7234EJQOM5RQ",
  "aws:ebs:id": "vol-ad8c6542"
```

```
}
```

Puoi utilizzare Secure Socket Layer (SSL) o Transport Layer Security (TLS) dalla tua applicazione per crittografare una connessione a un database che esegue Db2, MariaDB, Microsoft SQL Server, MySQL, Oracle o PostgreSQL.

Facoltativamente, la connessione SSL/TLS può eseguire la verifica dell'identità del server convalidando il certificato del server installato nel database. Per richiedere la verifica dell'identità del server, esegui questa procedura generale:

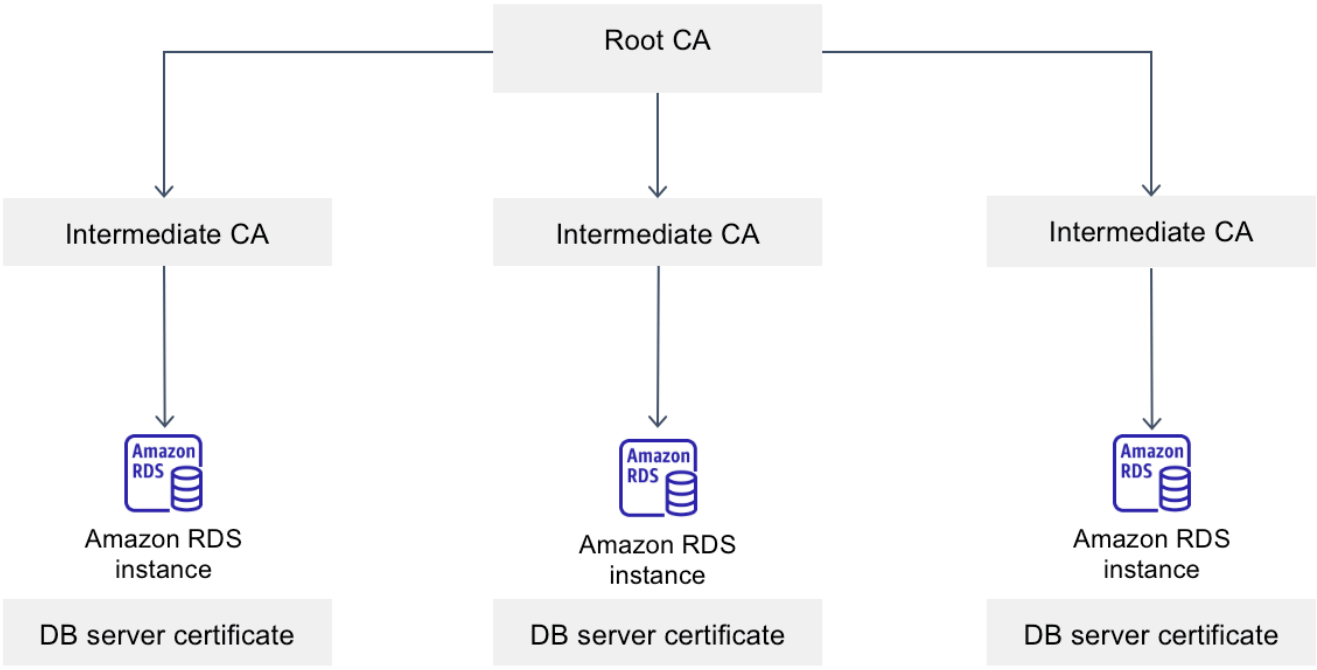
1. Scegli l'autorità di certificazione (CA) che firma il certificato del server di database per il database. Per ulteriori informazioni sulle autorità di certificazione, consulta Autorità di certificazione.
2. Scarica un bundle di certificati da utilizzare quando ti connetti al database. Per scaricare un bundle di certificati, consulta Pacchetti di certificati per tutti Regioni AWS e Pacchetti di certificati per scopi specifici Regioni AWS.

   > **Note**
   >
   > Tutti i certificati sono disponibili solo per il download tramite connessioni SSL/TLS.

3. Connettiti al database utilizzando il processo del motore di database per l'implementazione delle connessioni SSL/TLS. Ciascun motore database ha il proprio processo per l'implementazione di SSL/TLS. Per informazioni su come implementare SSL/TLS per il database, usa il collegamento corrispondente al motore di database in uso:
   - Utilizzo di SSL/TLS con un'istanza DB RDS per Db2
   - Utilizzo di SSL/TLS con un'istanza database MariaDB
   - Utilizzo di SSL con un'istanza database Microsoft SQL Server
   - Utilizzo di SSL/TLS con un'istanza database MySQL
   - Utilizzo di SSL con un'istanza database RDS per Oracle
   - Utilizzo del protocollo SSL con un'istanza database PostgreSQL

Autorità di certificazione

L'autorità di certificazione (CA) è il certificato che identifica la CA root della catena di certificati. La CA firma il certificato del server di database, che è il certificato del server installato su ogni istanza database. Il certificato del server di database identifica l'istanza database come server attendibile.



Amazon RDS fornisce le seguenti CA per firmare il certificato del server DB per un database.

Autorità di certificazione (CA)	Descrizione
rds-ca-2019	Utilizza un'autorità di certificazione con l'algoritmo a chiave privata RSA 2048 e l'algoritmo di firma SHA256. Questa CA scade nel 2024 e non supporta la rotazione automatica dei certificati del server. Se utilizzi questa CA e desideri mantenere lo stesso standard, ti consigliamo di passare alla CA rds-ca-rsa 2048-g1.
rds-ca-rsa2048-g1	Utilizza un'autorità di certificazione con l'algoritmo a chiave privata RSA 2048 e l'algoritmo di firma SHA256 nella maggior parte delle Regioni AWS. Nel AWS GovCloud (US) Regions, questa CA utilizza un'autorità di certificazione con algoritmo a chiave privata RSA 2048 e algoritmo di firma SHA384.

Autorità di certificazione (CA)	Descrizione
	Questa CA rimane valida più a lungo della CA rds-ca-2019 e supporta la rotazione automatica dei certificati del server.
rds-ca-rsa4096-g1	Utilizza un'autorità di certificazione con l'algoritmo a chiave privata RSA 4096 e l'algoritmo di firma SHA384. supporta la rotazione automatica dei certifica ti del server.
rds-ca-ecc384-g1	Utilizza un'autorità di certificazione con l'algoritmo a chiave privata ECC 384 e l'algoritmo di firma SHA384. supporta la rotazione automatica dei certificati del server.

> **Note**
>
> Se utilizzi il AWS CLI, puoi vedere le validità delle autorità di certificazione sopra elencate utilizzando describe-certificates.

Questi certificati CA sono inclusi nel bundle di certificati regionali e globali. Quando si utilizza la CA rds-ca-rsa 2048-g1, rds-ca-rsa 4096-g1 o rds-ca-ecc 384-g1 con un database, RDS gestisce il certificato del server DB sul database. RDS esegue automaticamente la rotazione del certificato del server di database prima della scadenza.

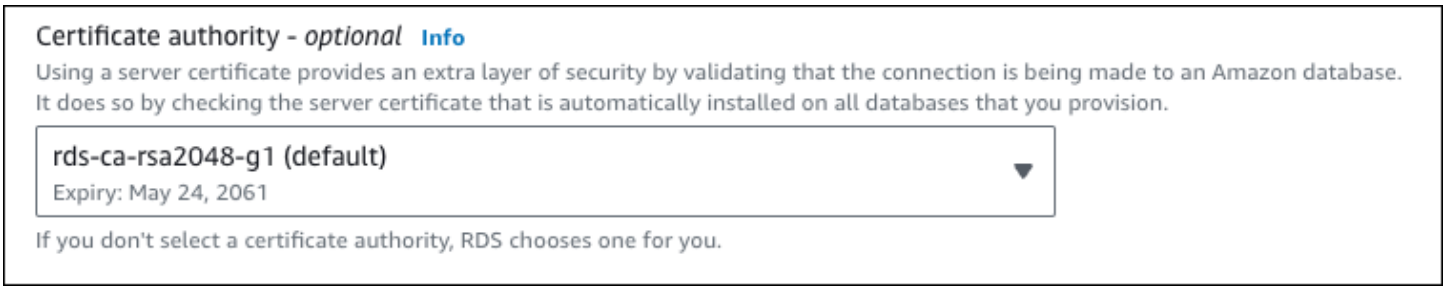
Impostazione della CA per il database

Puoi impostare la CA per un database quando esegui le seguenti attività:

- Crea un'istanza DB o un cluster DB Multi-AZ: puoi impostare la CA quando crei un'istanza o un cluster DB. Per istruzioni, consulta the section called “Creazione di un'istanza database” o the section called “Creazione di un cluster di database Multi-AZ”.
- Modifica un'istanza DB o un cluster DB Multi-AZ: è possibile impostare la CA per un'istanza o un cluster DB modificandola. Per istruzioni, consulta the section called “Modifica di un'istanza database” o the section called “Modifica di un cluster di database Multi-AZ”.

> **Note**
>
> La CA predefinita è impostata su 2048-g1. rds-ca-rsa È possibile sovrascrivere la CA predefinita per il proprio Account AWS utilizzando il comando modify-certificates.

Le CA disponibili dipendono dal motore di database e dalla versione del motore di database. Quando si utilizza la AWS Management Console, è possibile scegliere la CA usando l'impostazione Certificate authority (Autorità di certificazione), come mostrato nell'immagine seguente.



La console mostra solo le CA disponibili per il motore di database e la versione del motore di database. Se si utilizza il AWS CLI, è possibile impostare la CA per un'istanza DB utilizzando il create-db-instancecomando or. modify-db-instance È possibile impostare la CA per un cluster DB Multi-AZ utilizzando il modify-db-clustercomando create-db-clusteror.

Se utilizzi il AWS CLI, puoi vedere le CA disponibili per il tuo account utilizzando il comando describe-certificates. Questo comando mostra nell'output anche la data di scadenza per ogni CA in `ValidTill`. Puoi trovare le CA disponibili per uno specifico motore DB e una versione del motore DB utilizzando il comando. describe-db-engine-versions

L'esempio seguente mostra le CA disponibili per la versione predefinita del motore di database RDS per PostgreSQL.

```
aws rds describe-db-engine-versions --default-only --engine postgres
```

L'output è simile a quello riportato di seguito. Le CA disponibili sono elencate in `SupportedCACertificateIdentifiers`. L'output mostra anche se la versione del motore di database supporta la rotazione del certificato senza riavvio in `SupportsCertificateRotationWithoutRestart`.

```
{
    "DBEngineVersions": [
```

```
        {
            "Engine": "postgres",
            "MajorEngineVersion": "13",
            "EngineVersion": "13.4",
            "DBParameterGroupFamily": "postgres13",
            "DBEngineDescription": "PostgreSQL",
            "DBEngineVersionDescription": "PostgreSQL 13.4-R1",
            "ValidUpgradeTarget": [],
            "SupportsLogExportsToCloudwatchLogs": false,
            "SupportsReadReplica": true,
            "SupportedFeatureNames": [
                "Lambda"
            ],
            "Status": "available",
            "SupportsParallelQuery": false,
            "SupportsGlobalDatabases": false,
            "SupportsBabelfish": false,
            "SupportsCertificateRotationWithoutRestart": true,
            "SupportedCACertificateIdentifiers": [
                "rds-ca-2019",
                "rds-ca-rsa2048-g1",
                "rds-ca-ecc384-g1",
                "rds-ca-rsa4096-g1"
            ]
        }
    ]
}
```

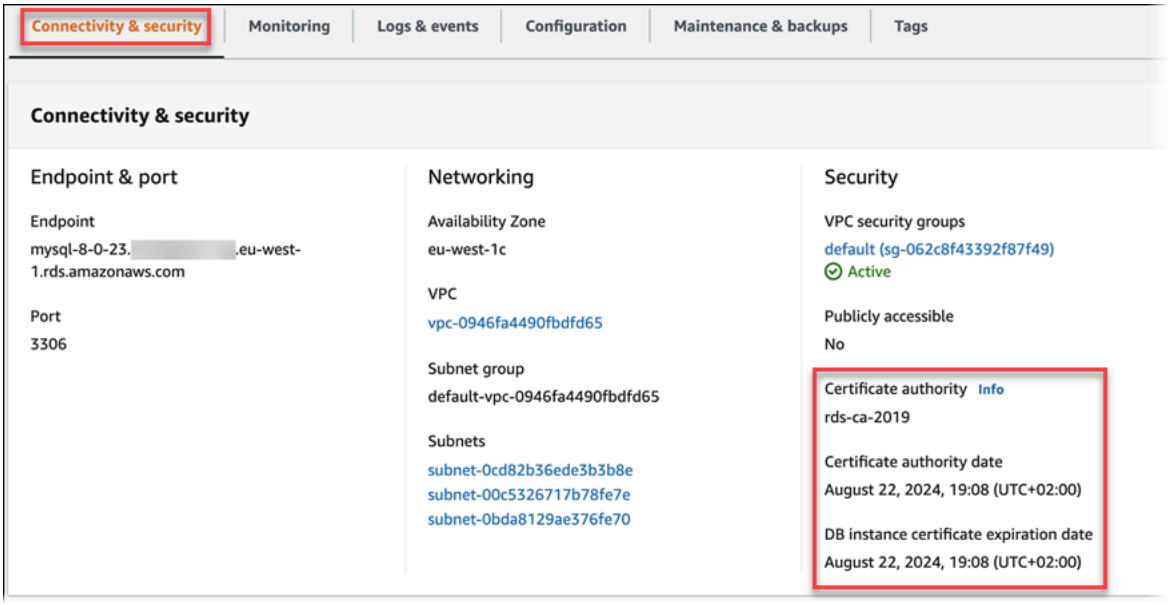
Validità dei certificati del server di database

La validità del certificato del server di database dipende dal motore di database e dalla versione del motore di database. Se la versione del motore di database supporta la rotazione del certificato senza riavvio, la validità del certificato del server di database è di 1 anno. In caso contrario, la validità è di 3 anni.

Per ulteriori informazioni sulla rotazione dei certificati del server di database, consulta Rotazione automatica dei certificati del server.

Visualizzazione della CA per l'istanza DB

È possibile visualizzare i dettagli sulla CA di un database visualizzando la scheda Connettività e sicurezza nella console, come nell'immagine seguente.



Se si utilizza il AWS CLI, è possibile visualizzare i dettagli sulla CA per un'istanza DB utilizzando il describe-db-instancescomando. È possibile visualizzare i dettagli sulla CA per un cluster DB Multi-AZ utilizzando il describe-db-clusterscomando.

Per verificare il contenuto del bundle di certificati CA, utilizza il comando seguente:

```
keytool -printcert -v -file global-bundle.pem
```

Pacchetti di certificati per tutti Regioni AWS

Per ottenere un pacchetto di certificati per tutti Regioni AWS, scaricalo da https://truststore.pki.rds.amazonaws.com/global/global-bundle.pem.

Il pacchetto contiene sia i certificati `rds-ca-2019` intermedi che quelli root. Il pacchetto contiene anche i certificati `rds-ca-rsa2048-g1` CA `rds-ca-rsa4096-g1` e `rds-ca-ecc384-g1` root. L'application trust store deve solo registrare il certificato CA principale.

Se l'applicazione è su Microsoft Windows e richiede un file PKCS7, è possibile scaricare il pacchetto di certificati PKCS7 da https://truststore.pki.rds.amazonaws.com/global/global-bundle.p7b.

> **Note**
>
> Il proxy i certificati di AWS Certificate Manager (ACM). Se utilizzi RDS Proxy, non è necessario scaricare certificati Amazon RDS o aggiornare applicazioni che utilizzano

connessioni proxy RDS. Per ulteriori informazioni, consulta Utilizzo di TLS/SSL con RDS Proxy.

Pacchetti di certificati per scopi specifici Regioni AWS

Il pacchetto contiene sia i certificati `rds-ca-2019` intermedi che quelli root. Il pacchetto contiene anche i certificati `rds-ca-rsa2048-g1` CA `rds-ca-rsa4096-g1` e `rds-ca-ecc384-g1` root. L'application trust store deve solo registrare il certificato CA principale.

Per ottenere un pacchetto di certificati per un Regione AWS, scaricalo dal link riportato Regione AWS nella tabella seguente.

AWS Region	Bundle di certificati (PEM)	Bundle di certificati (PKCS7)
Stati Uniti orientali (Virginia settentrionale)	us-east-1-bundle.pem	us-east-1-bundle.p7b
US East (Ohio)	us-east-2-bundle.pem	us-east-2-bundle.p7b
US West (N. California)	us-west-1-bundle.pem	us-west-1-bundle.p7b
US West (Oregon)	us-west-2-bundle.pem	us-west-2-bundle.p7b
Africa (Cape Town)	af-south-1-bundle.pem	af-sud-1-bundle.p7b
Asia Pacific (Hong Kong)	ap-east-1-bundle.pem	ap-east-1-bundle.p7b
Asia Pacific (Hyderabad)	ap-south-2-bundle.pem	ap-south-2-bundle.p7b
Asia Pacifico (Giacarta)	ap-southeast-3-bundle.pem	ap-southeast-3-bundle.p7b
Asia Pacifico (Melbourne)	ap-southeast-4-bundle.pem	ap-southeast-4-bundle.p7b
Asia Pacifico (Mumbai)	ap-south-1-bundle.pem	ap-south-1-bundle.p7b
Asia Pacific (Osaka)	ap-northeast-3-bundle.pem	ap-northeast-3-bundle.p7b
Asia Pacific (Tokyo)	ap-northeast-1-bundle.pem	ap-northeast-1-bundle.p7b
Asia Pacific (Seoul)	ap-northeast-2-bundle.pem	ap-northeast-2-bundle.p7b

AWS Region	Bundle di certificati (PEM)	Bundle di certificati (PKCS7)
Asia Pacific (Singapore)	ap-southeast-1-bundle.pem	ap-southeast-1-bundle.p7b
Asia Pacific (Sydney)	ap-southeast-2-bundle.pem	ap-southeast-2-bundle.p7b
Canada (Central)	ca-central-1-bundle.pem	ca-central-1-bundle.p7b
Canada occidentale (Calgary)	ca-west-1-bundle.pem	ca-west-1-bundle.p7b
Europa (Francoforte)	eu-central-1-bundle.pem	eu-central-1-bundle.p7b
Europe (Ireland)	eu-west-1-bundle.pem	eu-west-1-bundle.p7b
Europe (London)	eu-west-2-bundle.pem	eu-west-2-bundle.p7b
Europe (Milan)	eu-south-1-bundle.pem	eu-sud-1-bundle.p7b
Europe (Paris)	eu-west-3-bundle.pem	eu-west-3-bundle.p7b
Europa (Spagna)	eu-south-2-bundle.pem	eu-south-2-bundle.p7b
Europa (Stoccolma)	eu-nord-1-bundle.pem	eu-nord-1-bundle.p7b
Europa (Zurigo)	eu-central-2-bundle.pem	eu-central-2-bundle.p7b
Israele (Tel Aviv)	il-central-1-bundle.pem	il-central-1-bundle.p7b
Middle East (Bahrain)	me-sud-1-bundle.pem	me-sud-1-bundle.p7b
Medio Oriente (Emirati Arabi Uniti)	me-central-1-bundle.pem	me-central-1-bundle.p7b
Sud America (San Paolo)	sa-east-1-bundle.pem	sa-east-1-bundle.p7b

Certificati AWS GovCloud (US)

Per ottenere un pacchetto di certificati che contenga sia i certificati intermedi che i certificati root per il sistema AWS GovCloud (US) Region s, scaricalo da https://truststore.pki. us-gov-west-1.rds.amazonaws.com/global/global-bundle.pem.

Se l'applicazione è su Microsoft Windows e richiede un file PKCS7, è possibile scaricare il pacchetto di certificati PKCS7 da https://truststore.pki. us-gov-west-1.rds.amazonaws.com/global/global-bundle.p7b.

Il pacchetto contiene sia i certificati intermedi che quelli root. `rds-ca-2019` Il pacchetto contiene anche i certificati `rds-ca-rsa2048-g1` CA `rds-ca-rsa4096-g1` e `rds-ca-ecc384-g1` root. L'application trust store deve solo registrare il certificato CA principale.

Per ottenere un pacchetto di certificati per un AWS GovCloud (US) Region, scaricalo dal link riportato AWS GovCloud (US) Region nella tabella seguente.

AWS GovCloud (US) Region	Bundle di certificati (PEM)	Bundle di certificati (PKCS7)
AWS GovCloud (Stati Uniti orientali)	us-gov-east-1 bundle.pem	us-gov-east-1 pacchetto.p7b
AWS GovCloud (Stati Uniti occidentali)	us-gov-west-1 bundle.pem	us-gov-west-1 pacchetto.p7b

Rotazione del certificato SSL/TLS

I certificati dell'autorità di certificazione (CA) Amazon RDS rds-ca-2019 scadono ad agosto 2024. Se utilizzi o prevedi di utilizzare Secure Sockets Layer (SSL) o Transport Layer Security (TLS) con verifica del certificato per connetterti alle tue istanze DB RDS o ai cluster DB Multi-AZ, prendi in considerazione l'utilizzo di uno dei nuovi certificati CA 2048-g1, 4096-g1 o 384-g1. rds-ca-rsa rds-ca-rsa rds-ca-ecc Se attualmente non usi SSL/TLS con la verifica del certificato, è possibile che un certificato CA sia scaduto e che sia necessario aggiornarlo al nuovo certificato CA se prevedi di utilizzare SSL/TLS con la verifica del certificato per connetterti ai database RDS.

Segui queste istruzioni per completare gli aggiornamenti. Prima di aggiornare le istanze DB o i cluster DB Multi-AZ per utilizzare il nuovo certificato CA, assicurati di aggiornare i client o le applicazioni che si connettono ai database RDS.

Amazon RDS fornisce nuovi certificati CA come best practice di AWS sicurezza. Per informazioni sui nuovi certificati e sulle AWS regioni supportate, consulta.

> **Note**
>
> Il proxy i certificati di AWS Certificate Manager (ACM). Se utilizzi il proxy RDS, quando ruoti il certificato SSL/TLS, non devi aggiornare le applicazioni che utilizzano connessioni proxy RDS. Per ulteriori informazioni, consulta Utilizzo di TLS/SSL con RDS Proxy.

> **Note**
>
> Se utilizzi un'applicazione Go versione 1.15 con un'istanza DB o un cluster DB Multi-AZ creato o aggiornato al certificato rds-ca-2019 prima del 28 luglio 2020, devi aggiornare nuovamente il certificato. Esegui il `modify-db-instance` comando per un'istanza DB o il comando per un cluster DB Multi-AZ, utilizzando il nuovo identificatore di certificato CA. `modify-db-cluster` È possibile trovare le CA disponibili per un motore di database e una versione del motore di database specifici utilizzando il comando `describe-db-engine-versions`.
> Se hai creato il database o aggiornato il relativo certificato dopo il 28 luglio 2020, non è richiesta alcuna azione. Per ulteriori informazioni, consulta il GitHub numero #39568 di Go.

Argomenti

- Aggiornamento del certificato CA modificando l'istanza o il cluster di database
- Aggiornamento del certificato CA mediante l'applicazione di manutenzione
- Rotazione automatica dei certificati del server
- Script di esempio per l'importazione di certificati nel tuo archivio di trust

Aggiornamento del certificato CA modificando l'istanza o il cluster di database

L'esempio seguente aggiorna il certificato CA da rds-ca-2019 a 2048-g1. rds-ca-rsa Puoi scegliere un certificato diverso. Per ulteriori informazioni, consulta Autorità di certificazione.

Per aggiornare il certificato CA modificando l'istanza o il cluster di database

1. Scaricare il nuovo certificato SSL/TLS come descritto in .
2. Aggiornare le applicazioni per utilizzare il nuovo certificato SSL/TLS.

I metodi per l'aggiornamento delle applicazioni per i nuovi certificati SSL/TLS dipendono dalle applicazioni specifiche in uso. Collaborare con gli sviluppatori dell'applicazione per aggiornare i certificati SSL/TLS per le applicazioni.

Per maggiori informazioni sulla verifica delle connessioni SSL/TLS e sull'aggiornamento delle applicazioni per ciascun motore DB, consulta i seguenti argomenti:

- Aggiornamento delle applicazioni per la connessione a istanze database MariaDB mediante i nuovi certificati SSL/TLS
- Aggiornamento delle applicazioni per la connessione a istanze di database Microsoft SQL Server utilizzando nuovi certificati SSL/TLS
- Aggiornamento delle applicazioni per la connessione a istanze database MySQL mediante nuovi certificati SSL/TLS
- Aggiornamento delle applicazioni per la connessione a istanze database Oracle mediante nuovi certificati SSL/TLS
- Aggiornamento delle applicazioni per la connessione a istanze database PostgreSQL mediante nuovi certificati SSL/TLS.

Per uno script di esempio che aggiorna un archivio di trust in un sistema operativo Linux, vedi Script di esempio per l'importazione di certificati nel tuo archivio di trust.

> **Note**
>
> Il bundle di certificati contiene certificati per la vecchia e la nuova CA, pertanto puoi aggiornare l'applicazione in modo sicuro e mantenere la connettività durante il periodo di transizione. Se utilizzi il AWS Database Migration Service per migrare un database verso un'istanza DB o un cluster di , ti consigliamo di utilizzare il pacchetto di certificati per garantire la connettività durante la migrazione.

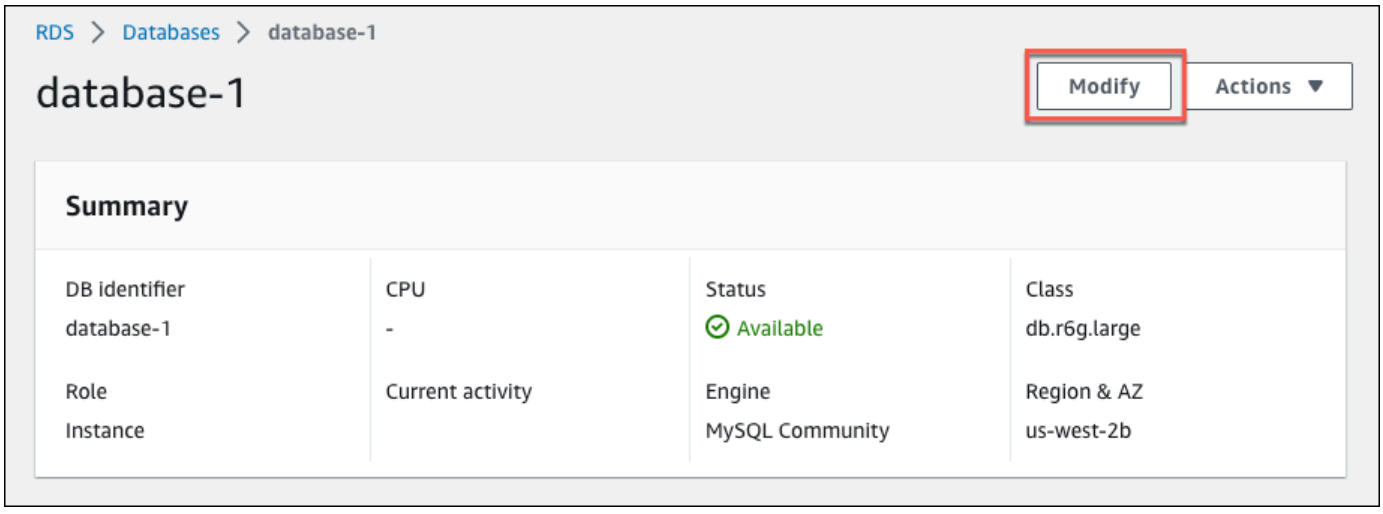
3. Modifica l'istanza DB o il cluster DB Multi-AZ per cambiare la CA da rds-ca-2019 a 2048-g1. rds-ca-rsa Per verificare se il database richiede un riavvio per aggiornare i certificati CA, utilizza il comando e seleziona il flag. describe-db-engine-versions`SupportsCertificateRotationWithoutRestart`

> **Important**
>
> Se si verificano problemi di connettività dopo la scadenza del certificato, utilizzare l'opzione Applica immediatamente specificando Apply immediately (Applica immediatamente) nella console o specificando l'opzione `--apply-immediately` mediante AWS CLI. Per impostazione predefinita, questa operazione è pianificata per l'esecuzione durante la prossima finestra di manutenzione.
> Per impostare una sostituzione della CA per la tua istanza diversa dalla CA RDS predefinita, usa il comando CLI modify-certificates.

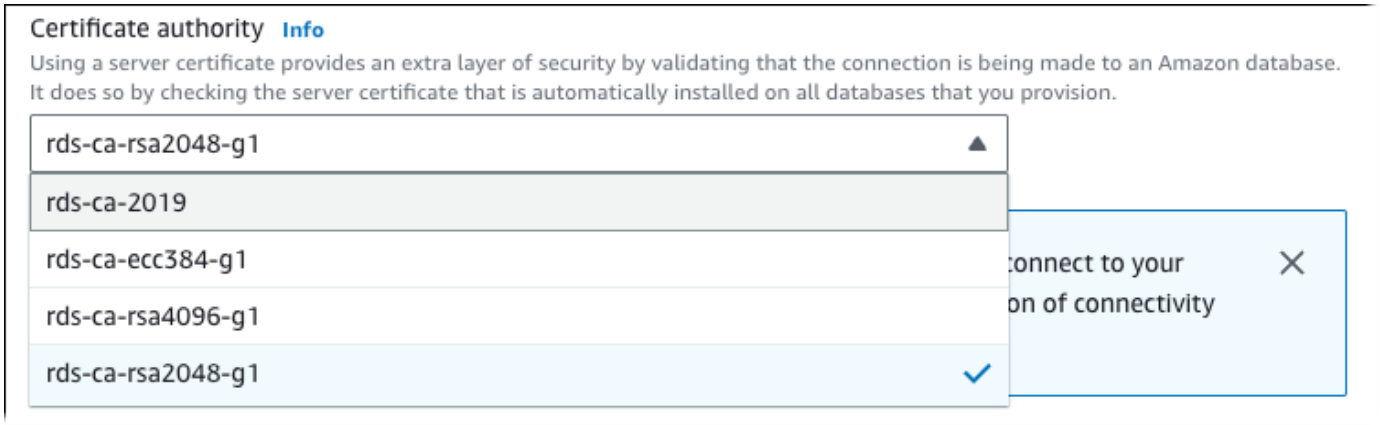
È possibile utilizzare AWS Management Console o the AWS CLI per modificare il certificato CA da rds-ca-2019 a rds-ca-rsa2048-g1 per un'istanza DB o un cluster DB Multi-AZ.

Console

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, scegli Database, quindi scegli l'istanza DB o il cluster DB Multi-AZ che desideri modificare.
3. Scegli Modifica.



4. Nella sezione Connettività, scegli rds-ca-rsa2048-g1.



5. Scegliere Continue (Continua) e controllare il riepilogo delle modifiche.
6. Per applicare immediatamente le modifiche, scegliere Apply immediately (Applica immediatamente).
7. Nella pagina di conferma esaminare le modifiche. Se sono corrette, scegli Modifica istanza DB o Modifica cluster per salvare le modifiche.

   > **Important**
   >
   > Quando si pianifica questa operazione, accertarsi di aver aggiornato in anticipo l'archivio di trust lato client.

   Oppure scegliere Back (Indietro) per cambiare le modifiche o Cancel (Annulla) per annullare le modifiche.

AWS CLI

Per utilizzare il AWS CLI per modificare la CA da rds-ca-2019 a rds-ca-rsa2048-g1 per un'istanza DB o un cluster DB Multi-AZ, chiama il comando or. modify-db-instancemodify-db-cluster Specificare l'identificatore dell'istanza DB o del cluster e l'opzione. `--ca-certificate-identifier`

Utilizzate il `--apply-immediately` parametro per applicare immediatamente l'aggiornamento. Per impostazione predefinita, questa operazione è pianificata per l'esecuzione durante la prossima finestra di manutenzione.

> **Important**
>
> Quando si pianifica questa operazione, accertarsi di aver aggiornato in anticipo l'archivio di trust lato client.

Example

Istanza database

L'esempio seguente esegue la modifica `mydbinstance` impostando il certificato CA su `rds-ca-rsa2048-g1`.

Per LinuxmacOS, oUnix:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --ca-certificate-identifier rds-ca-rsa2048-g1
```

Per Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --ca-certificate-identifier rds-ca-rsa2048-g1
```

> **Note**
>
> Se l'istanza richiede il riavvio, puoi utilizzare il comando modify-db-instanceCLI e specificare `--no-certificate-rotation-restart` l'opzione.

Example

Cluster DB Multi-AZ

L'esempio seguente esegue la modifica `mydbcluster` impostando il certificato CA su. `rds-ca-rsa2048-g1`

Per LinuxmacOS, oUnix:

```
aws rds modify-db-cluster \
    --db-cluster-identifier mydbcluster \
    --ca-certificate-identifier rds-ca-rsa2048-g1
```

Per Windows:

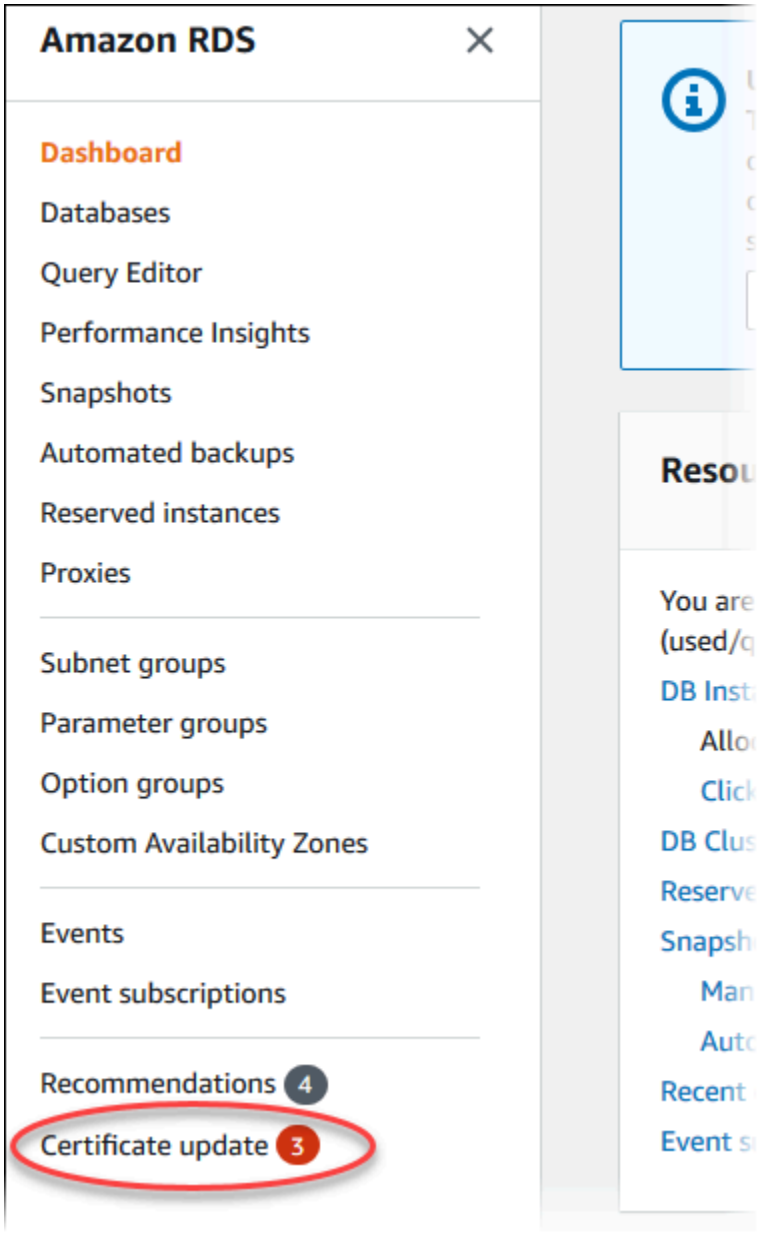
```
aws rds modify-db-cluster ^
    --db-cluster-identifier mydbcluster ^
    --ca-certificate-identifier rds-ca-rsa2048-g1
```

Aggiornamento del certificato CA mediante l'applicazione di manutenzione

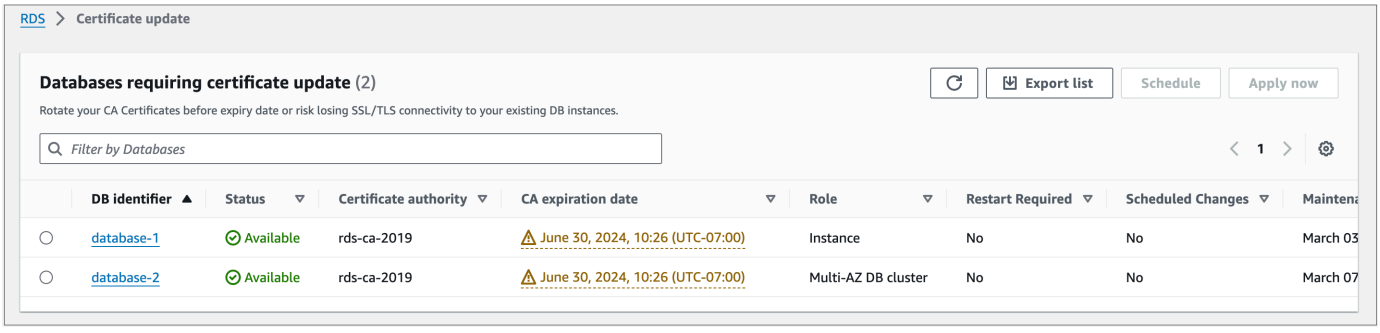
Esegui i passaggi seguenti per aggiornare il certificato CA applicando la manutenzione.

Per aggiornare il certificato CA applicando la manutenzione

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, scegli Aggiornamento del certificato.



Viene visualizzata la pagina Database con aggiornamento certificati richiesto.



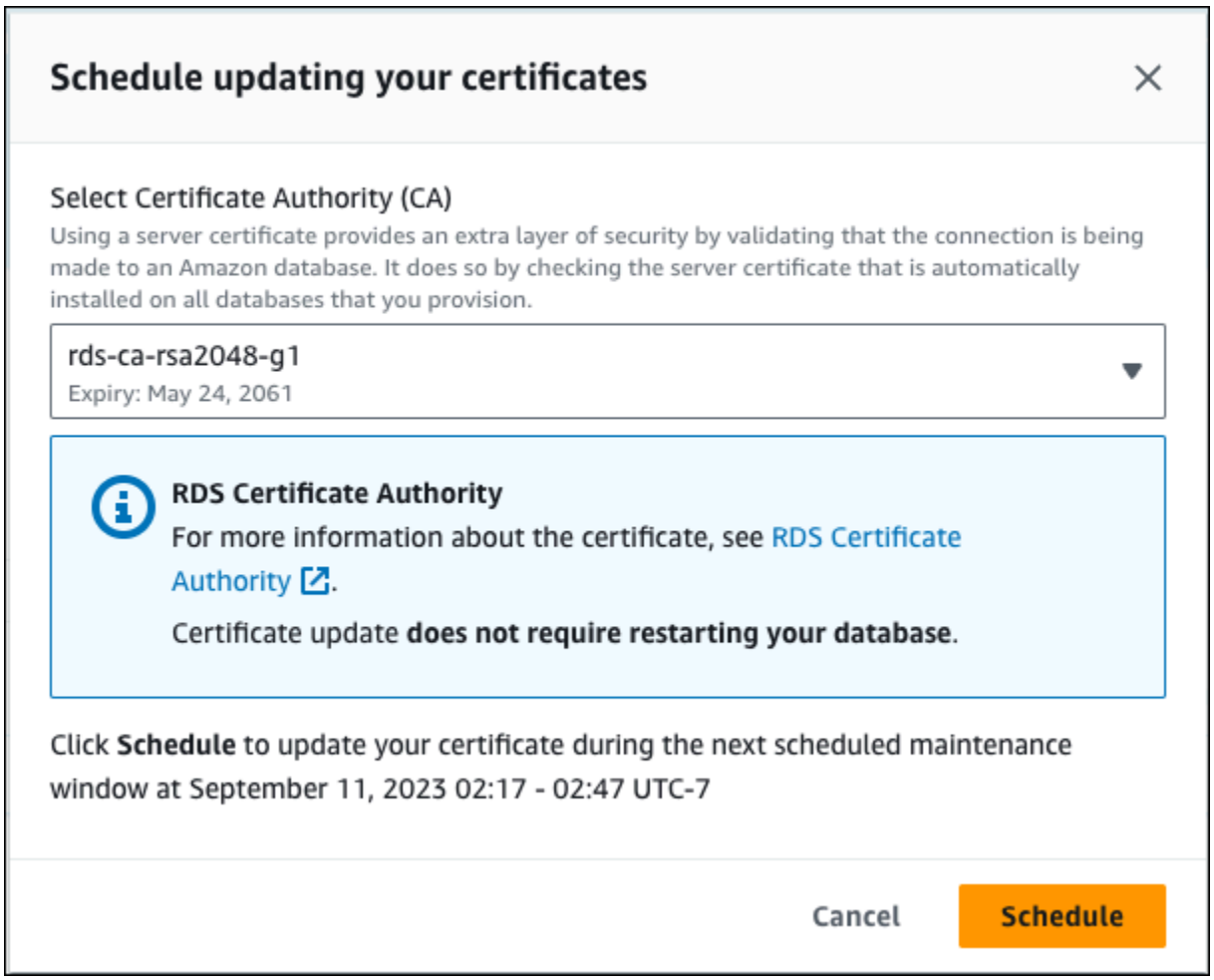
> **Note**
>
> Questa pagina mostra solo le istanze e i cluster DB della versione corrente. Regione AWS Se disponi di database in più di uno Regione AWS, controlla questa pagina in ciascuno di essi Regione AWS per vedere tutte le istanze DB con vecchi certificati SSL/TLS.

3. Scegli l'istanza DB o il cluster DB Multi-AZ che desideri aggiornare.

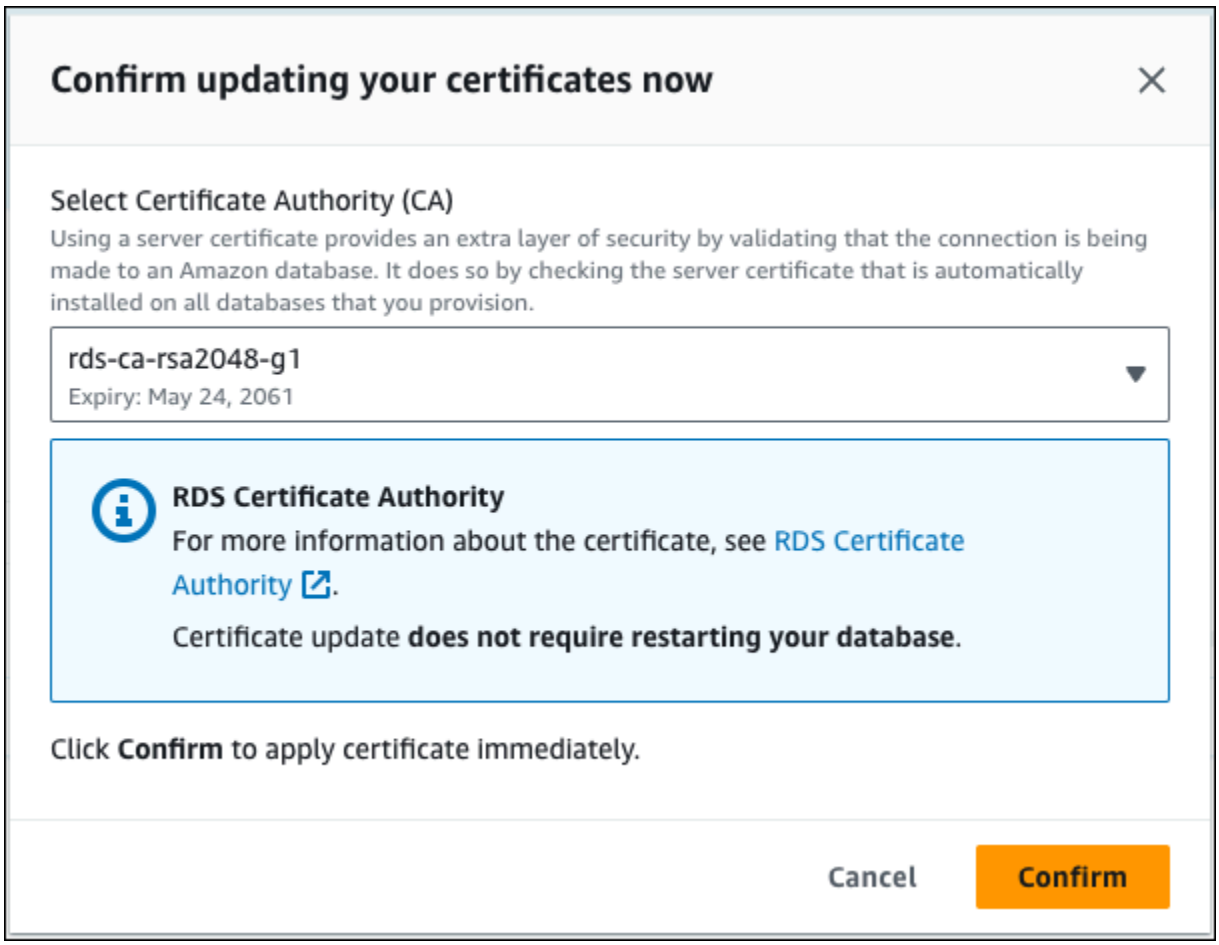
   È possibile pianificare la rotazione dei certificati per la finestra di manutenzione successiva scegliendo Pianifica. Applica immediatamente la rotazione scegliendo Applica ora.

   > **Important**
   >
   > Se si verificano problemi di connettività dopo la scadenza del certificato, utilizza l'opzione Applica ora.

4. a. Se scegli Pianifica, ti viene richiesto di confermare la rotazione dei certificati CA. Nella richiesta viene indicato anche la finestra pianificata per l'aggiornamento.



b. Se scegli Applica ora, ti viene richiesto di confermare la rotazione dei certificati CA.



> **Important**
>
> Prima di pianificare la rotazione del certificato CA nel database, aggiornare tutte le applicazioni client che utilizzano SSL/TLS e il certificato server per connettersi. Questi aggiornamenti sono specifici per il motore DB. Dopo avere aggiornato queste applicazioni client, è possibile confermare la rotazione del certificato CA.

Per continuare, scegliere la casella di controllo e quindi scegliere Confirm (Conferma).

5. Ripeti i passaggi 3 e 4 per ogni istanza DB e cluster che desideri aggiornare.

Rotazione automatica dei certificati del server

Se la CA supporta la rotazione automatica dei certificati del server, RDS gestisce automaticamente la rotazione dei certificati del server di database. Per questa rotazione automatica, RDS utilizza la

stessa CA root e pertanto non è necessario scaricare un nuovo bundle CA. Per informazioni, consulta Autorità di certificazione.

La rotazione e la validità del certificato del server di database dipendono dal motore di database:

- Se il motore di database supporta la rotazione senza riavvio, RDS esegue automaticamente la rotazione del certificato del server di database senza richiedere alcuna azione da parte dell'utente. RDS tenta di eseguire la rotazione del certificato del server di database nella finestra di manutenzione preferita in corrispondenza della semivita del certificato del server di database. Il nuovo certificato del server di database è valido per 12 mesi.
- Se il motore di database in uso non supporta la rotazione senza riavvio, RDS ti avvisa di un evento di manutenzione almeno 6 mesi prima della scadenza del certificato del server di database. Il nuovo certificato del server di database è valido per 36 mesi.

Usa il describe-db-engine-versionscomando e controlla il `SupportsCertificateRotationWithoutRestart` flag per identificare se la versione del motore DB supporta la rotazione del certificato senza riavvio. Per ulteriori informazioni, consulta Impostazione della CA per il database.

Script di esempio per l'importazione di certificati nel tuo archivio di trust

Di seguito sono riportati script di shell di esempio che importano il bundle di certificati in un archivio di trust.

Ogni script di shell di esempio utilizza keytool, che fa parte del Java Development Kit (JDK). Per informazioni sull'installazione di JDK, consulta la Guida di installazione di JDK.

Argomenti

- Script di esempio per l'importazione di certificati su Linux
- Script di esempio per l'importazione di certificati su macOS

Script di esempio per l'importazione di certificati su Linux

Il seguente script è uno script di esempio shell che importa il bundle di certificati in un archivio di trust su un sistema operativo Linux.

```
mydir=tmp/certs
if [ ! -e "${mydir}" ]
```

```
then
mkdir -p "${mydir}"
fi

truststore=${mydir}/rds-truststore.jks
storepassword=changeit

curl -sS "https://truststore.pki.rds.amazonaws.com/global/global-bundle.pem" >
 ${mydir}/global-bundle.pem
awk 'split_after == 1 {n++;split_after=0} /-----END CERTIFICATE-----/ {split_after=1}
{print > "rds-ca-" n+1 ".pem"}' < ${mydir}/global-bundle.pem

for CERT in rds-ca-*; do
  alias=$(openssl x509 -noout -text -in $CERT | perl -ne 'next unless /Subject:/;
 s/.*(CN=|CN = )//; print')
  echo "Importing $alias"
  keytool -import -file ${CERT} -alias "${alias}" -storepass ${storepassword} -keystore
 ${truststore} -noprompt
  rm $CERT
done

rm ${mydir}/global-bundle.pem

echo "Trust store content is: "

keytool -list -v -keystore "$truststore" -storepass ${storepassword} | grep Alias | cut
 -d " " -f3- | while read alias
do
   expiry=`keytool -list -v -keystore "$truststore" -storepass ${storepassword} -alias
 "${alias}" | grep Valid | perl -ne 'if(/until: (.*?)\n/) { print "$1\n"; }'`
   echo " Certificate ${alias} expires in '$expiry'"
done
```

Script di esempio per l'importazione di certificati su macOS

Il seguente script è uno script di shell di esempio che importa il bundle di certificati in un archivio di trust su un sistema operativo Linux.

```
mydir=tmp/certs
if [ ! -e "${mydir}" ]
then
```

```
mkdir -p "${mydir}"
fi

truststore=${mydir}/rds-truststore.jks
storepassword=changeit

curl -sS "https://truststore.pki.rds.amazonaws.com/global/global-bundle.pem" >
 ${mydir}/global-bundle.pem
split -p "-----BEGIN CERTIFICATE-----" ${mydir}/global-bundle.pem rds-ca-

for CERT in rds-ca-*; do
  alias=$(openssl x509 -noout -text -in $CERT | perl -ne 'next unless /Subject:/;
 s/.*(CN=|CN = )//; print')
  echo "Importing $alias"
  keytool -import -file ${CERT} -alias "${alias}" -storepass ${storepassword} -keystore
 ${truststore} -noprompt
  rm $CERT
done

rm ${mydir}/global-bundle.pem

echo "Trust store content is: "

keytool -list -v -keystore "$truststore" -storepass ${storepassword} | grep Alias | cut
 -d " " -f3- | while read alias
do
   expiry=`keytool -list -v -keystore "$truststore" -storepass ${storepassword} -alias
 "${alias}" | grep Valid | perl -ne 'if(/until: (.*?)\n/) { print "$1\n"; }'`
   echo " Certificate ${alias} expires in '$expiry'"
done
```

Riservatezza del traffico Internet

Le connessioni sono protette tra Amazon RDS e le applicazioni on-premise e tra Amazon RDS e altre risorse AWS nella stessa regione AWS.

Traffico tra servizio e applicazioni e client locali

Sono disponibili due opzioni di connettività tra la rete privata e AWS:

- Una connessione Site-to-Site VPN AWS Per ulteriori informazioni, consulta Che cos'è AWS Site-to-Site VPN?

- Una connessione AWS Direct Connect. Per ulteriori informazioni, consulta Che cos'è AWS Direct Connect?

Puoi ottenere l'accesso a Amazon RDS tramite la rete utilizzando le operazioni API pubblicate da AWS. I clienti devono supportare quanto segue:

- Transport Layer Security (TLS). È richiesto TLS 1.2 ed è consigliato TLS 1.3.
- Suite di cifratura con Perfect Forward Secrecy (PFS), ad esempio Ephemeral Diffie-Hellman (DHE) o Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). La maggior parte dei sistemi moderni, come Java 7 e versioni successive, supporta tali modalità.

Inoltre, le richieste devono essere firmate utilizzando un ID chiave di accesso e una chiave di accesso segreta associata a un principale IAM. In alternativa, è possibile utilizzare AWS Security Token Service (AWS STS) per generare le credenziali di sicurezza temporanee per sottoscrivere le richieste.

Gestione accessi e identità per Amazon RDS

AWS Identity and Access Management (IAM) è un programma Servizio AWS che aiuta un amministratore a controllare in modo sicuro l'accesso alle AWS risorse. Gli amministratori IAM controllano chi può essere autenticato (accesso effettuato) e autorizzato (dispone di autorizzazioni) per utilizzare le risorse Amazon RDS. IAM è un software Servizio AWS che puoi utilizzare senza costi aggiuntivi.

Argomenti

Destinatari

Il modo in cui utilizzi AWS Identity and Access Management (IAM) varia a seconda del lavoro svolto in Amazon RDS Amazon .

Utente del servizio –Se utilizzi il servizio Amazon RDS per eseguire il tuo lavoro, l'amministratore fornisce le credenziali e le autorizzazioni necessarie. All'aumentare del numero di funzionalità Amazon RDS utilizzate per il lavoro, potrebbero essere necessarie ulteriori autorizzazioni. La comprensione della gestione dell'accesso consente di richiedere le autorizzazioni corrette all'amministratore. Se non riesci ad accedere a una funzionalità di Amazon RDS, consulta Risoluzione dei problemi di identità e accesso in Amazon RDS.

Amministratore del servizio – Se sei il responsabile delle risorse Amazon RDSpresso la tua azienda, probabilmente disponi dell'accesso completo a Amazon RDS. Il tuo compito è determinare le

caratteristiche e le risorse Amazon RDS a cui i dipendenti devono accedere. Devi inviare le richieste all'amministratore per cambiare le autorizzazioni degli utenti del servizio. Esamina le informazioni contenute in questa pagina per comprendere le nozioni di base di IAM. Per ulteriori informazioni su come la tua azienda può utilizzare IAM con Amazon RDS, consulta Funzionamento di Amazon RDS con IAM.

Amministratore – Se sei un amministratore, potresti essere interessato a ottenere informazioni su come puoi scrivere policy per gestire l'accesso a Amazon RDS. Per visualizzare policy basate su identità Amazon RDS di esempio che puoi utilizzare in IAM, consulta Esempi di policy di Amazon RDS basate su identità.

Autenticazione con identità

L'autenticazione è il modo in cui accedi AWS utilizzando le tue credenziali di identità. Devi essere autenticato (aver effettuato l' Utente root dell'account AWS accesso AWS) come utente IAM o assumendo un ruolo IAM.

Puoi accedere AWS come identità federata utilizzando le credenziali fornite tramite una fonte di identità. AWS IAM Identity Center Gli utenti (IAM Identity Center), l'autenticazione Single Sign-On della tua azienda e le tue credenziali di Google o Facebook sono esempi di identità federate. Se accedi come identità federata, l'amministratore ha configurato in precedenza la federazione delle identità utilizzando i ruoli IAM. Quando accedi AWS utilizzando la federazione, assumi indirettamente un ruolo.

A seconda del tipo di utente, puoi accedere al AWS Management Console o al portale di AWS accesso. Per ulteriori informazioni sull'accesso a AWS, vedi Come accedere al tuo Account AWS nella Guida per l'Accedi ad AWS utente.

Se accedi a AWS livello di codice, AWS fornisce un kit di sviluppo software (SDK) e un'interfaccia a riga di comando (CLI) per firmare crittograficamente le tue richieste utilizzando le tue credenziali. Se non utilizzi AWS strumenti, devi firmare tu stesso le richieste. Per ulteriori informazioni sull'utilizzo del metodo consigliato per firmare autonomamente le richieste, consulta Signing AWS API request nella IAM User Guide.

A prescindere dal metodo di autenticazione utilizzato, potrebbe essere necessario specificare ulteriori informazioni sulla sicurezza. Ad esempio, ti AWS consiglia di utilizzare l'autenticazione a più fattori (MFA) per aumentare la sicurezza del tuo account. Per ulteriori informazioni, consulta Autenticazione a più fattori nella Guida per l'utente di AWS IAM Identity Center e Utilizzo dell'autenticazione a più fattori (MFA) in AWS nella Guida per l'utente di IAM.

AWS account utente root

Quando si crea un account Account AWS, si inizia con un'identità di accesso che ha accesso completo a tutte Servizi AWS le risorse dell'account. Questa identità è denominata utente Account AWS root ed è accessibile effettuando l'accesso con l'indirizzo e-mail e la password utilizzati per creare l'account. Si consiglia vivamente di non utilizzare l'utente root per le attività quotidiane. Conserva le credenziali dell'utente root e utilizzarle per eseguire le operazioni che solo l'utente root può eseguire. Per un elenco completo delle attività che richiedono l'accesso come utente root, consulta la sezione Attività che richiedono le credenziali dell'utente root nella Guida per l'utente di IAM.

Identità federata

Come procedura consigliata, richiedi agli utenti umani, compresi gli utenti che richiedono l'accesso come amministratore, di utilizzare la federazione con un provider di identità per accedere Servizi AWS utilizzando credenziali temporanee.

Un'identità federata è un utente dell'elenco utenti aziendale, di un provider di identità Web AWS Directory Service, della directory Identity Center o di qualsiasi utente che accede utilizzando le Servizi AWS credenziali fornite tramite un'origine di identità. Quando le identità federate accedono Account AWS, assumono ruoli e i ruoli forniscono credenziali temporanee.

Per la gestione centralizzata degli accessi, consigliamo di utilizzare AWS IAM Identity Center. Puoi creare utenti e gruppi in IAM Identity Center oppure puoi connetterti e sincronizzarti con un set di utenti e gruppi nella tua fonte di identità per utilizzarli su tutte le tue applicazioni. Account AWS Per ulteriori informazioni sul Centro identità IAM, consulta Cos'è Centro identità IAM? nella Guida per l'utente di AWS IAM Identity Center .

Utenti e gruppi IAM

Un utente IAM è un'identità interna Account AWS che dispone di autorizzazioni specifiche per una singola persona o applicazione. Ove possibile, consigliamo di fare affidamento a credenziali temporanee invece di creare utenti IAM con credenziali a lungo termine come le password e le chiavi di accesso. Tuttavia, per casi d'uso specifici che richiedono credenziali a lungo termine con utenti IAM, si consiglia di ruotare le chiavi di accesso. Per ulteriori informazioni, consulta la pagina Rotazione periodica delle chiavi di accesso per casi d'uso che richiedono credenziali a lungo termine nella Guida per l'utente di IAM.

Un gruppo IAM è un'identità che specifica un insieme di utenti IAM. Non è possibile eseguire l'accesso come gruppo. È possibile utilizzare gruppi per specificare le autorizzazioni per più utenti alla

volta. I gruppi semplificano la gestione delle autorizzazioni per set di utenti di grandi dimensioni. Ad esempio, è possibile avere un gruppo denominato Amministratori IAM e concedere a tale gruppo le autorizzazioni per amministrare le risorse IAM.

Gli utenti sono diversi dai ruoli. Un utente è associato in modo univoco a una persona o un'applicazione, mentre un ruolo è destinato a essere assunto da chiunque ne abbia bisogno. Gli utenti dispongono di credenziali a lungo termine permanenti, mentre i ruoli forniscono credenziali temporanee. Per ulteriori informazioni, consulta Quando creare un utente IAM (invece di un ruolo) nella Guida per l'utente di IAM.

Puoi eseguire l'autenticazione al dell'istanza database tramite l'autenticazione del database IAM.

L'autenticazione del database IAM funziona con i seguenti motori DB:

- RDS per MariaDB
- RDS for MySQL
- RDS per PostgreSQL

Per ulteriori informazioni sull'autenticazione all'istanza database tramite IAM, consulta Autenticazione del database IAM per MariaDB, MySQL e PostgreSQL.

Ruoli IAM

Un ruolo IAM è un'identità interna all'utente Account AWS che dispone di autorizzazioni specifiche. È simile a un utente, ma non è associato a una persona specifica. Puoi assumere temporaneamente un ruolo IAM in AWS Management Console cambiando ruolo. Puoi assumere un ruolo chiamando un'operazione AWS CLI o AWS API o utilizzando un URL personalizzato. Per ulteriori informazioni sui metodi per l'utilizzo dei ruoli, consulta Utilizzo di ruoli IAM nella Guida per l'utente di IAM.

I ruoli IAM con credenziali temporanee sono utili nelle seguenti situazioni:

- Autorizzazioni utente temporanee: un utente può assumere un ruolo IAM per ottenere temporaneamente autorizzazioni diverse per un'attività specifica.
- Accesso utente federato: per assegnare le autorizzazioni a una identità federata, è possibile creare un ruolo e definire le autorizzazioni per il ruolo. Quando un'identità federata viene autenticata, l'identità viene associata al ruolo e ottiene le autorizzazioni da esso definite. Per ulteriori informazioni sulla federazione dei ruoli, consulta Creazione di un ruolo per un provider di identità di terza parte nella Guida per l'utente di IAM. Se utilizzi IAM Identity Center, configura un

set di autorizzazioni. IAM Identity Center mette in correlazione il set di autorizzazioni con un ruolo in IAM per controllare a cosa possono accedere le identità dopo l'autenticazione. Per informazioni sui set di autorizzazioni, consultare Set di autorizzazioni nella Guida per l'utente AWS IAM Identity Center .

- Accesso multi-account: è possibile utilizzare un ruolo IAM per permettere a un utente (un principale affidabile) con un account diverso di accedere alle risorse nell'account. I ruoli sono lo strumento principale per concedere l'accesso multi-account. Tuttavia, con alcuni Servizi AWS, è possibile allegare una policy direttamente a una risorsa (anziché utilizzare un ruolo come proxy). Per informazioni sulle differenze tra ruoli e policy basate su risorse per l'accesso multi-account, consulta Differenza tra i ruoli IAM e le policy basate su risorse nella Guida per l'utente di IAM.
- Accesso a più servizi: alcuni Servizi AWS utilizzano le funzionalità di altri Servizi AWS. Ad esempio, quando effettui una chiamata in un servizio, è comune che tale servizio esegua applicazioni in Amazon EC2 o archivi oggetti in Amazon S3. Un servizio può eseguire questa operazione utilizzando le autorizzazioni dell'entità chiamante, utilizzando un ruolo di servizio o utilizzando un ruolo collegato al servizio.
  - Sessioni di accesso diretto: quando utilizzi un utente o un ruolo IAM per eseguire azioni AWS, sei considerato un principale. Quando si utilizzano alcuni servizi, è possibile eseguire un'operazione che attiva un'altra azione in un servizio diverso. FAS utilizza le autorizzazioni del principale che chiama an Servizio AWS, combinate con la richiesta Servizio AWS per effettuare richieste ai servizi downstream. Le richieste FAS vengono effettuate solo quando un servizio riceve una richiesta che richiede interazioni con altri Servizi AWS o risorse per essere completata. In questo caso è necessario disporre delle autorizzazioni per eseguire entrambe le azioni. Per i dettagli delle policy relative alle richieste FAS, consulta la pagina Forward access sessions.
  - Ruolo di servizio: un ruolo di servizio è un ruolo IAM che un servizio assume per eseguire azioni per tuo conto. Un amministratore IAM può creare, modificare ed eliminare un ruolo di servizio dall'interno di IAM. Per ulteriori informazioni, consulta la sezione Creazione di un ruolo per delegare le autorizzazioni a un Servizio AWS nella Guida per l'utente di IAM.
  - Ruolo collegato al servizio: un ruolo collegato al servizio è un tipo di ruolo di servizio collegato a un. Servizio AWS Il servizio può assumere il ruolo per eseguire un'azione per tuo conto. I ruoli collegati al servizio vengono visualizzati nel tuo account Account AWS e sono di proprietà del servizio. Un amministratore IAM può visualizzare le autorizzazioni per i ruoli collegati ai servizi, ma non modificarle.
- Applicazioni in esecuzione su Amazon EC2: puoi utilizzare un ruolo IAM per gestire le credenziali temporanee per le applicazioni in esecuzione su un'istanza EC2 e che AWS CLI effettuano

richieste API. AWS Ciò è preferibile all'archiviazione delle chiavi di accesso nell'istanza EC2. Per assegnare un AWS ruolo a un'istanza EC2 e renderlo disponibile per tutte le sue applicazioni, crei un profilo di istanza collegato all'istanza. Un profilo dell'istanza contiene il ruolo e consente ai programmi in esecuzione sull'istanza EC2 di ottenere le credenziali temporanee. Per ulteriori informazioni, consulta Utilizzo di un ruolo IAM per concedere autorizzazioni ad applicazioni in esecuzione su istanze di Amazon EC2 nella Guida per l'utente di IAM.

Per informazioni sull'utilizzo di ruoli IAM, consulta Quando creare un ruolo IAM invece di un utente nella Guida per l'utente di IAM.

Gestione dell'accesso con policy

Puoi controllare l'accesso AWS creando policy e collegandole a identità o risorse IAM. AWS Una policy è un oggetto AWS che, se associato a un'identità o a una risorsa, ne definisce le autorizzazioni. AWS valuta queste politiche quando un'entità (utente root, utente o ruolo IAM) effettua una richiesta. Le autorizzazioni nelle policy determinano l'approvazione o il rifiuto della richiesta. La maggior parte delle politiche viene archiviata AWS come documenti JSON. Per ulteriori informazioni sulla struttura e sui contenuti dei documenti delle policy JSON, consulta Panoramica delle policy JSON nella Guida per l'utente di IAM.

Un amministratore può utilizzare le policy per specificare chi ha accesso alle AWS risorse e quali azioni può eseguire su tali risorse. Ogni entità IAM (set di autorizzazioni o ruolo) inizialmente non dispone di autorizzazioni. Ovvero, di default, gli utenti non possono eseguire alcuna operazione, neppure modificare la propria password. Per autorizzare un utente a eseguire operazioni, un amministratore deve allegare una policy di autorizzazioni a tale utente. In alternativa, l'amministratore può aggiungere l'utente a un gruppo che dispone delle autorizzazioni desiderate. Quando un amministratore fornisce le autorizzazioni a un gruppo, le autorizzazioni vengono concesse a tutti gli utenti in tale gruppo.

Le policy IAM definiscono le autorizzazioni relative a un'operazione, indipendentemente dal metodo utilizzato per eseguirla. Ad esempio, supponiamo di disporre di una policy che consente l'azione `iam:GetRole`. Un utente con tale policy può ottenere informazioni sul ruolo dall' AWS Management Console AWS CLI, dall'o dall' AWS API.

Policy basate su identità

Le policy basate su identità sono documenti di policy di autorizzazione JSON che è possibile collegare a un'identità, ad esempio un set di autorizzazioni o un ruolo. Tali policy definiscono le

operazioni autorizzate per l'identità, nonché le risorse e le condizioni in cui possono essere eseguite. Per informazioni su come creare una policy basata su identità, consulta Creazione di policy IAM nella Guida per l'utente di IAM.

Le policy basate su identità possono essere ulteriormente classificate come policy inline o policy gestite. Le policy inline sono incorporate direttamente in un singolo set di autorizzazioni o ruolo. Le politiche gestite sono politiche autonome che puoi allegare a più set di autorizzazioni e ruoli nel tuo AWS account. Le politiche gestite includono politiche AWS gestite e politiche gestite dai clienti. Per informazioni su come scegliere tra una policy gestita o una policy inline, consulta Scelta fra policy gestite e policy inline nella Guida per l'utente di IAM.

Per informazioni sulle policy AWS gestite specifiche di Amazon RDS Aurora, consulta. AWS politiche gestite per Amazon RDS

Altri tipi di policy

AWS supporta tipi di policy aggiuntivi e meno comuni. Questi tipi di policy possono impostare il numero massimo di autorizzazioni concesse dai tipi di policy più comuni.

- Limiti delle autorizzazioni: un limite delle autorizzazioni è una funzione avanzata nella quale si imposta il numero massimo di autorizzazioni che una policy basata su identità può concedere a un'entità IAM (set di autorizzazioni o ruolo). È possibile impostare un limite delle autorizzazioni per un'entità. Le autorizzazioni risultanti sono l'intersezione delle policy basate su identità dell'entità e i suoi limiti delle autorizzazioni. Le policy basate su risorse che specificano il set di autorizzazioni o il ruolo nel campo `Principal` sono condizionate dal limite delle autorizzazioni. Un rifiuto esplicito in una qualsiasi di queste policy sostituisce l'autorizzazione. Per ulteriori informazioni sui limiti delle autorizzazioni, consulta Limiti delle autorizzazioni per le entità IAM nella Guida per l'utente di IAM.
- Politiche di controllo dei servizi (SCP): le SCP sono politiche JSON che specificano le autorizzazioni massime per un'organizzazione o un'unità organizzativa (OU) in. AWS Organizations AWS Organizations è un servizio per il raggruppamento e la gestione centralizzata di più AWS account di proprietà dell'azienda. Se abiliti tutte le funzionalità in un'organizzazione, puoi applicare le policy di controllo dei servizi (SCP) a uno o tutti i tuoi account. L'SCP limita le autorizzazioni per le entità negli account dei membri, inclusa ciascuna. Utente root dell'account AWS Per ulteriori informazioni su organizzazioni e policy SCP, consulta la pagina sulle Policy di controllo dei servizi nella Guida per l'utente di AWS Organizations .
- Policy di sessione: le policy di sessione sono policy avanzate che vengono trasmesse come parametro quando si crea in modo programmatico una sessione temporanea per un ruolo o un utente federato. Le autorizzazioni della sessione risultante sono l'intersezione delle policy basate

sull'identità del set di autorizzazioni o del ruolo e le policy di sessione. Le autorizzazioni possono anche provenire da una policy basata su risorse. Un rifiuto esplicito in una qualsiasi di queste policy sostituisce l'autorizzazione. Per ulteriori informazioni, consulta Policy di sessione nella Guida per l'utente di IAM.

Più tipi di policy

Quando più tipi di policy si applicano a una richiesta, le autorizzazioni risultanti sono più complicate da comprendere. Per scoprire come si AWS determina se consentire una richiesta quando sono coinvolti più tipi di policy, consulta Logica di valutazione delle policy nella IAM User Guide.

Funzionamento di Amazon RDS con IAM

Prima di utilizzare IAM per gestire l'accesso ad Amazon RDS, è necessario comprendere quali funzionalità IAM sono disponibili per l'uso con Amazon RDS.

Funzionalità IAM che è possibile utilizzare con Amazon RDS

Funzionalità IAM	Supporto di Amazon RDS
Policy basate su identità	Sì
Policy basate su risorse	No
Azioni di policy	Sì
Risorse relative alle policy	Sì
Chiavi di condizione della policy (specifica del servizio)	Sì
Liste di controllo degli accessi (ACL)	No
Controllo degli accessi basato su attributi (ABAC) (tag nelle policy)	Sì
Credenziali temporanee	Sì
Sessioni di accesso diretto	Sì
Ruoli di servizio	Sì

Funzionalità IAM	Supporto di Amazon RDS
Ruoli collegati al servizio	Sì

Per avere una visione di alto livello di come Amazon RDS Amazon e AWS altri servizi funzionano con IAM, AWS consulta i servizi che funzionano con IAM nella IAM User Guide.

Argomenti

- Policy basate su identità Amazon RDS
- Policy basate su risorse all'interno di Amazon RDS
- Operazioni delle policy per Amazon RDS
- Risorse delle policy per Amazon RDS
- Chiavi di condizione delle policy per Amazon RDS
- Liste di controllo degli accessi (ACL) in Amazon RDS
- Controllo degli accessi basato su attributi (ABAC) nelle policy con tag Amazon RDS
- Utilizzo di credenziali temporanee con Amazon RDS
-
- Ruoli di servizio per Amazon RDS
- Ruoli collegati ai servizi per Amazon RDS

Policy basate su identità Amazon RDS

Supporta le policy basate su identità	Sì

Le policy basate su identità sono documenti di policy di autorizzazione JSON che è possibile allegare a un'identità (utente, gruppo di utenti o ruolo IAM). Tali policy definiscono le azioni che utenti e ruoli possono eseguire, su quali risorse e in quali condizioni. Per informazioni su come creare una policy basata su identità, consulta Creazione di policy IAM nella Guida per l'utente di IAM.

Con le policy basate su identità di IAM, è possibile specificare quali operazioni e risorse sono consentite o respinte, nonché le condizioni in base alle quali le operazioni sono consentite o respinte. Non è possibile specificare l'entità principale in una policy basata sull'identità perché si applica all'utente o al ruolo a cui è associato. Per informazioni su tutti gli elementi utilizzabili in una policy

JSON, consulta Guida di riferimento agli elementi delle policy JSON IAM nella Guida per l'utente di IAM.

Esempi di policy basate su identità per Amazon RDS

Per visualizzare esempi di policy basate su identità Amazon RDS, consulta Esempi di policy di Amazon RDS basate su identità.

Policy basate su risorse all'interno di Amazon RDS

Supporta le policy basate su risorse	No

Le policy basate su risorse sono documenti di policy JSON che è possibile collegare a una risorsa. Gli esempi più comuni di policy basate su risorse sono le policy di attendibilità dei ruoli IAM e le policy dei bucket Amazon S3. Nei servizi che supportano policy basate sulle risorse, gli amministratori dei servizi possono utilizzarle per controllare l'accesso a una risorsa specifica. Quando è collegata a una risorsa, una policy definisce le azioni che un principale può eseguire su tale risorsa e a quali condizioni. È necessario specificare un principale in una policy basata sulle risorse. I principali possono includere account, utenti, ruoli, utenti federati o. Servizi AWS

Per consentire l'accesso multi-account, puoi specificare un intero account o entità IAM in un altro account come principale in una policy basata sulle risorse. L'aggiunta di un principale multi-account a una policy basata sulle risorse rappresenta solo una parte della relazione di trust. Quando il principale e la risorsa sono diversi Account AWS, un amministratore IAM dell'account affidabile deve inoltre concedere all'entità principale (utente o ruolo) l'autorizzazione ad accedere alla risorsa. L'autorizzazione viene concessa collegando all'entità una policy basata sull'identità. Tuttavia, se una policy basata su risorse concede l'accesso a un principale nello stesso account, non sono richieste ulteriori policy basate su identità. Per ulteriori informazioni, consulta Differenza tra i ruoli IAM e le policy basate su risorse nella Guida per l'utente di IAM.

Operazioni delle policy per Amazon RDS

Supporta le azioni di policy	Sì

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. Cioè, quale principale può eseguire azioni su quali risorse, e in quali condizioni.

L'elemento `Action` di una policy JSON descrive le azioni che è possibile utilizzare per consentire o negare l'accesso a un criterio. Le azioni politiche in genere hanno lo stesso nome dell'operazione AWS API associata. Ci sono alcune eccezioni, ad esempio le azioni di sola autorizzazione che non hanno un'operazione API corrispondente. Esistono anche alcune operazioni che richiedono più operazioni in una policy. Queste operazioni aggiuntive sono denominate operazioni dipendenti.

Includi le operazioni in una policy per concedere le autorizzazioni a eseguire l'operazione associata.

Le operazioni delle policy in Amazon RDS utilizzano il seguente prefisso prima dell'operazione: `rds:`. Ad esempio, per concedere a qualcuno l'autorizzazione per descrivere istanze database con l'operazione API Amazon RDS `DescribeDBInstances`, includi l'operazione `rds:DescribeDBInstances` nella policy. Le istruzioni della policy devono includere un elemento `Action` o `NotAction`. Amazon RDS definisce un proprio set di operazioni che descrivono le attività che puoi eseguire con quel servizio.

Per specificare più operazioni in una singola istruzione, separarle con una virgola come mostrato di seguito.

```
"Action": [
      "rds:action1",
      "rds:action2"
```

Puoi specificare più operazioni tramite caratteri jolly (*). Ad esempio, per specificare tutte le operazioni che iniziano con la parola `Describe`, includi la seguente operazione.

```
"Action": "rds:Describe*"
```

Per visualizzare un elenco di operazioni di Amazon RDS, consulta Operazioni definite da Amazon RDS nella Service Authorization Reference.

Risorse delle policy per Amazon RDS

Supporta le risorse di policy	Sì

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. Cioè, quale principale può eseguire operazioni su quali risorse, e in quali condizioni.

L'elemento JSON `Resource` della policy specifica l'oggetto o gli oggetti ai quali si applica l'azione. Le istruzioni devono includere un elemento `Resource` o un elemento `NotResource`. Come best practice, specifica una risorsa utilizzando il suo nome della risorsa Amazon (ARN). Puoi eseguire questa operazione per azioni che supportano un tipo di risorsa specifico, note come autorizzazioni a livello di risorsa.

Per le azioni che non supportano le autorizzazioni a livello di risorsa, ad esempio le operazioni di elenco, utilizza un carattere jolly (*) per indicare che l'istruzione si applica a tutte le risorse.

```
"Resource": "*"
```

La risorsa di un'istanza database ha il seguente nome della risorsa Amazon (ARN).

```
arn:${Partition}:rds:${Region}:${Account}:{ResourceType}/${Resource}
```

Per ulteriori informazioni sul formato degli ARN, consulta Amazon Resource Names (ARNs) e AWS service namespace.

Ad esempio, per specificare l'istanza database `dbtest` nell'istruzione, utilizza l'ARN riportato di seguito.

```
"Resource": "arn:aws:rds:us-west-2:123456789012:db:dbtest"
```

Per specificare tutte le istanze database che appartengono a un account specifico, utilizza il carattere jolly (*).

```
"Resource": "arn:aws:rds:us-east-1:123456789012:db:*"
```

Alcune operazioni API RDS, ad esempio quelle per la creazione di risorse, non possono essere eseguite su una risorsa specifica. In questi casi, utilizza il carattere jolly (*).

```
"Resource": "*"
```

Molte operazioni API di Amazon RDS coinvolgono più risorse. Ad esempio, `CreateDBInstance` crea un'istanza database. Puoi specificare che un utente deve utilizzare un gruppo specifico di sicurezza e un gruppo di parametri quando crea un'istanza database. Per specificare più risorse in una singola istruzione, separa gli ARN con le virgole.

```
"Resource": [
```

```
        "resource1",
        "resource2"
```

Per visualizzare un elenco di tipi di risorse di Amazon RDS, consulta Risorse definite da Amazon RDS nella Service Authorization Reference. Per informazioni sulle operazioni con cui è possibile specificare l'ARN di ogni risorsa, consulta Operazioni definite da Amazon RDS.

Chiavi di condizione delle policy per Amazon RDS

Supporta le chiavi di condizione delle policy specifiche del servizio	Sì

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. Cioè, quale principale può eseguire azioni su quali risorse, e in quali condizioni.

L'elemento `Condition` (o blocco `Condition`) consente di specificare le condizioni in cui un'istruzione è in vigore. L'elemento `Condition` è facoltativo. Puoi compilare espressioni condizionali che utilizzano operatori di condizione, ad esempio uguale a o minore di, per soddisfare la condizione nella policy con i valori nella richiesta.

Se specifichi più elementi `Condition` in un'istruzione o più chiavi in un singolo elemento `Condition`, questi vengono valutati da AWS utilizzando un'operazione `AND` logica. Se si specificano più valori per una singola chiave di condizione, AWS valuta la condizione utilizzando un'operazione logica. `OR` Tutte le condizioni devono essere soddisfatte prima che le autorizzazioni dell'istruzione vengano concesse.

Puoi anche utilizzare variabili segnaposto quando specifichi le condizioni. Ad esempio, puoi autorizzare un utente IAM ad accedere a una risorsa solo se è stata taggata con il relativo nome utente IAM. Per ulteriori informazioni, consulta Elementi delle policy IAM: variabili e tag nella Guida per l'utente di IAM.

AWS supporta chiavi di condizione globali e chiavi di condizione specifiche del servizio. Per visualizzare tutte le chiavi di condizione AWS globali, consulta le chiavi di contesto delle condizioni AWS globali nella Guida per l'utente IAM.

Amazon RDS definisce il proprio set di chiavi di condizione e, inoltre, supporta l'uso di alcune chiavi di condizione globali. Per vedere tutte le chiavi di condizione AWS globali, consulta le chiavi di contesto delle condizioni AWS globali nella Guida per l'utente IAM.

Tutte le operazioni API RDS supportano la chiave di condizione `aws:RequestedRegion`.

Per visualizzare un elenco di chiavi di condizione Amazon RDS, consulta Chiavi di condizione per Amazon RDS nella Service Authorization Reference. Per informazioni su operazioni e risorse con cui è possibile utilizzare una chiave di condizione, consulta Operazioni definite da Amazon RDS.

Liste di controllo degli accessi (ACL) in Amazon RDS

Supporta liste di controllo degli accessi (ACL)	No

Le liste di controllo degli accessi (ACL) controllano quali principali (membri, utenti o ruoli dell'account) hanno le autorizzazioni ad accedere a una risorsa. Le ACL sono simili alle policy basate su risorse, sebbene non utilizzino il formato del documento di policy JSON.

Controllo degli accessi basato su attributi (ABAC) nelle policy con tag Amazon RDS

Supporta tag di controllo degli accessi basato su attributi (ABAC) nelle policy	Sì

Il controllo dell'accesso basato su attributi (ABAC) è una strategia di autorizzazione che definisce le autorizzazioni in base agli attributi. In AWS, questi attributi sono chiamati tag. Puoi allegare tag a entità IAM (utenti o ruoli) e a molte AWS risorse. L'assegnazione di tag alle entità e alle risorse è il primo passaggio di ABAC. In seguito, vengono progettate policy ABAC per consentire operazioni quando il tag dell'entità principale corrisponde al tag sulla risorsa a cui si sta provando ad accedere.

La strategia ABAC è utile in ambienti soggetti a una rapida crescita e aiuta in situazioni in cui la gestione delle policy diventa impegnativa.

Per controllare l'accesso basato su tag, fornisci informazioni sui tag nell'elemento condizione di una policy utilizzando le chiavi di condizione `aws:ResourceTag/`*`key-name`*, `aws:RequestTag/`*`key-name`* o `aws:TagKeys`.

Se un servizio supporta tutte e tre le chiavi di condizione per ogni tipo di risorsa, il valore per il servizio è Yes (Sì). Se un servizio supporta tutte e tre le chiavi di condizione solo per alcuni tipi di risorsa, allora il valore sarà Parziale.

Per ulteriori informazioni su ABAC, consulta Che cos'è ABAC? nella Guida per l'utente di IAM. Per visualizzare un tutorial con i passaggi per l'impostazione di ABAC, consulta Utilizzo del controllo degli accessi basato su attributi (ABAC) nella Guida per l'utente di IAM.

Per ulteriori informazioni sul tagging delle risorse di Amazon RDS, consulta Specifica delle condizioni: Utilizzo di tag personalizzati. Per visualizzare una policy basata sulle identità di esempio per limitare l'accesso a una risorsa basata su tag su tale risorsa, consulta Concedere l'autorizzazione per le operazioni su una risorsa con un tag specifico con due valori diversi.

Utilizzo di credenziali temporanee con Amazon RDS

Supporta le credenziali temporanee	Sì

Alcuni Servizi AWS non funzionano quando accedi utilizzando credenziali temporanee. Per ulteriori informazioni, incluse quelle che Servizi AWS funzionano con credenziali temporanee, consulta la sezione relativa alla Servizi AWS compatibilità con IAM nella IAM User Guide.

Stai utilizzando credenziali temporanee se accedi AWS Management Console utilizzando qualsiasi metodo tranne nome utente e password. Ad esempio, quando accedi AWS utilizzando il link Single Sign-On (SSO) della tua azienda, tale processo crea automaticamente credenziali temporanee. Le credenziali temporanee vengono create in automatico anche quando accedi alla console come utente e poi cambi ruolo. Per ulteriori informazioni sullo scambio dei ruoli, consulta Cambio di un ruolo (console) nella Guida per l'utente di IAM.

È possibile creare manualmente credenziali temporanee utilizzando l'API or. AWS CLI AWS È quindi possibile utilizzare tali credenziali temporanee per accedere. AWS AWS consiglia di generare dinamicamente credenziali temporanee anziché utilizzare chiavi di accesso a lungo termine. Per ulteriori informazioni, consulta Credenziali di sicurezza provvisorie in IAM.

Supporta sessioni di accesso diretto	Sì

Quando utilizzi un utente o un ruolo IAM per eseguire azioni AWS, sei considerato un principale. Quando si utilizzano alcuni servizi, è possibile eseguire un'operazione che attiva un'altra azione in un servizio diverso. FAS utilizza le autorizzazioni del principale che chiama an Servizio AWS, in combinazione con la richiesta Servizio AWS per effettuare richieste ai servizi downstream. Le richieste FAS vengono effettuate solo quando un servizio riceve una richiesta che richiede interazioni

con altri Servizi AWS o risorse per essere completata. In questo caso è necessario disporre delle autorizzazioni per eseguire entrambe le azioni. Per i dettagli delle policy relative alle richieste FAS, consulta la pagina Forward access sessions.

Ruoli di servizio per Amazon RDS

Supporta i ruoli di servizio	Sì

Un ruolo di servizio è un ruolo IAM che un servizio assume per eseguire operazioni per tuo conto. Un amministratore IAM può creare, modificare ed eliminare un ruolo di servizio dall'interno di IAM. Per ulteriori informazioni, consulta la sezione Creazione di un ruolo per delegare le autorizzazioni a un Servizio AWS nella Guida per l'utente di IAM.

> **Warning**
>
> La modifica delle autorizzazioni per un ruolo di servizio potrebbe compromettere la funzionalità di Amazon RDS. Modificare i ruoli di servizio solo quando Amazon RDS fornisce le indicazioni per farlo.

Ruoli collegati ai servizi per Amazon RDS

Supporta i ruoli collegati ai servizi	Sì

Un ruolo collegato al servizio è un tipo di ruolo di servizio collegato a un. Servizio AWS Il servizio può assumere il ruolo per eseguire un'azione per tuo conto. I ruoli collegati al servizio vengono visualizzati nel tuo account Account AWS e sono di proprietà del servizio. Un amministratore IAM può visualizzare le autorizzazioni per i ruoli collegati ai servizi, ma non modificarle.

Per ulteriori informazioni sull'utilizzo di ruoli collegati ai servizi Amazon RDS, consulta Utilizzo di ruoli collegati ai servizi per Amazon RDS.

Esempi di policy di Amazon RDS basate su identità

Per impostazione predefinita, i set di autorizzazioni e i ruoli non dispongono dell'autorizzazione per creare o modificare risorse Amazon RDS. Inoltre, non possono eseguire attività utilizzando l'

AWS API AWS Management Console AWS CLI, o. Un amministratore deve creare policy IAM che concedono a set di autorizzazioni e ruoli l'autorizzazione per eseguire operazioni API specifiche sulle risorse specifiche di cui hanno bisogno. L'amministratore deve quindi collegare queste policy ai set di autorizzazioni o ai ruoli che richiedono tali autorizzazioni.

Per informazioni su come creare una policy basata su identità IAM utilizzando questi documenti di policy JSON di esempio, consultare Creazione di policy nella scheda JSON nella Guida per l'utente di IAM.

Argomenti

Best practice delle policy

Le policy basate su identità determinano se qualcuno può creare, accedere o eliminare risorse Amazon RDS nell'account. Queste azioni possono comportare costi aggiuntivi per l' Account AWS. Quando crei o modifichi policy basate su identità, segui queste linee guida e raccomandazioni:

- Inizia con le policy AWS gestite e passa alle autorizzazioni con privilegi minimi: per iniziare a concedere autorizzazioni a utenti e carichi di lavoro, utilizza le politiche AWS gestite che concedono le autorizzazioni per molti casi d'uso comuni. Sono disponibili nel tuo. Account AWS

Ti consigliamo di ridurre ulteriormente le autorizzazioni definendo politiche gestite dai AWS clienti specifiche per i tuoi casi d'uso. Per ulteriori informazioni, consulta Policy gestite da AWS o Policy gestite da AWS per le funzioni dei processi nella Guida per l'utente IAM.

- Applica le autorizzazioni con privilegi minimi: quando imposti le autorizzazioni con le policy IAM, concedi solo le autorizzazioni richieste per eseguire un'attività. Puoi farlo definendo le azioni che possono essere intraprese su risorse specifiche in condizioni specifiche, note anche come autorizzazioni con privilegi minimi. Per ulteriori informazioni sull'utilizzo di IAM per applicare le autorizzazioni, consulta Policy e autorizzazioni in IAM nella Guida per l'utente di IAM.
- Condizioni d'uso nelle policy IAM per limitare ulteriormente l'accesso: per limitare l'accesso ad azioni e risorse puoi aggiungere una condizione alle tue policy. Ad esempio, è possibile scrivere una condizione di policy per specificare che tutte le richieste devono essere inviate utilizzando SSL. Puoi anche utilizzare le condizioni per concedere l'accesso alle azioni del servizio se vengono utilizzate tramite uno specifico Servizio AWS, ad esempio AWS CloudFormation. Per ulteriori informazioni, consulta la sezione Elementi delle policy JSON di IAM: condizione nella Guida per l'utente di IAM.
- Utilizzo di IAM Access Analyzer per convalidare le policy IAM e garantire autorizzazioni sicure e funzionali: IAM Access Analyzer convalida le policy nuove ed esistenti in modo che aderiscano alla sintassi della policy IAM (JSON) e alle best practice di IAM. IAM Access Analyzer offre oltre 100 controlli delle policy e consigli utili per creare policy sicure e funzionali. Per ulteriori informazioni, consulta Convalida delle policy per IAM Access Analyzer nella Guida per l'utente di IAM.
- Richiedi l'autenticazione a più fattori (MFA): se hai uno scenario che richiede utenti IAM o un utente root nel Account AWS tuo, attiva l'MFA per una maggiore sicurezza. Per richiedere la MFA quando vengono chiamate le operazioni API, aggiungi le condizioni MFA alle policy. Per ulteriori informazioni, consulta Configurazione dell'accesso alle API protetto con MFA nella Guida per l'utente di IAM.

Per maggiori informazioni sulle best practice in IAM, consulta Best practice di sicurezza in IAM nella Guida per l'utente di IAM.

Utilizzo della console di Amazon RDS

Per accedere alla console Amazon RDS, è necessario disporre di un set di autorizzazioni minimo. Queste autorizzazioni devono consentirti di elencare e visualizzare i dettagli sulle risorse Amazon RDS presenti nel tuo. Account AWS Se crei una policy basata sull'identità più restrittiva rispetto alle autorizzazioni minime richieste, la console non funzionerà nel modo previsto per le entità (utenti o ruoli) associate a tale policy.

Non è necessario consentire autorizzazioni minime per la console per gli utenti che effettuano chiamate solo verso o l' AWS CLI API. AWS Al contrario, è possibile accedere solo alle operazioni che soddisfano l'operazione API che si sta cercando di eseguire.

Per garantire che tali entità possano ancora utilizzare la console Amazon RDS , allega anche la AWS seguente policy gestita alle entità.

```
AmazonRDSReadOnlyAccess
```

Per ulteriori informazioni, consulta Aggiunta di autorizzazioni a un utente nella Guida per l'utente IAM.

Consentire agli utenti di visualizzare le loro autorizzazioni

Questo esempio mostra in che modo è possibile creare una policy che consente agli utenti IAM di visualizzare le policy inline e gestite che sono cpllegate alla relativa identità utente. Questa policy include le autorizzazioni per completare questa azione sulla console o utilizzando programmaticamente l'API o. AWS CLI AWS

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
```

```
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

Consenti a un utente di creare istanze DB in un account AWS

Di seguito è riportato un esempio di policy che consente all'utente con l'ID di `123456789012` creare istanze DB per il tuo AWS account. La policy richiede che il nome della nuova istanza database inizi con `test`. La nuova istanza database deve anche utilizzare il motore del database MySQL e la classe di istanza database `db.t2.micro`. Inoltre, la nuova istanza database deve utilizzare un gruppo di opzioni e un gruppo di parametri database che inizia con `default`, e deve utilizzare il gruppo di sottoreti `default`.

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "AllowCreateDBInstanceOnly",
         "Effect": "Allow",
         "Action": [
            "rds:CreateDBInstance"
         ],
         "Resource": [
            "arn:aws:rds:*:123456789012:db:test*",
            "arn:aws:rds:*:123456789012:og:default*",
            "arn:aws:rds:*:123456789012:pg:default*",
            "arn:aws:rds:*:123456789012:subgrp:default"
         ],
         "Condition": {
            "StringEquals": {
               "rds:DatabaseEngine": "mysql",
               "rds:DatabaseClass": "db.t2.micro"
            }
         }
      }
   ]
}
```

La policy include una singola istruzione che specifica le autorizzazioni seguenti per l'utente :

- La policy consente all'utente di creare un'istanza DB utilizzando l'operazione API CreateDBInstance (ciò vale anche per create-db-instance AWS CLI il comando e il). AWS Management Console
- L'elemento `Resource` specifica che l'utente può eseguire azioni in o con altre risorse. Specifica le risorse usando Amazon Resource Name (ARN). Questo ARN include il nome del servizio a cui appartiene la risorsa (`rds`), la AWS regione (`*`indica qualsiasi regione in questo esempio), il numero di AWS account (`123456789012`è il numero di account in questo esempio) e il tipo di risorsa. Per ulteriori informazioni sulla creazione di ARN, consulta Utilizzo di Amazon Resource Name (ARN) in Amazon RDS.

  L'elemento `Resource` nell'esempio specifica i seguenti vincoli della policy sulle risorse per l'utente:

  - L'identificatore istanze DB per la nuova istanza database deve iniziare con `test` (per esempio, `testCustomerData1`, `test-region2-data`).
  - Il gruppo di opzioni per la nuova istanza database deve iniziare con `default`.
  - Il gruppo di parametri database per la nuova istanza database deve iniziare con `default`.
  - Il gruppo di sottoreti per la nuova istanza database deve essere il gruppo di sottoreti `default`.
- L'elemento `Condition` specifica che il motore database deve essere MySQL e la classe di istanza database deve essere `db.t2.micro`. L'elemento `Condition` specifica le condizioni quando deve essere applicata una policy. È possibile aggiungere permessi o restrizioni aggiuntivi usando l'elemento `Condition`. Per ulteriori informazioni su come specificare le condizioni;, consulta Chiavi di condizione delle policy per Amazon RDS. Questo esempio specifica le condizioni `rds:DatabaseEngine` e `rds:DatabaseClass`. Per informazioni sui valori di condizione validi per `rds:DatabaseEngine`, consultare l'elenco nel parametro`Engine` in CreateDBInstance. Per informazioni sui valori di condizione validi per `rds:DatabaseClass`, consulta Motori DB supportati per classi di istanza database .

La policy non specifica l'elemento `Principal` poiché in una policy basata su identità l'entità che ottiene l'autorizzazione non viene specificata. Quando si collega una policy a un utente, quest'ultimo è l'entità implicita. Quando si collega una policy di autorizzazione a un ruolo IAM;, l'entità identificata nella policy di attendibilità del ruolo ottiene le autorizzazioni.

Per visualizzare un elenco di operazioni di Amazon RDS, consulta Operazioni definite da Amazon RDS nella Service Authorization Reference.

Autorizzazioni necessarie per l'uso della console

Affinché un utente possa utilizzare la console, è necessario che disponga di un set minimo di autorizzazioni. Queste autorizzazioni consentono all'utente di descrivere le risorse Amazon RDS Amazon per il AWS proprio account e di fornire altre informazioni correlate, tra cui la sicurezza e le informazioni di rete di Amazon EC2.

Se decidi di creare una policy IAM più restrittiva delle autorizzazioni minime richieste, la console non funzionerà come previsto per gli utenti con tale policy IAM. Per garantire che gli utenti possano continuare a usare la console, collega anche la policy gestita `AmazonRDSReadOnlyAccess` all'utente, come descritto in Gestione dell'accesso con policy.

Non sono necessarie le autorizzazioni minime della console per gli utenti che effettuano chiamate solo a AWS CLI o all'API di Amazon RDS.

La seguente politica garantisce l'accesso completo a tutte le risorse Amazon RDS per l'account root: AWS

```
AmazonRDSFullAccess
```

Consentire a un utente di eseguire qualsiasi operazione Describe in qualsiasi risorsa RDS

La seguente policy di autorizzazione concede a un utente le autorizzazioni per eseguire tutte le operazioni che iniziano con `Describe`. Queste operazioni riportano informazioni su una risorsa RDS, ad esempio un'istanza database. Il carattere jolly (*) nell'elemento `Resource` indica che le operazioni sono permesse per tutte le risorse Amazon RDS di proprietà dell'account.

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "AllowRDSDescribe",
         "Effect": "Allow",
         "Action": "rds:Describe*",
         "Resource": "*"
      }
   ]
```

```
}
```

Consentire a un utente di creare un'istanza database che utilizza il gruppo parametri del database e il gruppo di sottorete specificati

La seguente policy di autorizzazioni concede le autorizzazioni per consentire a un utente di creare un'istanza database che deve usare il gruppo di parametri database `mydbpg` e il gruppo di sottorete DB `mydbsubnetgroup`.

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "VisualEditor0",
         "Effect": "Allow",
         "Action": "rds:CreateDBInstance",
         "Resource": [
            "arn:aws:rds:*:*:pg:mydbpg",
            "arn:aws:rds:*:*:subgrp:mydbsubnetgroup"
         ]
      }
   ]
}
```

Concedere l'autorizzazione per le operazioni su una risorsa con un tag specifico con due valori diversi

Puoi utilizzare le condizioni nella policy basata sulle identità per controllare l'accesso alle risorse di Amazon RDS in base ai tag. La seguente policy concede l'autorizzazione per eseguire l'operazione API `CreateDBSnapshot` sulle istanze database con il tag `stage` impostato su `development` o `test`.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"AllowAnySnapshotName",
         "Effect":"Allow",
         "Action":[
            "rds:CreateDBSnapshot"
         ],
```

```
         "Resource":"arn:aws:rds:*:123456789012:snapshot:*"
      },
      {
         "Sid":"AllowDevTestToCreateSnapshot",
         "Effect":"Allow",
         "Action":[
            "rds:CreateDBSnapshot"
         ],
         "Resource":"arn:aws:rds:*:123456789012:db:*",
         "Condition":{
            "StringEquals":{
                "rds:db-tag/stage":[
                  "development",
                  "test"
               ]
            }
         }
      }
   ]
}
```

La seguente policy concede l'autorizzazione per eseguire l'operazione API `ModifyDBInstance` sulle istanze database con il tag `stage` impostato su `development` o `test`.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"AllowChangingParameterOptionSecurityGroups",
         "Effect":"Allow",
         "Action":[
            "rds:ModifyDBInstance"
         ],
         "Resource": [
            "arn:aws:rds:*:123456789012:pg:*",
            "arn:aws:rds:*:123456789012:secgrp:*",
            "arn:aws:rds:*:123456789012:og:*"
         ]
      },
      {
         "Sid":"AllowDevTestToModifyInstance",
         "Effect":"Allow",
         "Action":[
```

```
            "rds:ModifyDBInstance"
         ],
         "Resource":"arn:aws:rds:*:123456789012:db:*",
         "Condition":{
            "StringEquals":{
                "rds:db-tag/stage":[
                  "development",
                  "test"
               ]
            }
         }
      }
   ]
}
```

Impedire a un utente di eliminare un'istanza database

La seguente policy di autorizzazione assegna le autorizzazioni per impedire a un utente di eliminare un'istanza database specifica. Ad esempio, potresti voler negare la possibilità di eliminare le istanze database di produzione a qualsiasi utente che non sia un amministratore.

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "DenyDelete1",
         "Effect": "Deny",
         "Action": "rds:DeleteDBInstance",
         "Resource": "arn:aws:rds:us-west-2:123456789012:db:my-mysql-instance"
      }
   ]
}
```

Negare tutti gli accessi a una risorsa

È anche possibile negare esplicitamente l'accesso a una risorsa. I criteri di negazione hanno la precedenza sui criteri di autorizzazione. La policy seguente nega esplicitamente a un utente la possibilità di gestire una risorsa:

```
{
```

```
   "Version": "2012-10-17",
   "Statement": [
      {
         "Effect": "Deny",
         "Action": "rds:*",
         "Resource": "arn:aws:rds:us-east-1:123456789012:db:mydb"
      }
   ]
}
```

Policy di esempio: Utilizzo di chiavi di condizione

Di seguito sono illustrati esempi di come è possibile utilizzare le chiavi di condizione nelle policy di autorizzazioni IAM di Amazon RDS.

Esempio 1: Concessione dell'autorizzazione per creare un'istanza database che utilizza un motore del database specifico e non è Multi-AZ

La policy seguente utilizza una chiave di condizione RDS e permette a un utente di creare solo istanze database che utilizzano il motore del database MySQL e non utilizzano Multi-AZ. L'elemento `Condition` indica il requisito secondo cui il motore del database deve essere MySQL.

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "AllowMySQLCreate",
         "Effect": "Allow",
         "Action": "rds:CreateDBInstance",
         "Resource": "*",
         "Condition": {
            "StringEquals": {
               "rds:DatabaseEngine": "mysql"
            },
            "Bool": {
               "rds:MultiAz": false
            }
         }
      }
   ]
}
```

Esempio 2: Rifiuto esplicito dell'autorizzazione per creare istanze database per determinate classi di istanze database e per creare istanze database che utilizzano Provisioned IOPS

La policy seguente rifiuta in modo esplicito l'autorizzazione per creare istanze database che utilizzano le classi di istanze database `r3.8xlarge` e `m4.10xlarge`, che sono le istanze database di dimensioni maggiori e più costose. Questa policy impedisce anche gli utenti di creare istanze database che utilizzano Provisioned IOPS, che comporta costi aggiuntivi.

Il rifiuto esplicito di un'autorizzazione prevale su qualsiasi altra autorizzazione concessa. In questo modo si evita che le identità ottengano accidentalmente un'autorizzazione che non desideravi concedere.

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "DenyLargeCreate",
         "Effect": "Deny",
         "Action": "rds:CreateDBInstance",
         "Resource": "*",
         "Condition": {
            "StringEquals": {
               "rds:DatabaseClass": [
                  "db.r3.8xlarge",
                  "db.m4.10xlarge"
               ]
            }
         }
      },
      {
         "Sid": "DenyPIOPSCreate",
         "Effect": "Deny",
         "Action": "rds:CreateDBInstance",
         "Resource": "*",
         "Condition": {
            "NumericNotEquals": {
               "rds:Piops": "0"
            }
         }
      }
   ]
}
```

Esempio 3: limita il set di chiavi di tag e valori che possono essere utilizzati per aggiungere tag a una risorsa

La policy seguente utilizza una chiave di condizione RDS che consente l'aggiunta di un tag con la chiave `stage` per essere aggiunta alla risorsa con i valori `test`, `qa` e `production`.

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Action": [
            "rds:AddTagsToResource",
            "rds:RemoveTagsFromResource"
         ],
         "Resource": "*",
         "Condition": {
            "streq": {
               "rds:req-tag/stage": [
                  "test",
                  "qa",
                  "production"
               ]
            }
         }
      }
   ]
}
```

Specifica delle condizioni: Utilizzo di tag personalizzati

Amazon RDS permette di specificare condizioni in una policy IAM utilizzando tag personalizzati.

Ad esempio, supponi di aggiungere alle istanze database un tag denominato `environment` con valori quali `beta`, `staging`, `production` e così via. In questo modo, puoi creare una policy che limita alcuni utenti alle istanze database in base al valore del tag `environment`.

> **Note**
>
> Per gli identificatori di tag personalizzati viene fatta distinzione tra maiuscole e minuscole.

Nella tabella seguente sono elencati gli identificatori di tag RDS che è possibile utilizzare in un elemento `Condition`.

Identificatore di tag RDS	Si applica a
`db-tag`	Istanze database, incluse repliche di lettura
`snapshot-tag`	Snapshot DB
`ri-tag`	Istanze database riservate
`og-tag`	Gruppi di opzioni database
`pg-tag`	Gruppi di parametri database
`subgrp-tag`	Gruppi di sottoreti database
`es-tag`	Abbonamenti a eventi
`cluster-tag`	Cluster database
`cluster-pg-tag`	Gruppi di parametri di cluster database
`cluster-snapshot-tag`	Snapshot cluster database

La sintassi per una condizione di un tag personalizzato è la seguente:

```
"Condition":{"StringEquals":{"rds:rds-tag-identifier/tag-name":
["value"]} }
```

Ad esempio, l'elemento `Condition` seguente si applica alle istanze database con un tag denominato `environment` e un valore di tag corrispondente a `production`.

```
"Condition":{"StringEquals":{"rds:db-tag/environment": ["production"]} }
```

Per informazioni sulla creazione di tag, consulta Tagging delle risorse Amazon RDS.

> **Important**
>
> Se gestisci l'accesso alle risorse RDS tramite tagging, è consigliabile proteggere l'accesso ai tag per le risorse RDS. È possibile gestire l'accesso ai tag creando policy per le operazioni

`AddTagsToResource` e `RemoveTagsFromResource`. Ad esempio, la policy seguente nega agli utenti la possibilità di aggiungere o rimuovere tag per tutte le risorse. Puoi quindi creare policy per permettere a utenti specifici di aggiungere o rimuovere tag.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"DenyTagUpdates",
         "Effect":"Deny",
         "Action":[
            "rds:AddTagsToResource",
            "rds:RemoveTagsFromResource"
         ],
         "Resource":"*"
      }
   ]
}
```

Per visualizzare un elenco di operazioni di Amazon RDS, consulta Operazioni definite da Amazon RDS nella Service Authorization Reference.

Policy di esempio: Utilizzo di tag personalizzati

Di seguito sono illustrati esempi di come è possibile utilizzare i tag personalizzati nelle policy di autorizzazioni IAM di Amazon RDS. Per ulteriori informazioni sull'aggiunta di tag a una risorsa Amazon RDS, consulta Utilizzo di Amazon Resource Name (ARN) in Amazon RDS.

> **Note**
>
> Tutti gli esempi utilizzano la regione us-west-2 e contengono ID account fittizi.

Esempio 1: Concessione dell'autorizzazione per le operazioni su una risorsa con un tag specifico con due valori diversi

La seguente policy concede l'autorizzazione per eseguire l'operazione API `CreateDBSnapshot` sulle istanze database con il tag `stage` impostato su `development` o `test`.

```
{
```

```
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"AllowAnySnapshotName",
         "Effect":"Allow",
         "Action":[
            "rds:CreateDBSnapshot"
         ],
         "Resource":"arn:aws:rds:*:123456789012:snapshot:*"
      },
      {
         "Sid":"AllowDevTestToCreateSnapshot",
         "Effect":"Allow",
         "Action":[
            "rds:CreateDBSnapshot"
         ],
         "Resource":"arn:aws:rds:*:123456789012:db:*",
         "Condition":{
            "StringEquals":{
                "rds:db-tag/stage":[
                  "development",
                  "test"
               ]
            }
         }
      }
   ]
}
```

La seguente policy concede l'autorizzazione per eseguire l'operazione API `ModifyDBInstance` sulle istanze database con il tag `stage` impostato su `development` o `test`.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"AllowChangingParameterOptionSecurityGroups",
         "Effect":"Allow",
         "Action":[
            "rds:ModifyDBInstance"
         ],
         "Resource":" [
            "arn:aws:rds:*:123456789012:pg:*",
```

```
            "arn:aws:rds:*:123456789012:secgrp:*",
            "arn:aws:rds:*:123456789012:og:*"
         ]
      },
      {
         "Sid":"AllowDevTestToModifyInstance",
         "Effect":"Allow",
         "Action":[
            "rds:ModifyDBInstance"
         ],
         "Resource":"arn:aws:rds:*:123456789012:db:*",
         "Condition":{
            "StringEquals":{
                "rds:db-tag/stage":[
                  "development",
                  "test"
               ]
            }
         }
      }
   ]
}
```

Esempio 2: Rifiuto esplicito dell'autorizzazione per creare un'istanza database che utilizza gruppi di parametri database specificati

La policy seguente rifiuta in modo esplicito l'autorizzazione per creare un'istanza database che utilizza gruppi di parametri database con valori di tag specifici. Puoi applicare questa policy se desideri che uno specifico gruppo di parametri database creato dal cliente venga sempre utilizzato nella creazione di istanze database. Le policy che utilizzano `Deny` servono per lo più a limitare l'accesso concesso da una policy più ampia.

Il rifiuto esplicito di un'autorizzazione prevale su qualsiasi altra autorizzazione concessa. In questo modo si evita che le identità ottengano accidentalmente un'autorizzazione che non desideravi concedere.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
```

```
         "Sid":"DenyProductionCreate",
         "Effect":"Deny",
         "Action":"rds:CreateDBInstance",
         "Resource":"arn:aws:rds:*:123456789012:pg:*",
         "Condition":{
            "StringEquals":{
               "rds:pg-tag/usage":"prod"
            }
         }
      }
   ]
}
```

Esempio 3: Concessione dell'autorizzazione per le operazioni in un'istanza database con un nome di istanza che ha come prefisso un nome utente

La policy seguente concede l'autorizzazione per chiamare qualsiasi API (ad eccezione di `AddTagsToResource` o `RemoveTagsFromResource`) in un'istanza database con un nome di istanza che ha come prefisso il nome dell'utente e che dispone di un tag denominato `stage` equivalente a `devo` o che non dispone di un tag `stage`.

La riga `Resource` nella policy identifica una risorsa in base al relativo Amazon Resource Name (ARN). Per ulteriori informazioni sull'utilizzo di ARN con le risorse Amazon RDS, consulta Utilizzo di Amazon Resource Name (ARN) in Amazon RDS.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"AllowFullDevAccessNoTags",
         "Effect":"Allow",
         "NotAction":[
            "rds:AddTagsToResource",
            "rds:RemoveTagsFromResource"
         ],
         "Resource":"arn:aws:rds:*:123456789012:db:${aws:username}*",
         "Condition":{
            "StringEqualsIfExists":{
               "rds:db-tag/stage":"devo"
            }
         }
      }
```

```
    ]
}
```

AWS politiche gestite per Amazon RDS

Per aggiungere autorizzazioni ai set di autorizzazioni e ai ruoli, è più facile utilizzare politiche AWS gestite piuttosto che scrivere politiche da soli. Creare policy gestite dal cliente IAM per fornire al tuo team solo le autorizzazioni di cui ha bisogno richiede tempo e competenza. Per iniziare rapidamente, puoi utilizzare le nostre politiche AWS gestite. Queste policy coprono i casi d'uso comuni e sono disponibili nel tuo Account AWS. Per ulteriori informazioni sulle policy AWS gestite, consulta le policy AWS gestite nella IAM User Guide.

Servizi AWS mantenere e aggiornare le politiche AWS gestite. Non è possibile modificare le autorizzazioni nelle politiche AWS gestite. I servizi aggiungono occasionalmente autorizzazioni aggiuntive a una policy AWS gestita per supportare nuove funzionalità. Questo tipo di aggiornamento interessa tutte le identità (set di autorizzazioni e ruoli) a cui è collegata la policy. È più probabile che i servizi aggiornino una politica AWS gestita quando viene lanciata una nuova funzionalità o quando diventano disponibili nuove operazioni. I servizi non rimuovono le autorizzazioni da una policy AWS gestita, quindi gli aggiornamenti delle policy non compromettono le autorizzazioni esistenti.

Inoltre, AWS supporta politiche gestite per le funzioni lavorative che si estendono su più servizi. Ad esempio, la policy `ReadOnlyAccess` AWS gestita fornisce l'accesso in sola lettura a tutte le Servizi AWS risorse. Quando un servizio lancia una nuova funzionalità, AWS aggiunge autorizzazioni di sola lettura per nuove operazioni e risorse. Per l'elenco e la descrizione delle policy di funzione dei processi, consulta la sezione Policy gestite da AWS per funzioni di processi nella Guida per l'utente di IAM.

Argomenti

AWS politica gestita: AmazonRDS ReadOnlyAccess

Questa policy consente l'accesso in sola lettura ad Amazon RDS tramite. AWS Management Console

Dettagli dell'autorizzazione

Questa policy include le seguenti autorizzazioni:

- `rds`: consente ai principali di descrivere le risorse Amazon RDS e di elencare i tag per le risorse Amazon RDS.
- `cloudwatch`— Consente ai mandanti di ottenere statistiche sui CloudWatch parametri di Amazon.
- `ec2`: consente ai principali di descrivere le zone di disponibilità e le risorse di rete.
- `logs`— Consente ai responsabili di descrivere CloudWatch Logs, i flussi di log dei gruppi di log e di ottenere gli eventi di log di Logs. CloudWatch
- `devops-guru`— Consente ai responsabili di descrivere le risorse che hanno una copertura Amazon DevOps Guru, specificata dai nomi degli CloudFormation stack o dai tag delle risorse.

Per ulteriori informazioni su questa politica, incluso il documento sulla politica JSON, consulta AmazonRDS ReadOnlyAccess nella Managed Policy Reference Guide. AWS

AWS politica gestita: AmazonRDS FullAccess

Questa policy fornisce l'accesso completo ad Amazon RDS tramite. AWS Management Console

Dettagli dell'autorizzazione

Questa policy include le seguenti autorizzazioni:

- `rds`: consente ai principali l'accesso completo ad Amazon RDS.
- `application-autoscaling`: consente ai principali di descrivere e gestire gli obiettivi e le policy di dimensionamento di Application Auto Scaling.
- `cloudwatch`— Consente ai responsabili di ottenere statistiche CloudWatch metriche e gestire gli allarmi. CloudWatch
- `ec2`: consente ai principali di descrivere le zone di disponibilità e le risorse di rete.
- `logs`— Consente ai responsabili di descrivere CloudWatch Logs, log, flussi di log dei gruppi di log e ottenere gli eventi di log di Logs. CloudWatch
- `outposts`— Consente ai principali di ottenere i tipi di istanza. AWS Outposts

- `pi`: consente ai principali di ottenere i parametri di Performance Insights.
- `sns`: consente ai principali di accedere a iscrizioni e argomenti Amazon Simple Notification Service (Amazon SNS) e di pubblicare messaggi Amazon SNS.
- `devops-guru`— Consente ai responsabili di descrivere le risorse che hanno una copertura Amazon DevOps Guru, specificata dai nomi degli CloudFormation stack o dai tag delle risorse.

Per ulteriori informazioni su questa politica, incluso il documento sulla politica JSON, consulta AmazonRDS FullAccess nella Managed Policy Reference Guide. AWS

AWS politica gestita: AmazonRDS DataFullAccess

Questa politica consente l'accesso completo all'utilizzo della Data API e dell'editor di query sui Aurora Serverless cluster in uno specifico caso. Account AWS Questa politica consente di Account AWS ottenere il valore di un segreto da AWS Secrets Manager.

È possibile allegare la policy `AmazonRDSDataFullAccess`alle identità IAM.

Dettagli dell'autorizzazione

Questa policy include le seguenti autorizzazioni:

- `dbqms`: consente ai principali di accedere, creare, eliminare, descrivere e aggiornare le query. Il Database Query Metadata Service (`dbqms`) è un servizio solo interno. Fornisce le tue query recenti e salvate per l'editor di query su AWS Management Console for multiple Servizi AWS, incluso Amazon RDS.
- `rds-data`: consente ai principali di eseguire istruzioni SQL su database Aurora Serverless.
- `secretsmanager`— Consente ai mandanti di ottenere il valore di un segreto da. AWS Secrets Manager

Per ulteriori informazioni su questa politica, incluso il documento sulla politica JSON, consulta AmazonRDS DataFullAccess nella Managed Policy Reference Guide. AWS

AWS politica gestita: AmazonRDS EnhancedMonitoringRole

Questa policy fornisce l'accesso ad Amazon CloudWatch Logs for Amazon RDS Enhanced Monitoring.

Dettagli dell'autorizzazione

Questa policy include le seguenti autorizzazioni:

- `logs`— Consente ai responsabili di creare CloudWatch Logs, gruppi di log e politiche di conservazione e di creare e descrivere i flussi di log dei gruppi di CloudWatch log. Consente inoltre ai principali di inserire e ottenere eventi di log. CloudWatch

Per ulteriori informazioni su questa politica, incluso il documento sulla politica JSON, consulta AmazonRDS EnhancedMonitoringRole nella Managed Policy Reference Guide. AWS

AWS politica gestita: AmazonRDS PerformanceInsightsReadOnly

Questa policy fornisce accesso in sola lettura alle istanze Amazon RDS Performance Insights per istanze database Amazon RDS e cluster di database Amazon Aurora.

Questa policy ora include `Sid` (ID istruzione) come identificativo per l'istruzione della policy.

Dettagli dell'autorizzazione

Questa policy include le seguenti autorizzazioni:

- `rds`: consente ai principali di descrivere le istanze database Amazon RDS e i cluster di database Amazon Aurora.
- `pi`: consente ai principali di effettuare chiamate all'API Amazon RDS Performance Insights e accedere ai parametri di Performance Insights.

Per ulteriori informazioni su questa politica, incluso il documento sulla politica JSON, consulta AmazonRDS PerformanceInsightsReadOnly nella Managed Policy Reference Guide. AWS

AWS politica gestita: AmazonRDS PerformanceInsightsFullAccess

Questa policy fornisce l'accesso completo ad Approfondimenti sulle prestazioni di Amazon RDS per istanze database Amazon RDS e cluster database Amazon Aurora.

Questa policy ora include `Sid` (ID istruzione) come identificativo per l'istruzione della policy.

Dettagli dell'autorizzazione

Questa policy include le seguenti autorizzazioni:

- `rds`: consente ai principali di descrivere le istanze database Amazon RDS e i cluster di database Amazon Aurora.

- `pi`: consente ai principali di effettuare chiamate all'API Approfondimenti sulle prestazioni di Amazon RDS e di creare, visualizzare ed eliminare report di analisi delle prestazioni.
- `cloudwatch`— Consente ai responsabili di elencare tutte le CloudWatch metriche di Amazon e ottenere dati e statistiche sui parametri.

Per ulteriori informazioni su questa politica, incluso il documento sulla politica JSON, consulta AmazonRDS PerformanceInsightsFullAccess nella Managed Policy Reference Guide. AWS

AWS politica gestita: AmazonRDS DirectoryServiceAccess

Questa policy permette ad Amazon RDS di effettuare chiamate alla AWS Directory Service.

Dettagli dell'autorizzazione

Questa policy include la seguente autorizzazione:

- `ds`— Consente ai responsabili di descrivere le AWS Directory Service directory e di controllarne l'autorizzazione. AWS Directory Service

Per ulteriori informazioni su questa politica, incluso il documento sulla politica JSON, consulta AmazonRDS DirectoryServiceAccess nella Managed Policy Reference Guide. AWS

AWS politica gestita: AmazonRDS ServiceRolePolicy

Non è possibile allegare la policy `AmazonRDSServiceRolePolicy` alle entità IAM. Questa policy è allegata a un ruolo collegato ai servizi che consente ad Amazon RDS di eseguire operazioni per conto dell'utente. Per ulteriori informazioni, consulta Autorizzazioni del ruolo collegato ai servizi per Amazon RDS.

AWS politica gestita: AmazonRDS CustomServiceRolePolicy

Non è possibile allegare la policy `AmazonRDSCustomServiceRolePolicy` alle entità IAM. Questa policy è allegata a un ruolo collegato ai servizi che consente ad Amazon RDS di eseguire operazioni per conto dell'utente. Per ulteriori informazioni, consulta Autorizzazioni del ruolo collegato ai servizi per Amazon RDS Custom.

AWSpolitica gestita: AmazonRDSCustom Instance ProfileRolePolicy

Non bisogna collegare `AmazonRDSCustomInstanceProfileRolePolicy` alle entità IAM. Deve essere associato solo a un ruolo del profilo di istanza utilizzato per concedere autorizzazioni

all'istanza DB personalizzata di Amazon RDS per eseguire varie azioni di automazione e attività di gestione del database. Passa il profilo dell'istanza come `custom-iam-instance-profile` parametro durante la creazione dell'istanza RDS Custom e RDS Custom associa questo profilo di istanza all'istanza DB.

Dettagli dell'autorizzazione

Questa policy include le seguenti autorizzazioni:

- `ssm`,`ssmmessages`, `ec2messages` - Consente a RDS Custom di comunicare, eseguire l'automazione e gestire gli agenti sull'istanza DB tramite Systems Manager.
- `ec2`, `s3` - Consente a RDS Custom di eseguire operazioni di backup sull'istanza DB che fornisce funzionalità di point-in-time ripristino.
- `secretsmanager`- Consente a RDS Custom di gestire i segreti specifici dell'istanza DB creati da RDS Custom.
- `cloudwatch`, `logs` - Consente a RDS Custom di caricare le metriche e i log delle istanze DB tramite agente. CloudWatch CloudWatch
- `events`, `sqs` - Consente a RDS Custom di inviare e ricevere informazioni sullo stato dell'istanza DB.
- `kms`- Consente a RDS Custom di utilizzare una chiave KMS specifica dell'istanza per eseguire la crittografia dei segreti e degli oggetti S3 gestiti da RDS Custom.

Per ulteriori informazioni su questa politica, incluso il documento sulla policy JSON, consulta AmazonRDSCustom Instance nella Managed Policy Reference Guide. ProfileRolePolicy AWS

Aggiornamenti Amazon RDS alle politiche AWS gestite

Visualizza i dettagli sugli aggiornamenti delle politiche AWS gestite per Amazon RDS da quando questo servizio ha iniziato a tracciare queste modifiche. Per gli avvisi automatici sulle modifiche apportate a questa pagina, sottoscrivere il feed RSS nella pagina di Cronologia dei documenti di Amazon RDS.

Modifica	Descrizione	Data
Autorizzazioni del ruolo collegato ai servizi per Amazon RDS Custom: aggiornamento a policy esistente	Amazon RDS ha aggiunto nuove autorizzazioni alla `AmazonRDSCustomServiceRolePolicy` del ruolo collegato al servizio `AWSServiceRoleForRDSCustom` . Questa nuova autorizzazione consente a RDS Custom di associare un ruolo di servizio come profilo di istanza a un'istanza personalizzata RDS. Per ulteriori informazioni, consulta Autorizzazioni del ruolo collegato ai servizi per Amazon RDS Custom.	19 aprile 2024
AWS politiche gestite per Amazon RDS: aggiornamento a policy esistente	Amazon RDS ha aggiunto una nuova autorizzazione al ruolo `AWSServiceRoleForRDSCustom` collegato al servizio per consentire a RDS Custom for SQL Server di modificare il tipo di istanza host del database sottostante. `AmazonRDSCustomServiceRolePolicy`	8 aprile 2024

Modifica	Descrizione	Data
	RDS ha inoltre aggiunto l'`ec2:DescribeInstanceTypes` autorizzazione a ottenere informazioni sul tipo di istanza per l'host del database. Per ulteriori informazioni, consulta AWS politiche gestite per Amazon RDS.	
AWS politiche gestite per Amazon RDS: nuova policy	Amazon RDS ha aggiunto una nuova policy gestita denominata `AmazonRDSCustomInstanceProfileRolePolicy` per consentire a RDS Custom di eseguire azioni di automazione e attività di gestione del database tramite un profilo di istanza EC2. Per ulteriori informazioni, consulta AWS politiche gestite per Amazon RDS.	27 febbraio 2024
Autorizzazioni del ruolo collegato ai servizi per Amazon RDS: aggiornamento a una policy esistente	Amazon RDS ha aggiunto nuovi ID `AmazonRDSServiceRolePolicy` di dichiarazione al ruolo collegato al `AWSServiceRoleForRDS` servizio. Per ulteriori informazioni, consulta Autorizzazioni del ruolo collegato ai servizi per Amazon RDS.	19 gennaio 2024

Modifica	Descrizione	Data
AWS politiche gestite per Amazon RDS: aggiornamento a policy esistenti	Le policy gestite `AmazonRDSPerformanceInsightsReadOnly` e `AmazonRDSPerformanceInsightsFullAccess` ora includono `Sid` (ID istruzione) come identificativo nell'istruzione della policy. Per ulteriori informazioni, consulta AWS politica gestita: AmazonRDS PerformanceInsightsReadOnly e AWS politica gestita: AmazonRDS PerformanceInsightsFullAccess.	23 ottobre 2023
Autorizzazioni del ruolo collegato ai servizi per Amazon RDS: aggiornamento a una policy esistente	Amazon RDS ha aggiunto nuove autorizzazioni alla `AmazonRDSCustomServiceRolePolicy` del ruolo collegato al servizio `AWSServiceRoleForRDSCustom`. Queste nuove autorizzazioni consentono a RDS Custom for Oracle di creare, modificare ed eliminare Managed Rules. EventBridge Per ulteriori informazioni, consulta Autorizzazioni del ruolo collegato ai servizi per Amazon RDS Custom.	20 settembre 2023

Modifica	Descrizione	Data
AWS politiche gestite per Amazon RDS: aggiornamento a policy esistente	Amazon RDS ha aggiunto nuove autorizzazioni alla policy gestita `AmazonRDSFullAccess`. Le autorizzazioni consentono di generare, visualizzare ed eliminare il report di analisi delle prestazioni per un periodo di tempo. Per ulteriori informazioni sulla configurazione delle policy di accesso per Performance Insights, consulta Configurazione delle policy di accesso per Performance Insights	17 agosto 2023

Modifica	Descrizione	Data
AWS politiche gestite per Amazon RDS: nuova policy e aggiornamento a una policy esistente	Amazon RDS ha aggiunto nuove autorizzazioni alla policy gestita `AmazonRDSPerformanceInsightsReadOnly` e una nuova policy gestita denominata `AmazonRDSPerformanceInsightsFullAccess`. Queste autorizzazioni consentono di analizzare Performance Insights per un periodo di tempo, visualizzare i risultati dell'analisi insieme ai suggerimenti ed eliminare i report. Per ulteriori informazioni sulla configurazione delle policy di accesso per Performance Insights, consulta Configurazione delle policy di accesso per Performance Insights	16 agosto 2023

Modifica	Descrizione	Data
Autorizzazioni del ruolo collegato ai servizi per Amazon RDS: aggiornamento a una policy esistente	Amazon RDS ha aggiunto nuove autorizzazioni alla `AmazonRDSCustomServiceRolePolicy` del ruolo collegato al servizio `AWSServiceRoleForRDSCustom` . Queste nuove autorizzazioni consentono a RDS Custom per Oracle di utilizzare gli snapshot DB. Per ulteriori informazioni, consulta Autorizzazioni del ruolo collegato ai servizi per Amazon RDS Custom.	23 giugno 2023
Autorizzazioni del ruolo collegato ai servizi per Amazon RDS: aggiornamento a una policy esistente	Amazon RDS ha aggiunto nuove autorizzazioni alla `AmazonRDSCustomServiceRolePolicy` del ruolo collegato al servizio `AWSServiceRoleForRDSCustom` . Queste nuove autorizzazioni consentono a RDS Custom per Oracle di utilizzare gli snapshot DB. Per ulteriori informazioni, consulta Autorizzazioni del ruolo collegato ai servizi per Amazon RDS Custom.	23 giugno 2023

Modifica	Descrizione	Data
Autorizzazioni del ruolo collegato ai servizi per Amazon RDS: aggiornamento a una policy esistente	Amazon RDS ha aggiunto nuove autorizzazioni alla `AmazonRDSCustomServiceRolePolicy` del ruolo collegato al servizio `AWSServiceRoleForRDSCustom` . Queste nuove autorizzazioni consentono a RDS Custom di creare interfacce di rete. Per ulteriori informazioni, consulta Autorizzazioni del ruolo collegato ai servizi per Amazon RDS Custom.	30 maggio 2023
Autorizzazioni del ruolo collegato ai servizi per Amazon RDS: aggiornamento a una policy esistente	Amazon RDS ha aggiunto nuove autorizzazioni alla `AmazonRDSCustomServiceRolePolicy` del ruolo collegato al servizio `AWSServiceRoleForRDSCustom` . Queste nuove autorizzazioni consentono a RDS Custom di chiamare Amazon EBS per verificare la quota di archiviazione. Per ulteriori informazioni, consulta Autorizzazioni del ruolo collegato ai servizi per Amazon RDS Custom.	18 aprile 2023

Modifica	Descrizione	Data
Autorizzazioni del ruolo collegato ai servizi per Amazon RDS: aggiornamento a una policy esistente	Amazon RDS Custom ha aggiunto nuove autorizzazioni alla policy `AmazonRDSCustomServiceRolePolicy` del ruolo collegato al servizio `AWSServiceRoleForRDSCustom` per l'integrazione con Amazon SQS. RDS Custom richiede l'integrazione con Amazon SQS per creare e gestire le code SQS nell'account del cliente. I nomi delle code SQS sono conformi al formato `do-not-delete-rds-custom-[identifier]` e sono contrassegnati con`Amazon RDS Custom`. È stata aggiunta anche l'autorizzazione per `ec2:CreateSnapshot` per consentire a RDS Custom di creare backup per i volumi collegati all'istanza. Per ulteriori informazioni, consulta Autorizzazioni del ruolo collegato ai servizi per Amazon RDS Custom.	6 aprile 2023

Modifica	Descrizione	Data
AWS politiche gestite per Amazon RDS: aggiornamento a una policy esistente	Amazon RDS ha aggiunto un nuovo spazio dei CloudWatch nomi `ListMetrics` Amazon a and. `AmazonRDSFullAccess` `AmazonRDSReadOnlyAccess` Questo spazio dei nomi è necessario affinché Amazon RDS elenchi specifici parametri di utilizzo delle risorse. Per ulteriori informazioni, consulta Panoramica della gestione delle autorizzazioni di accesso alle tue CloudWatch risorse nella Amazon CloudWatch User Guide.	4 aprile 2023

Modifica	Descrizione	Data
AWS politiche gestite per Amazon RDS: aggiornamento a una policy esistente	Amazon RDS ha aggiunto nuove autorizzazioni `AmazonRDSFullAccess` e politiche `AmazonRDSReadOnlyAccess` gestite per consentire la visualizzazione dei risultati di Amazon DevOps Guru nella console RDS. Questa autorizzazione è necessaria per consentire la visualizzazione dei risultati di Guru. DevOps Per ulteriori informazioni, consulta Amazon RDS updates to AWS managed policy.	30 marzo 2023

Modifica	Descrizione	Data
Autorizzazioni del ruolo collegato ai servizi per Amazon RDS: aggiornamento a una policy esistente	Amazon RDS ha aggiunto nuove autorizzazioni al ruolo `AWSServiceRoleForRDS` collegato al `AmazonRDSServiceRolePolicy` servizio per l'integrazione con. AWS Secrets Manager RDS richiede l'integrazione con Secrets Manager per la gestione delle password degli utenti master in Secrets Manager. Il segreto utilizza una convenzione di denominazione riservata e limita gli aggiornamenti del cliente. Per ulteriori informazioni, consulta Gestione delle password con Amazon RDS e AWS Secrets Manager.	22 dicembre 2022

Modifica	Descrizione	Data
Autorizzazioni del ruolo collegato ai servizi per Amazon RDS: aggiornamento a una policy esistente	Amazon RDS ha aggiunto nuove autorizzazioni alla `AmazonRDSCustomServiceRolePolicy` del ruolo collegato al servizio `AWSServiceRoleForRDSCustom` . RDS Custom supporta i cluster database. Queste nuove autorizzazioni nella politica consentono a RDS Custom di effettuare chiamate per Servizi AWS conto dei cluster DB. Per ulteriori informazioni, consulta Autorizzazioni del ruolo collegato ai servizi per Amazon RDS Custom.	9 novembre 2022

Modifica	Descrizione	Data
Autorizzazioni del ruolo collegato ai servizi per Amazon RDS: aggiornamento a una policy esistente	Amazon RDS ha aggiunto nuove autorizzazioni al ruolo collegato al servizio `AWSServiceRoleForRDS` per l'integrazione con AWS Secrets Manager. L'integrazione con Secrets Manager è necessaria per il corretto funzionamento di SQL Server Reporting Services (SSRS) Email su RDS. SSRS Email crea un segreto per conto del cliente. Il segreto utilizza una convenzione di denominazione riservata e limita gli aggiornamenti del cliente. Per ulteriori informazioni, consulta Utilizzo di SSRS Email per inviare report.	26 agosto 2022

Modifica	Descrizione	Data
Autorizzazioni del ruolo collegato ai servizi per Amazon RDS: aggiornamento a una policy esistente	Amazon RDS ha aggiunto un nuovo spazio dei CloudWatch nomi Amazon a for. `AmazonRDSPreviewServiceRolePolicy` `PutMetricData` Questo spazio dei nomi è necessario affinché Amazon RDS pubblichi i parametri di utilizzo delle risorse. Per ulteriori informazioni, consulta Usare le chiavi di condizione per limitare l'accesso ai CloudWatch namespace nella Amazon CloudWatch User Guide.	7 luglio 2022

Modifica	Descrizione	Data
Autorizzazioni del ruolo collegato ai servizi per Amazon RDS: aggiornamento a una policy esistente	Amazon RDS ha aggiunto un nuovo spazio dei CloudWatch nomi Amazon a for. `AmazonRDSBetaServiceRolePolicy` `PutMetricData` Questo spazio dei nomi è necessario affinché Amazon RDS pubblichi i parametri di utilizzo delle risorse. Per ulteriori informazioni, consulta Usare le chiavi di condizione per limitare l'accesso ai CloudWatch namespace nella Amazon CloudWatch User Guide.	7 luglio 2022
Autorizzazioni del ruolo collegato ai servizi per Amazon RDS: aggiornamento a una policy esistente	Amazon RDS ha aggiunto un nuovo spazio dei CloudWatch nomi Amazon a for. `AWSServiceRoleForRDS` `PutMetricData` Questo spazio dei nomi è necessario affinché Amazon RDS pubblichi i parametri di utilizzo delle risorse. Per ulteriori informazioni, consulta Usare le chiavi di condizione per limitare l'accesso ai CloudWatch namespace nella Amazon CloudWatch User Guide.	22 aprile 2022

Modifica	Descrizione	Data
Autorizzazioni del ruolo collegato ai servizi per Amazon RDS: aggiornamento a una policy esistente	Amazon RDS ha aggiunto nuove autorizzazioni al ruolo collegato al servizio `AWSServiceRoleForRDS` per gestire le autorizzazioni per pool di IP di proprietà del cliente e tabelle di routing del gateway locali (LGW-RTB). Queste autorizzazioni sono necessarie per RDS su Outposts per eseguire la replica Multi-AZ nella rete locale di Outposts. Per ulteriori informazioni, consulta Operare con le implementazioni Multi-AZ per Amazon RDS su AWS Outposts.	19 aprile 2022

Modifica	Descrizione	Data
Policy basate su identità: aggiornamento a una policy esistente	Amazon RDS ha aggiunto una nuova autorizzazione alla policy gestita `AmazonRDSFullAccess` per descrivere le autorizzazioni per LGW-RTB. Questa autorizzazione è necessaria per descrivere le autorizzazioni per RDS su Outposts per eseguire la replica Multi-AZ nella rete locale di Outposts. Per ulteriori informazioni, consulta Operare con le implementazioni Multi-AZ per Amazon RDS su AWS Outposts.	19 aprile 2022
AWS politiche gestite per Amazon RDS: nuova policy	Amazon RDS ha aggiunto una nuova policy gestita denominata `AmazonRDSPerformanceInsightsReadOnly` per consentire ad Amazon RDS di chiamare i AWS servizi per conto delle tue istanze DB. Per ulteriori informazioni sulla configurazione delle policy di accesso per Performance Insights, consulta Configurazione delle policy di accesso per Performance Insights	10 marzo 2022

Modifica	Descrizione	Data
Autorizzazioni del ruolo collegato ai servizi per Amazon RDS: aggiornamento a una policy esistente	Amazon RDS ha aggiunto nuovi CloudWatch namespace Amazon a for. `AWSServiceRoleForRDS` `PutMetricData` Questi namespace sono necessari per Amazon DocumentDB (con compatibilità MongoDB) e Amazon Neptune per pubblicare i parametri. CloudWatch Per ulteriori informazioni, consulta Usare le chiavi di condizione per limitare l'accesso ai CloudWatch namespace nella Amazon CloudWatch User Guide.	4 marzo 2022
Autorizzazioni del ruolo collegato ai servizi per Amazon RDS Custom: nuova policy	Amazon RDS ha aggiunto un nuovo ruolo collegato al servizio denominato `AWSServiceRoleForRDSCustom` per consentire a RDS Custom di invocare Servizi AWS per conto delle istanze database.	26 ottobre 2021
Amazon RDS ha iniziato a monitorare le modifiche	Amazon RDS ha iniziato a tracciare le modifiche per le sue politiche AWS gestite.	26 ottobre 2021

Prevenzione del problema "confused deputy" tra servizi

Il problema confused deputy è un problema di sicurezza in cui un'entità che non dispone dell'autorizzazione per eseguire un'azione può costringere un'entità maggiormente privilegiata a eseguire l'azione. In AWS, la rappresentazione cross-service può comportare il problema confused deputy.

La rappresentazione tra servizi può verificarsi quando un servizio (il servizio chiamante) effettua una chiamata a un altro servizio (il servizio chiamato). Il servizio chiamante può essere manipolato per utilizzare le proprie autorizzazioni e agire sulle risorse di un altro cliente, a cui normalmente non avrebbe accesso. Per evitare ciò, AWS fornisce strumenti per aiutarti a proteggere i tuoi dati per tutti i servizi con entità di servizio a cui è stato concesso l'accesso alle risorse del tuo account. Per ulteriori informazioni, consultare Problema del "confused deputy" nella Guida per l'utente di IAM.

Per limitare le autorizzazioni alle risorse che Amazon RDS fornisce a un altro servizio per una risorsa specifica, si consiglia di utilizzare le chiavi di contesto delle condizioni globali `aws:SourceArn` e `aws:SourceAccount` nelle policy delle risorse.

In alcuni casi, il valore `aws:SourceArn` non contiene l'ID dell'account, ad esempio quando utilizzi Amazon Resource Name (ARN) per un bucket Simple Storage Service (Amazon S3). In questi casi, assicurati di utilizzare entrambe le chiavi di contesto delle condizioni globali per limitare le autorizzazioni. In alcuni casi, si utilizzano le chiavi di contesto delle condizioni globali e il valore `aws:SourceArn` contiene l'ID dell'account. In questi casi, assicurati che il valore `aws:SourceAccount` e l'account in `aws:SourceArn` utilizzano lo stesso ID dell'account quando vengono utilizzati nella stessa dichiarazione di policy. Utilizza `aws:SourceArn` se desideri consentire l'associazione di una sola risorsa all'accesso tra servizi. Utilizza `aws:SourceAccount` se desideri consentire l'associazione di qualsiasi nell'account AWS specificato all'uso tra servizi.

Assicurati che il valore di `aws:SourceArn` sia un ARN per un tipo di risorsa Amazon RDS. Per ulteriori informazioni, consultare Utilizzo di Amazon Resource Name (ARN) in Amazon RDS.

Il modo più efficace per proteggersi dal problema "confused deputy" è quello di usare la chiave di contesto della condizione globale `aws:SourceArn` con l'ARN completo della risorsa. In alcuni casi, potresti non conoscere l'ARN completo della risorsa o potresti specificare più risorse. In questi casi, utilizza la chiave di contesto delle condizioni globali `aws:SourceArn` con caratteri jolly (*) per le parti sconosciute dell'ARN. Un esempio è `arn:aws:rds:*:123456789012:*`.

L'esempio seguente mostra il modo in cui puoi utilizzare le chiavi di contesto `aws:SourceArn` e `aws:SourceAccount` delle condizioni globali in Amazon RDS per prevenire il problema “confused deputy”.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Sid": "ConfusedDeputyPreventionExamplePolicy",
    "Effect": "Allow",
    "Principal": {
      "Service": "rds.amazonaws.com"
    },
    "Action": "sts:AssumeRole",
    "Condition": {
      "ArnLike": {
        "aws:SourceArn": "arn:aws:rds:us-east-1:123456789012:db:mydbinstance"
      },
      "StringEquals": {
        "aws:SourceAccount": "123456789012"
      }
    }
  }
}
```

Per altri esempi di policy che utilizzano le chiavi di contesto delle condizioni globali `aws:SourceArn` e `aws:SourceAccount`, consulta le sezioni seguenti:

- Concessione di autorizzazioni per pubblicare le notifiche in un argomento Amazon SNS
- Creazione manuale di un ruolo IAM per backup e ripristino nativi
- Configurazione dell'autenticazione di Windows per le istanze di database di SQL Server
- Prerequisiti per l'integrazione di RDS for SQL Server con S3
- Creazione manuale di un ruolo IAM per l'audit in SQL Server
- Configurazione delle autorizzazioni IAM per l'integrazione di RDS per Oracle con Amazon S3
- Configurazione dell'accesso a un bucket Amazon S3 (importazione PostgreSQL)
- Configurazione dell'accesso a un bucket Amazon S3 (esportazione PostgreSQL)

Autenticazione del database IAM per MariaDB, MySQL e PostgreSQL

Puoi autenticarti nel tuo di istanze DB utilizzando l'autenticazione del database AWS Identity and Access Management (IAM). L'autenticazione del database IAM funziona con MariaDB, MySQL e PostgreSQL. Con questo metodo di autenticazione, non devi utilizzare una password quando esegui la connessione all'istanza database. Utilizzi invece un token di autenticazione.

Un token di autenticazione è una stringa univoca di caratteri generata da Amazon RDS su richiesta. I token di autenticazione vengono generati utilizzando AWS Signature Version 4. Ciascun token ha un ciclo di vita di 15 minuti. Non devi archiviare le credenziali dell'utente nel database, perché l'autenticazione è gestita esternamente utilizzando IAM. Puoi anche utilizzare ancora l'autenticazione standard del database. Il token viene utilizzato solo per l'autenticazione e non influisce sulla sessione dopo che è stato stabilito.

L'autenticazione del database IAM fornisce i seguenti vantaggi:

- Il traffico di rete da e verso il database viene crittografato utilizzando Secure Socket Layer (SSL) o Transport Layer Security (TLS). Per ulteriori informazioni sull'uso di SSL/TLS con Amazon RDS, consulta
- Puoi usare IAM per gestire in modo centralizzato l'accesso alle risorse del database invece di gestire l'accesso singolarmente in ogni istanza database.
- Per le applicazioni in esecuzione su Amazon EC2, puoi utilizzare le credenziali specifiche dell'istanza EC2 per accedere al database invece di una password, per maggiore sicurezza.

In generale, prendi in considerazione l'utilizzo dell'autenticazione del database IAM quando le applicazioni creano meno di 200 connessioni al secondo e non desideri gestire nomi utente e password direttamente nel codice dell'applicazione.

Il driver JDBC di Amazon Web Services (AWS) supporta l'autenticazione del database IAM. Per ulteriori informazioni, consulta AWS IAM Authentication Plugin nel repository di driver GitHub JDBC di Amazon Web Services (AWS).

Il driver Python di Amazon Web Services (AWS) supporta l'autenticazione del database IAM. Per ulteriori informazioni, consulta AWS IAM Authentication Plugin nel repository Python Driver GitHub di Amazon Web Services (AWS).

Argomenti

- Disponibilità di regioni e versioni

- Supporto per CLI e SDK
- Limitazioni per l'autenticazione database IAM
- Consigli per l'autenticazione del database IAM
- Chiavi di contesto relative alle condizioni globali non supportate AWS
- Abilitazione e disabilitazione dell'autenticazione database IAM
- Creazione e utilizzo di una policy IAM per l'accesso al database IAM
- Creazione di un account database tramite l'autenticazione IAM
- Connessione all'istanza tramite l'autenticazione IAM

Disponibilità di regioni e versioni

Il supporto varia a seconda delle versioni specifiche di ciascun motore di database e a seconda delle Regioni AWS. Per ulteriori informazioni sulla disponibilità di versioni e regioni per Amazon RDS con autenticazione del database IAM, consulta Regioni e motori DB supportati per l'autenticazione del database IAM in Amazon RDS.

Supporto per CLI e SDK

L'autenticazione del database IAM è disponibile per AWS CLIe per i seguenti SDK specifici del linguaggio AWS :

- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for PHP
- AWS SDK for Python (Boto3)
- AWS SDK for Ruby

Limitazioni per l'autenticazione database IAM

Quando utilizzi l'autenticazione database IAM, tieni presenti le seguenti limitazioni:

- Il numero massimo di connessioni al secondo per il di istanze DB potrebbe essere limitato a seconda della classe di istanza database e del carico di lavoro. L'autenticazione IAM può fallire in caso di esaurimento delle risorse durante i picchi di carico del DB.
- Attualmente, l'autenticazione database IAM non supporta nessuna delle chiavi di contesto delle condizioni globali.

  Per ulteriori informazioni sulle chiavi di contesto delle condizioni globali, consulta Chiavi di contesto delle condizioni globali AWS nella Guida per l'utente di IAM.
- Per PostgreSQL, se il ruolo IAM (`rds_iam`) viene aggiunto a un utente (incluso l'utente principale RDS), l'autenticazione IAM ha la precedenza sull'autenticazione tramite password, quindi l'utente deve accedere come un utente IAM.
- Per PostgreSQL, Amazon RDS non supporta l'attivazione dei metodi di autenticazione IAM e Kerberos contemporaneamente.
- Per PostgreSQL, non è possibile utilizzare l'autenticazione IAM per stabilire una connessione di replica.
- Non è possibile utilizzare un record DNS Route 53 personalizzato anziché l'endpoint dell'istanza databaseper generare il token di autenticazione.
- CloudWatch e CloudTrail non registrate l'autenticazione IAM. Questi servizi non tengono traccia delle chiamate `generate-db-auth-token` API che autorizzano il ruolo IAM a abilitare la connessione al database. Per ulteriori informazioni, consulta Raggiungere la verificabilità con l'autenticazione IAM di Amazon RDS utilizzando il controllo degli accessi basato sugli attributi.

Consigli per l'autenticazione del database IAM

Quando si utilizza l'autenticazione del database IAM, è consigliabile procedere come segue:

- Utilizzare l'autenticazione del database IAM quando l'applicazione richiede meno di 200 nuove connessioni di autenticazione del database IAM al secondo.

  I motori di database che funzionano con Amazon RDS non prevedono limitazioni per i tentativi di autenticazione al secondo. Tuttavia, quando utilizzi un'autenticazione database IAM, l'applicazione deve generare un token di autenticazione. L'applicazione usa quindi il token per connettersi all'istanza database. Se eccedi il limite massimo di nuove connessioni al secondo, la gestione extra dell'autenticazione database IAM può causare throttling della connessione.

  Valuta la possibilità di utilizzare il pool di connessioni nelle applicazioni per mitigare la creazione continua di connessioni. Questo può ridurre il sovraccarico derivante dall'autenticazione DB IAM

e consentire alle applicazioni di riutilizzare le connessioni esistenti. In alternativa, per questi casi d'uso considera l'utilizzo di Server proxy per RDS. Per Server proxy per RDS sono previsti costi aggiuntivi. Consulta i prezzi per Server proxy per RDS.

- La dimensione di un token di autenticazione del database IAM dipende da molti fattori, tra cui il numero di tag IAM, le policy di servizio IAM, la lunghezza del nome della risorsa Amazon (ARN) e altre proprietà IAM e del database. La dimensione minima del token è generalmente di circa 1 KB, ma può essere maggiore. Poiché questo token viene utilizzato come password nella stringa di connessione al database tramite l'autenticazione IAM, è necessario assicurarsi che il driver del database (ad esempio ODBC) e/o qualsiasi strumento non limitino o altrimenti tronchino questo token a causa delle sue dimensioni. Un token troncato causa l'esito negativo della convalida dell'autenticazione effettuata dal database e da IAM.
- Se si utilizzano credenziali temporanee durante la creazione di un token di autenticazione del database IAM, le credenziali temporanee devono essere ancora valide quando si utilizza il token di autenticazione del database IAM per effettuare una richiesta di connessione.

Chiavi di contesto relative alle condizioni globali non supportate AWS

L'autenticazione del database IAM non supporta il seguente sottoinsieme di chiavi di contesto delle condizioni AWS globali.

- `aws:Referer`
- `aws:SourceIp`
- `aws:SourceVpc`
- `aws:SourceVpce`
- `aws:UserAgent`
- `aws:VpcSourceIp`

Per ulteriori informazioni, consultare Chiavi di contesto delle condizioni globali AWS nella Guida per l'utente IAM.

Abilitazione e disabilitazione dell'autenticazione database IAM

Per impostazione predefinita, l'autenticazione database IAM è disabilitata nelle istanze database. Puoi abilitare o disabilitare l'autenticazione database IAM tramite la AWS Management Console, l'AWS CLI o l'API.

Puoi abilitare l'autenticazione database IAM quando esegui una delle seguenti operazioni:

- Per creare una nuova istanza database con l'autenticazione database IAM abilitata, consulta Creazione di un'istanza database Amazon RDS.
- Per modificare un'istanza database per abilitare l'autenticazione database IAM, consulta Modifica di un'istanza database Amazon RDS.
- Per ripristinare un'istanza database da uno snapshot con l'autenticazione del database IAM abilitata, consulta Ripristino da uno snapshot database.
- Per ripristinare un'istanza database a un momento specifico con l'autenticazione del database IAM abilitata, consulta Ripristino a un'ora specifica per un'istanza database.

Per l'autenticazione IAM per le istanze database PostgreSQL, il valore SSL deve essere 1. Non è possibile abilitare l'autenticazione IAM per un'istanza database PostgreSQL se il valore SSL è 0. Non è possibile impostare il valore SSL su 0 se l'autenticazione IAM è abilitata per un'istanza database PostgreSQL.

Console

Ogni flusso di lavoro di creazione o modifica include una sezione Database authentication (Autenticazione database) in cui puoi abilitare o disabilitare l'autenticazione database IAM. In questa sezione scegli Password and IAM database authentication (Autenticazione password e database IAM) per abilitare l'autenticazione database IAM.

Per abilitare o disabilitare l'autenticazione database IAM per un'istanza database esistente

1. Apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel pannello di navigazione, scegliere Databases (Database).
3. Scegliere l'istanza database che si vuole modificare.

   > **Note**
   >
   > Verifica che l'istanza database sia compatibile con l'autenticazione IAM. Controllare i requisiti di compatibilità in Disponibilità di regioni e versioni.

4. Scegliere Modify (Modifica).

5. Nella sezione Autenticazione database seleziona Autenticazione database con password e IAM per abilitare l'autenticazione del database IAM. Scegli Autenticazione password o Autenticazione di password e Kerberos per disabilitare l'autenticazione IAM.
6. Scegli Continue (Continua).
7. Per applicare immediatamente le modifiche, scegli Immediately (Immediatamente) nella sezione Scheduling of modifications (Pianificazione delle modifiche).
8. Scegliere Modify DB instance (Modifica istanza database) .

AWS CLI

Per creare una nuova istanza database con autenticazione IAM tramite AWS CLI, utilizzare il comando `create-db-instance`. Specifica l'opzione `--enable-iam-database-authentication`, come visualizzato nell'esempio seguente.

```
aws rds create-db-instance \
    --db-instance-identifier mydbinstance \
    --db-instance-class db.m3.medium \
    --engine MySQL \
    --allocated-storage 20 \
    --master-username masterawsuser \
    --manage-master-user-password \
    --enable-iam-database-authentication
```

Per aggiornare un'istanza database esistente in modo da abilitare o disabilitare l'autenticazione IAM, utilizzare il comando AWS CLI `modify-db-instance`. Specifica l'opzione `--enable-iam-database-authentication` o `--no-enable-iam-database-authentication`, come appropriato.

> **Note**
>
> Verifica che l'istanza database sia compatibile con l'autenticazione IAM. Controllare i requisiti di compatibilità in Disponibilità di regioni e versioni.

Per impostazione predefinita, Amazon RDS esegue la modifica durante la finestra di manutenzione successiva. Se desideri sostituire ciò e abilitare l'autenticazione database IAM il prima possibile, utilizza il parametro `--apply-immediately`.

L'esempio seguente mostra come abilitare immediatamente l'autenticazione IAM per un'istanza database esistente.

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --apply-immediately \
    --enable-iam-database-authentication
```

Se ripristini un'istanza database, usa uno dei comandi AWS CLI seguenti:

- `restore-db-instance-to-point-in-time`
- `restore-db-instance-from-db-snapshot`

L'impostazione di autenticazione del database IAM corrisponde a quella della snapshot origine. Per modificare questa impostazione, imposta l'opzione `--enable-iam-database-authentication` or `--no-enable-iam-database-authentication`, come appropriato.

API RDS

Per creare una nuova istanza database con autenticazione IAM tramite l'API, utilizzare l'operazione API `CreateDBInstance`. Imposta il parametro `EnableIAMDatabaseAuthentication` su `true`.

Per aggiornare un'istanza database esistente in modo da abilitare l'autenticazione IAM, utilizzare l'operazione API `ModifyDBInstance`. Imposta il parametro `EnableIAMDatabaseAuthentication` su `true` per abilitare l'autenticazione IAM o su `false` per disabilitarla.

> **Note**
>
> Verifica che l'istanza database sia compatibile con l'autenticazione IAM. Controllare i requisiti di compatibilità in Disponibilità di regioni e versioni.

Se ripristini un'istanza database, usa una delle operazioni API seguenti:

- `RestoreDBInstanceFromDBSnapshot`
- `RestoreDBInstanceToPointInTime`

L'impostazione di autenticazione del database IAM corrisponde a quella della snapshot origine. Per modificare questa impostazione, imposta il parametro `EnableIAMDatabaseAuthentication` su `true` per abilitare l'autenticazione IAM o su `false` per disabilitarla.

Creazione e utilizzo di una policy IAM per l'accesso al database IAM

Per permettere a un utente o un ruolo di connettersi all'istanza database, devi creare una policy IAM. Dopo questa operazione, collega la policy a un set di autorizzazioni o un ruolo.

> **Note**
>
> Per ulteriori informazioni sulle policy IAM, consulta Gestione accessi e identità per Amazon RDS.

La policy nell'esempio seguente permette a un utente di connettersi a un'istanza database tramite l'autenticazione database IAM.

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Action": [
             "rds-db:connect"
         ],
         "Resource": [
             "arn:aws:rds-db:us-east-2:1234567890:dbuser:db-ABCDEFGHIJKL01234/db_user"
         ]
      }
   ]
}
```

> **Important**
>
> Un utente con le autorizzazioni di amministratore può accedere alle istanze database senza una policy IAM esplicita. Se vuoi limitare l'accesso come amministratore a di istanze, puoi

creare un ruolo IAM con le autorizzazioni appropriate, con privilegi minori e assegnarlo all'amministratore.

> **Note**
>
> Non confondere il prefisso `rds-db:` con altri prefissi di operazione API RDS che iniziano con `rds:`. Utilizzi il prefisso `rds-db:` e l'operazione `rds-db:connect` solo per l'autenticazione database IAM. Non sono validi in altri contesti.

La policy di esempio include una singola istruzione con i seguenti elementi:

- `Effect` – Specifica `Allow` per concedere l'accesso all'istanza database. Se non si concede esplicitamente l'accesso, questo viene rifiutato per impostazione predefinita.
- `Action` – Specifica `rds-db:connect` per consentire le connessioni al dell'istanza database.
- `Resource` – Specifica un Amazon Resource Name (ARN) che descrive un account database in un'istanza database. Di seguito è riportato il formato ARN.

```
arn:aws:rds-db:region:account-id:dbuser:DbiResourceId/db-user-name
```

  In questo formato, sostituisci quanto segue:

  - *`region`* è la regione AWS per l'istanza. Nella policy di esempio, la regione AWS è `us-east-2`.
  - *`account-id`* è il numero di account AWS per l'istanza. Nella policy di esempio, il numero di account è `1234567890`. L'utente deve essere nello stesso account dell'account dell'istanza database.

    Per eseguire l'accesso multi-account, crea un ruolo IAM con la policy mostrata in precedenza nell'account per l'istanza database e consenti all'altro account di assumere il ruolo.
  - *`DbiResourceId`* è l'identificatore del dell'istanza database. L'identificatore è univoco per una regione AWS e non cambia mai. Nella policy di esempio, l'identificatore è `db-ABCDEFGHIJKL01234`.

    Per trovare l'ID risorsa di un'istanza database nella AWS Management Console per Amazon RDS, scegli l'istanza database per visualizzarne i dettagli. Quindi seleziona la scheda

Configuration (Configurazione). Resource ID (ID risorsa) è visualizzato nella sezione Configuration (Configurazione).

In alternativa, puoi utilizzare il comando AWS CLI per elencare gli identificatori e gli ID della risorsa per tutte le istanze nella regione AWS corrente, come riportato di seguito.

```
aws rds describe-db-instances --query "DBInstances[*].
[DBInstanceIdentifier,DbiResourceId]"
```

Se utilizzi Amazon Aurora, specifica `DbClusterResourceId` anziché `DbiResourceId`. Per ulteriori informazioni, consulta Creazione e utilizzo di una policy IAM per l'accesso al database IAM nella Guida per l'utente di Amazon Aurora.

> **Note**
>
> Se si sta effettuando la connessione a un database tramite proxy RDS, specificare l'ID risorsa proxy, ad esempio `prx-ABCDEFGHIJKL01234`. Per informazioni sull'utilizzo dell'autenticazione del database IAM con proxy RDS, vedere Connessione a un proxy mediante autenticazione IAM.

- *`db-user-name`* è il nome dell'account database da associare all'autenticazione IAM. Nella policy di esempio, l'account database è `db_user`.

Puoi creare altri nomi ARN per supportare i vari modelli di accesso. La policy seguente permette l'accesso a due diversi account database in un dell'istanza database.

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Action": [
             "rds-db:connect"
         ],
         "Resource": [
             "arn:aws:rds-db:us-east-2:123456789012:dbuser:db-ABCDEFGHIJKL01234/
jane_doe",
```

```
            "arn:aws:rds-db:us-east-2:123456789012:dbuser:db-ABCDEFGHIJKL01234/
mary_roe"
         ]
      }
   ]
}
```

La policy seguente usa il carattere "*" per trovare le corrispondenze di tutte le istanze DB e di tutti gli account database per un account AWS e una regione AWS specifici.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "rds-db:connect"
            ],
            "Resource": [
                "arn:aws:rds-db:us-east-2:1234567890:dbuser:*/*"
            ]
        }
    ]
}
```

La policy seguente ricerca la corrispondenza di tutte le istanze per un account AWS e una regione AWS specifici. Tuttavia, la policy concede l'accesso solo a istanze database che hanno un account database `jane_doe`.

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Effect": "Allow",
         "Action": [
             "rds-db:connect"
         ],
         "Resource": [
```

```
            "arn:aws:rds-db:us-east-2:123456789012:dbuser:*/jane_doe"
         ]
      }
   ]
}
```

L'utente o il ruolo ha accesso solo a quei database ai quali l'utente ha accesso. Supponiamo, ad esempio, che il dell'istanza database abbia un database denominato dev e un altro database denominato test. Se l'utente del database `jane_doe` ha accesso solo a dev, anche qualsiasi utente o ruolo che accede all'istanza database con l'utente `jane_doe` ha accesso solo a dev. Questa restrizione dell'accesso è anche valida per altri oggetti database, come tabelle, visualizzazioni e così via.

Un amministratore deve creare policy IAM che concedono alle entità l'autorizzazione per eseguire operazioni API specifiche sulle risorse specificate di cui hanno bisogno. L'amministratore deve quindi collegare queste policy ai set di autorizzazioni o ai ruoli che richiedono tali autorizzazioni. Per esempi di policy, consulta Esempi di policy di Amazon RDS basate su identità.

Collegamento di una policy IAM a un set di autorizzazioni o un ruolo

Dopo aver creato una policy IAM per consentire l'autenticazione del database, devi collegare la policy a un set di autorizzazioni o un ruolo. Per un tutorial su questo argomento, consulta Creare e collegare la tua prima policy gestita dal cliente nella Guida per l'utente IAM.

Durante il tutorial, puoi utilizzare uno degli esempi di policy indicati in questa sezione come punto di partenza e personalizzarli in base alle tue esigenze. Al termine del tutorial, avrai un set di autorizzazioni con una policy collegata che può utilizzare l'operazione `rds-db:connect`.

> **Note**
>
> Puoi mappare più set di autorizzazioni o ruoli allo stesso account utente del database. Ad esempio, supponiamo che la policy IAM abbia specificato la seguente risorsa ARN.
>
> ```
> arn:aws:rds-db:us-east-2:123456789012:dbuser:db-12ABC34DEFG5HIJ6KLMNOP78QR/
> jane_doe
> ```

Se colleghi la policy agli utenti Jane, Bob e Diego, ciascuno di quegli utenti potrà connettersi all'istanza database specificato utilizzando l'account di database `jane_doe`.

Creazione di un account database tramite l'autenticazione IAM

Con l'autenticazione database IAM, non devi assegnare password di database agli account utente che crei. Se rimuovi un utente mappato a un account database, devi anche rimuovere l'account database con l'istruzione `DROP USER`.

> **Note**
>
> Il nome utente utilizzato per l'autenticazione IAM deve avere lo stesso formato maiuscolo/minuscolo del nome utente nel database.

Argomenti

- Utilizzo dell'autenticazione IAM con MariaDB e MySQL
- Utilizzo dell'autenticazione IAM con PostgreSQL

Utilizzo dell'autenticazione IAM con MariaDB e MySQL

Con MariaDB e MySQL, l'autenticazione viene gestita da `AWSAuthenticationPlugin`, un plug-in fornito da AWS che funziona perfettamente con IAM per autenticare gli utenti. Connettiti all'istanza database come utente master o altro utente che può creare utenti e concedere privilegi. Dopo la connessione, immetti l'istruzione `CREATE USER`, come mostrato nell'esempio seguente.

```
CREATE USER jane_doe IDENTIFIED WITH AWSAuthenticationPlugin AS 'RDS';
```

La clausola `IDENTIFIED WITH` permette a MariaDB e MySQL di utilizzare `AWSAuthenticationPlugin` per autenticare l'account database (`jane_doe`). La clausola `AS 'RDS'` fa riferimento al metodo di autenticazione. Assicurarsi che il nome utente del database specificato sia lo stesso di una risorsa nella policy IAM per l'accesso al database IAM. Per ulteriori informazioni, consulta Creazione e utilizzo di una policy IAM per l'accesso al database IAM.

> **Note**
>
> Se viene visualizzato il messaggio seguente, significa che il plug-in fornito da AWS non è disponibile per l'istanza corrente.
> `ERROR 1524 (HY000): Plugin 'AWSAuthenticationPlugin' is not loaded`
> Per correggere questo errore, assicurati di usare una configurazione supportata e di aver abilitato l'autenticazione database IAM nell'istanza database. Per ulteriori informazioni, consulta Disponibilità di regioni e versioni e Abilitazione e disabilitazione dell'autenticazione database IAM.

Dopo aver creato un account utilizzando `AWSAuthenticationPlugin`, lo gestisci nello stesso modo di altri account database. Ad esempio, puoi modificare i privilegi di account con le istruzioni `GRANT` e `REVOKE` o modificare vari attributi di account con l'istruzione `ALTER USER`.

Il traffico di rete del database viene crittografato utilizzando SSL/TLS quando si utilizza IAM. Per consentire le connessioni SSL, modifica l'account utente con il comando seguente.

```
ALTER USER 'jane_doe'@'%' REQUIRE SSL;
```

Utilizzo dell'autenticazione IAM con PostgreSQL

Per utilizzare l'autenticazione IAM con PostgreSQL, connettiti all'istanza database come utente master o altro utente che può creare utenti e concedere privilegi. Dopo la connessione, crea gli utenti di database e autorizza il ruolo `rds_iam`, come mostrato nell'esempio seguente.

```
CREATE USER db_userx;
GRANT rds_iam TO db_userx;
```

Assicurarsi che il nome utente del database specificato sia lo stesso di una risorsa nella policy IAM per l'accesso al database IAM. Per ulteriori informazioni, consulta Creazione e utilizzo di una policy IAM per l'accesso al database IAM.

Connessione all'istanza tramite l'autenticazione IAM

Con l'autenticazione database IAM devi usare un token di autenticazione per la connessione all'istanza. Un token di autenticazione è una stringa unica di caratteri che utilizzi invece di una

password. Trascorsi 15 minuti dalla sua creazione, un token di autenticazione scade. Se cerchi di eseguire la connessione utilizzando un token scaduto la richiesta di connessione viene negata.

Ogni token di autenticazione deve essere accompagnato da una firma valida, utilizzando AWS Signature Version 4. (Per ulteriori informazioni, vedere Processo di firma Signature Version 4 in Riferimenti generali di AWS.) AWS CLI E un AWS SDK, come AWS SDK for Java o AWS SDK for Python (Boto3), possono firmare automaticamente ogni token creato.

Puoi utilizzare un token di autenticazione quando ti connetti ad Amazon RDS Aurora da AWS un altro servizio, ad esempio. AWS Lambda Utilizzando un token, eviti di inserire una password nel codice. In alternativa, puoi utilizzare un AWS SDK per creare e firmare programmaticamente un token di autenticazione.

Quando hai un token di autenticazione IAM firmato, puoi connetterti a un'istanza database Amazon RDS. Di seguito, puoi scoprire come eseguire questa operazione utilizzando uno strumento da riga di comando o un AWS SDK, come o. AWS SDK for Java AWS SDK for Python (Boto3)

Per ulteriori informazioni, consulta il seguente post sul blog:

- Uso dell'autenticazione IAM per la connessione con SQL Workbench/J a Aurora MySQL o Amazon RDS for MySQL.
- Utilizzo dell'autenticazione IAM per connettersi con pgAdmin Amazon Aurora PostgreSQL o Amazon RDS for PostgreSQL

Prerequisiti

Di seguito sono riportati i prerequisiti per la connessione al di istanzaDB utilizzando l'autenticazione IAM:

- Abilitazione e disabilitazione dell'autenticazione database IAM
- Creazione e utilizzo di una policy IAM per l'accesso al database IAM
- Creazione di un account database tramite l'autenticazione IAM

Argomenti

- Connessione al di istanze DB utilizzando l'autenticazione IAM con i AWS driver
- Connessione al di istanze DB utilizzando l'autenticazione IAM dalla riga di comando: AWS CLI e il client mysql

- Connessione all'istanza database tramite Autenticazione IAM dalla riga di comando: AWS CLI e client psql
- Connessione al cluster tramite Autenticazione IAM e AWS SDK for .NET
- Connessione al cluster tramite Autenticazione IAM e AWS SDK for Go
- Connessione al cluster tramite Autenticazione IAM e AWS SDK for Java
- Connessione al cluster tramite Autenticazione IAM e AWS SDK for Python (Boto3)

Connessione al di istanze DB utilizzando l'autenticazione IAM con i AWS driver

La AWS suite di driver è stata progettata per fornire supporto per tempi di switchover e failover più rapidi e l'autenticazione con AWS Secrets Manager, AWS Identity and Access Management (IAM) e Federated Identity. I AWS driver si basano sul monitoraggio dello stato dell'istanza DB del DB e sulla conoscenza della topologia dell'istanza del per determinare il nuovo writer. Questo approccio riduce i tempi di switchover e failover a secondi a una cifra, rispetto alle decine di secondi dei driver open source.

Per ulteriori informazioni sui AWS driver, consulta il driver di lingua corrispondente per la tua istanza DB RDS per MariaDB, RDS per MySQL o RDS per PostgreSQL DB.

> **Note**
>
> Le uniche funzionalità supportate per RDS per MariadB sono l'autenticazione AWS Secrets Manager con AWS Identity and Access Management , (IAM) e l'identità federata.

Connessione al di istanze DB utilizzando l'autenticazione IAM dalla riga di comando: AWS CLI e il client mysql

Puoi connetterti dalla riga di comando a un cluster DB di istanze Amazon RDS con AWS CLI lo strumento da riga di comando `mysql` and come descritto di seguito.

Prerequisiti

Di seguito sono riportati i prerequisiti per la connessione al di istanzaDB utilizzando l'autenticazione IAM:

- Abilitazione e disabilitazione dell'autenticazione database IAM

- Creazione e utilizzo di una policy IAM per l'accesso al database IAM
- Creazione di un account database tramite l'autenticazione IAM

> **Note**
>
> Per informazioni sulla connessione al database tramite SQL Workbench/J con autenticazione IAM, consulta il post del blog Utilizzo dell'autenticazione IAM per connettersi con SQL Workbench/J a Aurora MySQL o Amazon RDS for MySQL.

Argomenti

- Generazione di un token di autenticazione IAM
- Connessione a un'istanza database

Generazione di un token di autenticazione IAM

L'esempio seguente mostra come ottenere un token di autenticazione firmato utilizzando AWS CLI.

```
aws rds generate-db-auth-token \
   --hostname rdsmysql.123456789012.us-west-2.rds.amazonaws.com \
   --port 3306 \
   --region us-west-2 \
   --username jane_doe
```

Nell'esempio, i parametri sono come segue:

- `--hostname` – Nome host dell'istanza database cui vuoi accedere.
- `--port` – Numero di porta usato per la connessione al cluster
- `--region`— La AWS regione in cui è in esecuzione il di istanze DB
- `--username` – L'account database cui vuoi accedere.

I primi caratteri del token hanno il seguente aspetto.

```
rdsmysql.123456789012.us-west-2.rds.amazonaws.com:3306/?
Action=connect&DBUser=jane_doe&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Expires=900...
```

> **Note**
>
> Non è possibile utilizzare un record DNS Route 53 personalizzato anziché l'endpoint dell'istanza database per generare il token di autenticazione.

Connessione a un'istanza database

Il formato generale per la connessione è visualizzato di seguito.

```
mysql --host=hostName --port=portNumber --ssl-ca=full_path_to_ssl_certificate --enable-cleartext-plugin --user=userName --password=authToken
```

I parametri sono i seguenti:

- `--host` – Nome host dell'istanza database cui vuoi accedere.
- `--port` – Numero di porta usato per la connessione al cluster
- `--ssl-ca` – Il percorso completo del file del certificato SSL che contiene la chiave pubblica

  Per ulteriori informazioni sul supporto SSL/TLS per MariaDB, consulta Utilizzo di SSL/TLS con un'istanza database MariaDB.

  Per ulteriori informazioni sul supporto SSL/TLS per MySQL, consulta Utilizzo di SSL/TLS con un'istanza database MySQL.

  Per scaricare un certificato SSL consulta
- `--enable-cleartext-plugin` – Valore che specifica che per la connessione deve essere usato `AWSAuthenticationPlugin`.

  Se si utilizza un client MariaDB, l'opzione `--enable-cleartext-plugin` non è richiesta.
- `--user` – L'account database cui vuoi accedere.
- `--password` – Token di autenticazione IAM firmato.

Il token di autenticazione consiste in diverse centinaia di caratteri. Può essere macchinoso nella riga di comando. Una soluzione è di salvare il token in una variabile di ambiente e utilizzare quella variabile durante la connessione. L'esempio seguente mostra un modo per eseguire questa soluzione. Nell'esempio,*/sample_dir/* è il percorso completo del file del certificato SSL che contiene la chiave pubblica.

```
RDSHOST="mysqldb.123456789012.us-east-1.rds.amazonaws.com"
TOKEN="$(aws rds generate-db-auth-token --hostname $RDSHOST --port 3306 --region us-
west-2 --username jane_doe )"

mysql --host=$RDSHOST --port=3306 --ssl-ca=/sample_dir/global-bundle.pem --enable-
cleartext-plugin --user=jane_doe --password=$TOKEN
```

Quando si esegue la connessione utilizzando `AWSAuthenticationPlugin`, la connessione viene protetta utilizzando SSL. Per verificare ciò, digita quanto segue al prompt del comando `mysql>`.

```
show status like 'Ssl%';
```

Le righe seguenti nell'output mostrano più dettagli.

```
+---------------+-------------+
| Variable_name | Value

                                                                    |

+---------------+-------------+
| ...           | ...
| Ssl_cipher    | AES256-SHA

                                                                    |

| ...           | ...
| Ssl_version   | TLSv1.1

                                                                    |

| ...           | ...
+-----------------------------+
```

Se desideri connetterti a un'istanza database tramite un proxy, consulta Connessione a un proxy mediante autenticazione IAM.

Connessione all'istanza database tramite Autenticazione IAM dalla riga di comando: AWS CLI e client psql

Puoi eseguire la connessione dalla riga di comando a una istanza database Amazon RDS for PostgreSQL con AWS CLI e lo strumento a riga di comando psql come descritto di seguito.

Prerequisiti

Di seguito sono riportati i prerequisiti per la connessione al di istanzaDB utilizzando l'autenticazione IAM:

- Abilitazione e disabilitazione dell'autenticazione database IAM
- Creazione e utilizzo di una policy IAM per l'accesso al database IAM
- Creazione di un account database tramite l'autenticazione IAM

> **Note**
>
> Per informazioni sulla connessione al database tramite pgAdmin con autenticazione IAM, consulta il post sul blog Utilizzo dell'autenticazione IAM per connettersi con PGadmin Amazon Aurora PostgreSQL o Amazon RDS for PostgreSQL.

Argomenti

- Generazione di un token di autenticazione IAM
- Connessione a un'istanza Amazon RDS PostgreSQL

Generazione di un token di autenticazione IAM

Il token di autenticazione è costituito da diverse centinaia di caratteri, quindi può essere complesso nella riga di comando. Una soluzione è di salvare il token in una variabile di ambiente e utilizzare quella variabile durante la connessione. Il seguente esempio mostra come usare l'AWS CLI per ottenere un token di autenticazione firmato utilizzando il comando `generate-db-auth-token` e archiviarlo in una variabile di ambiente `PGPASSWORD`.

```
export RDSHOST="rdspostgres.123456789012.us-west-2.rds.amazonaws.com"
export PGPASSWORD="$(aws rds generate-db-auth-token --hostname $RDSHOST --port 5432 --
region us-west-2 --username jane_doe )"
```

Nell'esempio, i parametri per il comando `generate-db-auth-token` sono i seguenti:

- `--hostname` – Nome host dell'istanza database cui desideri accedere.
- `--port` – Numero di porta usato per la connessione al cluster
- `--region`: la regione AWS in cui è in esecuzione l'istanza database
- `--username` – L'account database cui vuoi accedere.

I primi caratteri del token generato hanno il seguente aspetto.

```
rdspostgres.123456789012.us-west-2.rds.amazonaws.com:5432/?
Action=connect&DBUser=jane_doe&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Expires=900...
```

> **Note**
>
> Non è possibile utilizzare un record DNS Route 53 personalizzato anziché l'endpoint dell'istanza database per generare il token di autenticazione.

Connessione a un'istanza Amazon RDS PostgreSQL

Di seguito è mostrato il formato generale per l'utilizzo di psql per la connessione.

```
psql "host=hostName port=portNumber sslmode=verify-full
 sslrootcert=full_path_to_ssl_certificate dbname=DBName user=userName
 password=authToken"
```

I parametri sono i seguenti:

- `host` – Nome host dell'istanza database cui desideri accedere.
- `port` – Numero di porta usato per la connessione al cluster
- `sslmode` – Modalità SSL da utilizzare.

  Quando si utilizza `sslmode=verify-full`, la connessione SSL verifica l'endpoint dell'istanza database rispetto a quello nel certificato SSL.
- `sslrootcert` – Il percorso completo del file del certificato SSL che contiene la chiave pubblica

  Per ulteriori informazioni, consulta Utilizzo del protocollo SSL con un'istanza database PostgreSQL.

  Per scaricare un certificato SSL consulta
- `dbname` – Database a cui accedere.
- `user` – L'account database cui vuoi accedere.
- `password` – Token di autenticazione IAM firmato.

> **Note**
>
> Non è possibile utilizzare un record DNS Route 53 personalizzato anziché l'endpoint dell'istanza database per generare il token di autenticazione.

L'esempio seguente mostra l'utilizzo di psql per la connessione. Nell'esempio, psql utilizza la variabile d'ambiente RDSHOST per l'host e la variabile d'ambiente PGPASSWORD per il token generato. Inoltre, */sample_dir/* è il percorso completo al file del certificato SSL che contiene la chiave pubblica.

```
export RDSHOST="rdspostgres.123456789012.us-west-2.rds.amazonaws.com"
export PGPASSWORD="$(aws rds generate-db-auth-token --hostname $RDSHOST --port 5432 --
region us-west-2 --username jane_doe )"

psql "host=$RDSHOST port=5432 sslmode=verify-full sslrootcert=/sample_dir/global-
bundle.pem dbname=DBName user=jane_doe password=$PGPASSWORD"
```

Se desideri connetterti a un'istanza database tramite un proxy, consulta Connessione a un proxy mediante autenticazione IAM.

Connessione al cluster tramite Autenticazione IAM e AWS SDK for .NET

Puoi connetterti a un'istanza database RDS for MariaDB, MySQL o PostgreSQL con la AWS SDK for .NET come descritto di seguito.

Prerequisiti

Di seguito sono riportati i prerequisiti per la connessione al di istanzaDB utilizzando l'autenticazione IAM:

- Abilitazione e disabilitazione dell'autenticazione database IAM
- Creazione e utilizzo di una policy IAM per l'accesso al database IAM
- Creazione di un account database tramite l'autenticazione IAM

Esempi

Il seguente esempio di codice mostra come generare un token di autenticazione e utilizzarlo per eseguire la connessione a un'istanza del database.

Per eseguire questo codice di esempio è necessario AWS SDK for .NET, disponibile sul sito AWS. I pacchetti `AWSSDK.CORE` e `AWSSDK.RDS` sono obbligatori. Per connetterti a un'istanza database, utilizza il connettore di database .NET per il motore di database, ad esempio MySQLConnector per MariaDB o MySQL o Npgsql per PostgreSQL.

Questo codice si connette a un'istanza database MariaDB o MySQL. Modifica i valori delle variabili seguenti in base alle esigenze.

- `server`: l'endpoint dell'istanza database cui vuoi accedere
- `user` – L'account database cui vuoi accedere.
- `database` – Database a cui accedere.
- `port` – Numero di porta usato per la connessione al cluster
- `SslMode` – Modalità SSL da utilizzare.

  Quando si utilizza `SslMode=Required`, la connessione SSL verifica l'endpoint dell'istanza database rispetto a quello nel certificato SSL.
- `SslCa` - Percorso completo del certificato SSL per Amazon RDS

  Per scaricare un certificato, consultare .

> **Note**
>
> Non è possibile utilizzare un record DNS Route 53 personalizzato anziché l'endpoint dell'istanza database per generare il token di autenticazione.

```
using System;
using System.Data;
using MySql.Data;
using MySql.Data.MySqlClient;
using Amazon;

namespace ubuntu
{
  class Program
  {
    static void Main(string[] args)
    {
```

```
      var pwd =
 Amazon.RDS.Util.RDSAuthTokenGenerator.GenerateAuthToken(RegionEndpoint.USEast1,
 "mysqldb.123456789012.us-east-1.rds.amazonaws.com", 3306, "jane_doe");
      // for debug only Console.Write("{0}\n", pwd);  //this verifies the token is
 generated

      MySqlConnection conn = new MySqlConnection($"server=mysqldb.123456789012.us-
east-1.rds.amazonaws.com;user=jane_doe;database=mydB;port=3306;password={pwd};SslMode=Required;
      conn.Open();

      // Define a query
      MySqlCommand sampleCommand = new MySqlCommand("SHOW DATABASES;", conn);

      // Execute a query
      MySqlDataReader mysqlDataRdr = sampleCommand.ExecuteReader();

      // Read all rows and output the first column in each row
      while (mysqlDataRdr.Read())
        Console.WriteLine(mysqlDataRdr[0]);

      mysqlDataRdr.Close();
      // Close connection
      conn.Close();
    }
  }
}
```

Questo codice si connette a un'istanza database PostgreSQL.

Modifica i valori delle variabili seguenti in base alle esigenze.

- `Server`: l'endpoint dell'istanza database cui vuoi accedere
- `User ID` – L'account database cui vuoi accedere.
- `Database` – Database a cui accedere.
- `Port` – Numero di porta usato per la connessione al cluster
- `SSL Mode` – Modalità SSL da utilizzare.

  Quando si utilizza `SSL Mode=Required`, la connessione SSL verifica l'endpoint dell'istanza database rispetto a quello nel certificato SSL.
- `Root Certificate` - Percorso completo del certificato SSL per Amazon RDS

Per scaricare un certificato, consultare .

> **Note**
>
> Non è possibile utilizzare un record DNS Route 53 personalizzato anziché l'endpoint dell'istanza database per generare il token di autenticazione.

```
using System;
using Npgsql;
using Amazon.RDS.Util;

namespace ConsoleApp1
{
    class Program
    {
        static void Main(string[] args)
        {
            var pwd =
 RDSAuthTokenGenerator.GenerateAuthToken("postgresmydb.123456789012.us-
east-1.rds.amazonaws.com", 5432, "jane_doe");
// for debug only Console.Write("{0}\n", pwd);  //this verifies the token is generated

            NpgsqlConnection conn = new
 NpgsqlConnection($"Server=postgresmydb.123456789012.us-east-1.rds.amazonaws.com;User
 Id=jane_doe;Password={pwd};Database=mydb;SSL Mode=Require;Root
 Certificate=full_path_to_ssl_certificate");
            conn.Open();

            // Define a query
                   NpgsqlCommand cmd = new NpgsqlCommand("select count(*) FROM
 pg_user", conn);

            // Execute a query
            NpgsqlDataReader dr = cmd.ExecuteReader();

            // Read all rows and output the first column in each row
            while (dr.Read())
                Console.Write("{0}\n", dr[0]);

            // Close connection
```

```
            conn.Close();
        }
    }
}
```

Se desideri connetterti a un'istanza database tramite un proxy, consulta Connessione a un proxy mediante autenticazione IAM.

Connessione al cluster tramite Autenticazione IAM e AWS SDK for Go

Puoi connetterti a un'istanza database RDS for MariaDB, MySQL o PostgreSQL con la AWS SDK for Go come descritto di seguito.

Prerequisiti

Di seguito sono riportati i prerequisiti per la connessione al di istanzaDB utilizzando l'autenticazione IAM:

- Abilitazione e disabilitazione dell'autenticazione database IAM
- Creazione e utilizzo di una policy IAM per l'accesso al database IAM
- Creazione di un account database tramite l'autenticazione IAM

Esempi

Per eseguire questo codice di esempio è necessario AWS SDK for Go, disponibile sul sito AWS.

Modifica i valori delle variabili seguenti in base alle esigenze.

- `dbName` – Database a cui accedere.
- `dbUser` – L'account database cui vuoi accedere.
- `dbHost`: l'endpoint dell'istanza database cui vuoi accedere

> **Note**
>
> Non è possibile utilizzare un record DNS Route 53 personalizzato anziché l'endpoint dell'istanza database per generare il token di autenticazione.

- `dbPort` – Numero di porta usato per la connessione al cluster
- `region`: la regione AWS in cui è in esecuzione l'istanza database

Inoltre, assicurarsi che le librerie importate nel codice di esempio esistano nel sistema.

> **Important**
>
> Negli esempi riportati in questa sezione viene utilizzato il codice seguente per fornire credenziali che accedono a un database da un ambiente locale:
> `creds := credentials.NewEnvCredentials()`
> Se si accede a un database da un servizio AWS, ad esempio Amazon EC2 o Amazon ECS, è possibile sostituire il codice con il seguente:
> `sess := session.Must(session.NewSession())`
> `creds := sess.Config.Credentials`
> Se si apporta questa modifica, assicurarsi di aggiungere la seguente importazione:
> `"github.com/aws/aws-sdk-go/aws/session"`

Argomenti

- Connessione tramite Autenticazione IAM e AWS SDK for Go V2
- Connessione mediante IAM e AWS SDK for Go V1.

Connessione tramite Autenticazione IAM e AWS SDK for Go V2

È possibile connettersi a un di istanze database utilizzando l'autenticazione IAM e AWS SDK for Go V2.

Il seguente esempio di codice mostra come generare un token di autenticazione e utilizzarlo per eseguire la connessione a un'istanza del database.

Questo codice si connette a un'istanza database MariaDB o MySQL.

```
package main

import (
     "context"
     "database/sql"
     "fmt"

     "github.com/aws/aws-sdk-go-v2/config"
     "github.com/aws/aws-sdk-go-v2/feature/rds/auth"
     _ "github.com/go-sql-driver/mysql"
```

```
)

func main() {

     var dbName string = "DatabaseName"
     var dbUser string = "DatabaseUser"
     var dbHost string = "mysqldb.123456789012.us-east-1.rds.amazonaws.com"
     var dbPort int = 3306
     var dbEndpoint string = fmt.Sprintf("%s:%d", dbHost, dbPort)
     var region string = "us-east-1"

    cfg, err := config.LoadDefaultConfig(context.TODO())
    if err != nil {
     panic("configuration error: " + err.Error())
    }

    authenticationToken, err := auth.BuildAuthToken(
     context.TODO(), dbEndpoint, region, dbUser, cfg.Credentials)
    if err != nil {
     panic("failed to create authentication token: " + err.Error())
    }

    dsn := fmt.Sprintf("%s:%s@tcp(%s)/%s?tls=true&allowCleartextPasswords=true",
        dbUser, authenticationToken, dbEndpoint, dbName,
    )

    db, err := sql.Open("mysql", dsn)
    if err != nil {
        panic(err)
    }

    err = db.Ping()
    if err != nil {
        panic(err)
    }
}
```

Questo codice si connette a un'istanza database PostgreSQL.

```
package main

import (
     "context"
```

```
     "database/sql"
     "fmt"

     "github.com/aws/aws-sdk-go-v2/config"
     "github.com/aws/aws-sdk-go-v2/feature/rds/auth"
     _ "github.com/lib/pq"
)

func main() {

     var dbName string = "DatabaseName"
     var dbUser string = "DatabaseUser"
     var dbHost string = "postgresmydb.123456789012.us-east-1.rds.amazonaws.com"
     var dbPort int = 5432
     var dbEndpoint string = fmt.Sprintf("%s:%d", dbHost, dbPort)
     var region string = "us-east-1"

    cfg, err := config.LoadDefaultConfig(context.TODO())
    if err != nil {
     panic("configuration error: " + err.Error())
    }

    authenticationToken, err := auth.BuildAuthToken(
     context.TODO(), dbEndpoint, region, dbUser, cfg.Credentials)
    if err != nil {
     panic("failed to create authentication token: " + err.Error())
    }

    dsn := fmt.Sprintf("host=%s port=%d user=%s password=%s dbname=%s",
        dbHost, dbPort, dbUser, authenticationToken, dbName,
    )

    db, err := sql.Open("postgres", dsn)
    if err != nil {
        panic(err)
    }

    err = db.Ping()
    if err != nil {
        panic(err)
    }
}
```

Se desideri connetterti a un'istanza database tramite un proxy, consulta Connessione a un proxy mediante autenticazione IAM.

Connessione mediante IAM e AWS SDK for Go V1.

È possibile connettersi a un di istanze database utilizzando l'autenticazione IAM e AWS SDK for Go V1

Il seguente esempio di codice mostra come generare un token di autenticazione e utilizzarlo per eseguire la connessione a un'istanza del database.

Questo codice si connette a un'istanza database MariaDB o MySQL.

```
package main

import (
    "database/sql"
    "fmt"
    "log"

    "github.com/aws/aws-sdk-go/aws/credentials"
    "github.com/aws/aws-sdk-go/service/rds/rdsutils"
    _ "github.com/go-sql-driver/mysql"
)

func main() {
    dbName := "app"
    dbUser := "jane_doe"
    dbHost := "mysqldb.123456789012.us-east-1.rds.amazonaws.com"
    dbPort := 3306
    dbEndpoint := fmt.Sprintf("%s:%d", dbHost, dbPort)
    region := "us-east-1"

    creds := credentials.NewEnvCredentials()
    authToken, err := rdsutils.BuildAuthToken(dbEndpoint, region, dbUser, creds)
    if err != nil {
        panic(err)
    }

    dsn := fmt.Sprintf("%s:%s@tcp(%s)/%s?tls=true&allowCleartextPasswords=true",
        dbUser, authToken, dbEndpoint, dbName,
    )
```

```
    db, err := sql.Open("mysql", dsn)
    if err != nil {
        panic(err)
    }

    err = db.Ping()
    if err != nil {
        panic(err)
    }
}
```

Questo codice si connette a un'istanza database PostgreSQL.

```
package main

import (
 "database/sql"
 "fmt"

 "github.com/aws/aws-sdk-go/aws/credentials"
 "github.com/aws/aws-sdk-go/service/rds/rdsutils"
 _ "github.com/lib/pq"
)

func main() {
    dbName := "app"
    dbUser := "jane_doe"
    dbHost := "postgresmydb.123456789012.us-east-1.rds.amazonaws.com"
    dbPort := 5432
    dbEndpoint := fmt.Sprintf("%s:%d", dbHost, dbPort)
    region := "us-east-1"

    creds := credentials.NewEnvCredentials()
    authToken, err := rdsutils.BuildAuthToken(dbEndpoint, region, dbUser, creds)
    if err != nil {
        panic(err)
    }

    dsn := fmt.Sprintf("host=%s port=%d user=%s password=%s dbname=%s",
        dbHost, dbPort, dbUser, authToken, dbName,
    )

    db, err := sql.Open("postgres", dsn)
```

```
    if err != nil {
        panic(err)
    }

    err = db.Ping()
    if err != nil {
        panic(err)
    }
}
```

Se desideri connetterti a un'istanza database tramite un proxy, consulta Connessione a un proxy mediante autenticazione IAM.

Connessione al cluster tramite Autenticazione IAM e AWS SDK for Java

Puoi connetterti a un'istanza database RDS for MariaDB, MySQL o PostgreSQL con la AWS SDK for Java come descritto di seguito.

Prerequisiti

Di seguito sono riportati i prerequisiti per la connessione al di istanzaDB utilizzando l'autenticazione IAM:

- Abilitazione e disabilitazione dell'autenticazione database IAM
- Creazione e utilizzo di una policy IAM per l'accesso al database IAM
- Creazione di un account database tramite l'autenticazione IAM
- Configura l'SDK AWS per Java

Argomenti

- Generazione di un token di autenticazione IAM
- Creazione manuale di un token di autenticazione IAM
- Connessione a un'istanza database

Generazione di un token di autenticazione IAM

Se scrivi programmi usando l'AWS SDK for Java, puoi ottenere un token di autenticazione firmato tramite la classe `RdsIamAuthTokenGenerator`. L'utilizzo di questa classe richiede che vengano fornite le credenziali AWS. A questo scopo, devi creare un'istanza della classe `DefaultAWSCredentialsProviderChain`. `DefaultAWSCredentialsProviderChain`

utilizza la prima chiave di accesso e la chiave segreta AWS che trova nella catena di provider delle credenziali predefinite. Per ulteriori informazioni sulle chiavi di accesso AWS, consulta Gestione delle chiavi di accesso per gli utenti IAM.

> **Note**
>
> Non è possibile utilizzare un record DNS Route 53 personalizzato anziché l'endpoint dell'istanza database per generare il token di autenticazione.

Dopo aver creato un'istanza di `RdsIamAuthTokenGenerator`, puoi chiamare il metodo `getAuthToken` per ottenere un token firmato. Specifica la regione AWS, il nome host, il numero di porta e il nome utente. Il seguente esempio di codice dimostra come eseguire questa operazione.

```
package com.amazonaws.codesamples;

import com.amazonaws.auth.DefaultAWSCredentialsProviderChain;
import com.amazonaws.services.rds.auth.GetIamAuthTokenRequest;
import com.amazonaws.services.rds.auth.RdsIamAuthTokenGenerator;

public class GenerateRDSAuthToken {

    public static void main(String[] args) {

     String region = "us-west-2";
     String hostname = "rdsmysql.123456789012.us-west-2.rds.amazonaws.com";
     String port = "3306";
     String username = "jane_doe";

     System.out.println(generateAuthToken(region, hostname, port, username));
    }

    static String generateAuthToken(String region, String hostName, String port, String
 username) {

     RdsIamAuthTokenGenerator generator = RdsIamAuthTokenGenerator.builder()
      .credentials(new DefaultAWSCredentialsProviderChain())
      .region(region)
      .build();

     String authToken = generator.getAuthToken(
      GetIamAuthTokenRequest.builder()
```

```
      .hostname(hostName)
      .port(Integer.parseInt(port))
      .userName(username)
      .build());

     return authToken;
    }

}
```

Creazione manuale di un token di autenticazione IAM

In Java, il modo più semplice di generare un token di autenticazione è di utilizzare `RdsIamAuthTokenGenerator`. Questa classe crea un token di autenticazione e lo firma utilizzando AWS Signature Version 4. Per ulteriori informazioni, consulta la pagina relativa al processo di firma Signature Version 4 nella Riferimenti generali di AWS.

Tuttavia, puoi anche creare e firmare un token di autenticazione manualmente, come visualizzato nel seguente esempio di codice.

```
package com.amazonaws.codesamples;

import com.amazonaws.SdkClientException;
import com.amazonaws.auth.DefaultAWSCredentialsProviderChain;
import com.amazonaws.auth.SigningAlgorithm;
import com.amazonaws.util.BinaryUtils;
import org.apache.commons.lang3.StringUtils;

import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;
import java.nio.charset.Charset;
import java.security.MessageDigest;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.SortedMap;
import java.util.TreeMap;

import static com.amazonaws.auth.internal.SignerConstants.AWS4_TERMINATOR;
import static com.amazonaws.util.StringUtils.UTF8;

public class CreateRDSAuthTokenManually {
    public static String httpMethod = "GET";
    public static String action = "connect";
```

```
    public static String canonicalURIParameter = "/";
    public static SortedMap<String, String> canonicalQueryParameters = new TreeMap();
    public static String payload = StringUtils.EMPTY;
    public static String signedHeader = "host";
    public static String algorithm = "AWS4-HMAC-SHA256";
    public static String serviceName = "rds-db";
    public static String requestWithoutSignature;

    public static void main(String[] args) throws Exception {

        String region = "us-west-2";
        String instanceName = "rdsmysql.123456789012.us-west-2.rds.amazonaws.com";
        String port = "3306";
        String username = "jane_doe";

        Date now = new Date();
        String date = new SimpleDateFormat("yyyyMMdd").format(now);
        String dateTimeStamp = new
SimpleDateFormat("yyyyMMdd'T'HHmmss'Z'").format(now);
        DefaultAWSCredentialsProviderChain creds = new
DefaultAWSCredentialsProviderChain();
    String awsAccessKey = creds.getCredentials().getAWSAccessKeyId();
    String awsSecretKey = creds.getCredentials().getAWSSecretKey();
        String expiryMinutes = "900";

        System.out.println("Step 1:  Create a canonical request:");
        String canonicalString = createCanonicalString(username, awsAccessKey, date,
dateTimeStamp, region, expiryMinutes, instanceName, port);
        System.out.println(canonicalString);
        System.out.println();

        System.out.println("Step 2:  Create a string to sign:");
        String stringToSign = createStringToSign(dateTimeStamp, canonicalString,
awsAccessKey, date, region);
        System.out.println(stringToSign);
        System.out.println();

        System.out.println("Step 3:  Calculate the signature:");
        String signature = BinaryUtils.toHex(calculateSignature(stringToSign,
newSigningKey(awsSecretKey, date, region, serviceName)));
        System.out.println(signature);
        System.out.println();
```

```
        System.out.println("Step 4:  Add the signing info to the request");

        System.out.println(appendSignature(signature));
        System.out.println();

    }

    //Step 1: Create a canonical request date should be in format YYYYMMDD and dateTime
should be in format YYYYMMDDTHHMMSSZ
    public static String createCanonicalString(String user, String accessKey, String
date, String dateTime, String region, String expiryPeriod, String hostName, String
port) throws Exception {
        canonicalQueryParameters.put("Action", action);
        canonicalQueryParameters.put("DBUser", user);
        canonicalQueryParameters.put("X-Amz-Algorithm", "AWS4-HMAC-SHA256");
        canonicalQueryParameters.put("X-Amz-Credential", accessKey + "%2F" + date +
"%2F" + region + "%2F" + serviceName + "%2Faws4_request");
        canonicalQueryParameters.put("X-Amz-Date", dateTime);
        canonicalQueryParameters.put("X-Amz-Expires", expiryPeriod);
        canonicalQueryParameters.put("X-Amz-SignedHeaders", signedHeader);
        String canonicalQueryString = "";
        while(!canonicalQueryParameters.isEmpty()) {
            String currentQueryParameter = canonicalQueryParameters.firstKey();
            String currentQueryParameterValue =
canonicalQueryParameters.remove(currentQueryParameter);
            canonicalQueryString = canonicalQueryString + currentQueryParameter + "=" +
currentQueryParameterValue;
            if (!currentQueryParameter.equals("X-Amz-SignedHeaders")) {
                canonicalQueryString += "&";
            }
        }
        String canonicalHeaders = "host:" + hostName + ":" + port + '\n';
        requestWithoutSignature = hostName + ":" + port + "/?" + canonicalQueryString;

        String hashedPayload = BinaryUtils.toHex(hash(payload));
        return httpMethod + '\n' + canonicalURIParameter + '\n' + canonicalQueryString
+ '\n' + canonicalHeaders + '\n' + signedHeader + '\n' + hashedPayload;

    }

    //Step 2: Create a string to sign using sig v4
    public static String createStringToSign(String dateTime, String canonicalRequest,
String accessKey, String date, String region) throws Exception {
```

```
        String credentialScope = date + "/" + region + "/" + serviceName + "/
aws4_request";
        return algorithm + '\n' + dateTime + '\n' + credentialScope + '\n' +
 BinaryUtils.toHex(hash(canonicalRequest));

    }

    //Step 3: Calculate signature
    /**
     * Step 3 of the &AWS; Signature version 4 calculation. It involves deriving
     * the signing key and computing the signature. Refer to
     * http://docs.aws.amazon
     * .com/general/latest/gr/sigv4-calculate-signature.html
     */
    public static byte[] calculateSignature(String stringToSign,
                                            byte[] signingKey) {
        return sign(stringToSign.getBytes(Charset.forName("UTF-8")), signingKey,
                SigningAlgorithm.HmacSHA256);
    }

    public static byte[] sign(byte[] data, byte[] key,
                          SigningAlgorithm algorithm) throws SdkClientException {
        try {
            Mac mac = algorithm.getMac();
            mac.init(new SecretKeySpec(key, algorithm.toString()));
            return mac.doFinal(data);
        } catch (Exception e) {
            throw new SdkClientException(
                    "Unable to calculate a request signature: "
                            + e.getMessage(), e);
        }
    }

    public static byte[] newSigningKey(String secretKey,
                                   String dateStamp, String regionName, String
 serviceName) {
        byte[] kSecret = ("AWS4" + secretKey).getBytes(Charset.forName("UTF-8"));
        byte[] kDate = sign(dateStamp, kSecret, SigningAlgorithm.HmacSHA256);
        byte[] kRegion = sign(regionName, kDate, SigningAlgorithm.HmacSHA256);
        byte[] kService = sign(serviceName, kRegion,
                SigningAlgorithm.HmacSHA256);
        return sign(AWS4_TERMINATOR, kService, SigningAlgorithm.HmacSHA256);
    }
```

```
    public static byte[] sign(String stringData, byte[] key,
                      SigningAlgorithm algorithm) throws SdkClientException {
        try {
            byte[] data = stringData.getBytes(UTF8);
            return sign(data, key, algorithm);
        } catch (Exception e) {
            throw new SdkClientException(
                    "Unable to calculate a request signature: "
                            + e.getMessage(), e);
        }
    }

    //Step 4: append the signature
    public static String appendSignature(String signature) {
        return requestWithoutSignature + "&X-Amz-Signature=" + signature;
    }

    public static byte[] hash(String s) throws Exception {
        try {
            MessageDigest md = MessageDigest.getInstance("SHA-256");
            md.update(s.getBytes(UTF8));
            return md.digest();
        } catch (Exception e) {
            throw new SdkClientException(
                    "Unable to compute hash while signing request: "
                            + e.getMessage(), e);
        }
    }
}
```

Connessione a un'istanza database

Il seguente esempio di codice mostra come generare un token di autenticazione e utilizzarlo per eseguire la connessione a un'istanza eseguendo MariaDB o MySQL.

Per eseguire questo codice di esempio è necessario AWS SDK for Java, disponibile sul sito AWS. Inoltre, hai bisogno di quanto segue:

- MySQL Connector/J. Questo esempio di codice è stato testato con `mysql-connector-java-5.1.33-bin.jar`.

- Un certificato intermedio per Amazon RDS specifico per una regione AWS. Per ulteriori informazioni, consult .) Durante il runtime, il loader della classe cerca un certificato nella stessa directory di questo esempio di codice Java, per permettere al loader della classe di trovarlo.
- Modifica i valori delle variabili seguenti in base alle esigenze.
  - `RDS_INSTANCE_HOSTNAME`: il nome host dell'istanza database cui vuoi accedere.
  - `RDS_INSTANCE_PORT`: il numero di porta usato per la connessione all'istanza database PostgreSQL.
  - `REGION_NAME`: la regione AWS in cui è in esecuzione l'istanza database.
  - `DB_USER`: l'account database cui vuoi accedere.
  - `SSL_CERTIFICATE`: un certificato intermedio per Amazon RDS specifico per una regione AWS.

    Per scaricare un certificato per la regione AWS, consulta . Inserisci il certificato SSL nella stessa directory di questo file di programma Java, per permettere al loader di classe di trovare il certificato durante il runtime.

Questo esempio di codice ottiene le credenziali AWS dalla catena di provider delle credenziali predefinita.

> **Note**
>
> Specifica una password per `DEFAULT_KEY_STORE_PASSWORD` diversa da quella mostrata qui come best practice di sicurezza.

```
package com.amazonaws.samples;

import com.amazonaws.services.rds.auth.RdsIamAuthTokenGenerator;
import com.amazonaws.services.rds.auth.GetIamAuthTokenRequest;
import com.amazonaws.auth.BasicAWSCredentials;
import com.amazonaws.auth.DefaultAWSCredentialsProviderChain;
import com.amazonaws.auth.AWSStaticCredentialsProvider;

import java.io.File;
import java.io.FileOutputStream;
import java.io.InputStream;
import java.security.KeyStore;
import java.security.cert.CertificateFactory;
import java.security.cert.X509Certificate;
```

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
import java.util.Properties;

import java.net.URL;

public class IAMDatabaseAuthenticationTester {
    //&AWS; Credentials of the IAM user with policy enabling IAM Database Authenticated
 access to the db by the db user.
    private static final DefaultAWSCredentialsProviderChain creds = new
 DefaultAWSCredentialsProviderChain();
    private static final String AWS_ACCESS_KEY =
 creds.getCredentials().getAWSAccessKeyId();
    private static final String AWS_SECRET_KEY =
 creds.getCredentials().getAWSSecretKey();

    //Configuration parameters for the generation of the IAM Database Authentication
 token
    private static final String RDS_INSTANCE_HOSTNAME = "rdsmysql.123456789012.us-
west-2.rds.amazonaws.com";
    private static final int RDS_INSTANCE_PORT = 3306;
    private static final String REGION_NAME = "us-west-2";
    private static final String DB_USER = "jane_doe";
    private static final String JDBC_URL = "jdbc:mysql://" + RDS_INSTANCE_HOSTNAME +
 ":" + RDS_INSTANCE_PORT;

    private static final String SSL_CERTIFICATE = "rds-ca-2019-us-west-2.pem";

    private static final String KEY_STORE_TYPE = "JKS";
    private static final String KEY_STORE_PROVIDER = "SUN";
    private static final String KEY_STORE_FILE_PREFIX = "sys-connect-via-ssl-test-
cacerts";
    private static final String KEY_STORE_FILE_SUFFIX = ".jks";
    private static final String DEFAULT_KEY_STORE_PASSWORD = "changeit";

    public static void main(String[] args) throws Exception {
        //get the connection
        Connection connection = getDBConnectionUsingIam();

        //verify the connection is successful
        Statement stmt= connection.createStatement();
```

```
        ResultSet rs=stmt.executeQuery("SELECT 'Success!' FROM DUAL;");
        while (rs.next()) {
             String id = rs.getString(1);
            System.out.println(id); //Should print "Success!"
        }

        //close the connection
        stmt.close();
        connection.close();

        clearSslProperties();

    }

    /**
     * This method returns a connection to the db instance authenticated using IAM
Database Authentication
     * @return
     * @throws Exception
     */
    private static Connection getDBConnectionUsingIam() throws Exception {
        setSslProperties();
        return DriverManager.getConnection(JDBC_URL, setMySqlConnectionProperties());
    }

    /**
     * This method sets the mysql connection properties which includes the IAM Database
Authentication token
     * as the password. It also specifies that SSL verification is required.
     * @return
     */
    private static Properties setMySqlConnectionProperties() {
        Properties mysqlConnectionProperties = new Properties();
        mysqlConnectionProperties.setProperty("verifyServerCertificate","true");
        mysqlConnectionProperties.setProperty("useSSL", "true");
        mysqlConnectionProperties.setProperty("user",DB_USER);
        mysqlConnectionProperties.setProperty("password",generateAuthToken());
        return mysqlConnectionProperties;
    }

    /**
     * This method generates the IAM Auth Token.
     * An example IAM Auth Token would look like follows:
```

```
     * btusi123.cmz7kenwo2ye.rds.cn-north-1.amazonaws.com.cn:3306/?
Action=connect&DBUser=iamtestuser&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-
Date=20171003T010726Z&X-Amz-SignedHeaders=host&X-Amz-Expires=899&X-Amz-
Credential=AKIAPFXHGVDI5RNFO4AQ%2F20171003%2Fcn-north-1%2Frds-db%2Faws4_request&X-Amz-
Signature=f9f45ef96c1f770cdad11a53e33ffa4c3730bc03fdee820cfdf1322eed15483b
     * @return
     */
    private static String generateAuthToken() {
        BasicAWSCredentials awsCredentials = new BasicAWSCredentials(AWS_ACCESS_KEY,
 AWS_SECRET_KEY);

        RdsIamAuthTokenGenerator generator = RdsIamAuthTokenGenerator.builder()
                .credentials(new
 AWSStaticCredentialsProvider(awsCredentials)).region(REGION_NAME).build();
        return generator.getAuthToken(GetIamAuthTokenRequest.builder()

 .hostname(RDS_INSTANCE_HOSTNAME).port(RDS_INSTANCE_PORT).userName(DB_USER).build());
    }

    /**
     * This method sets the SSL properties which specify the key store file, its type
 and password:
     * @throws Exception
     */
    private static void setSslProperties() throws Exception {
        System.setProperty("javax.net.ssl.trustStore", createKeyStoreFile());
        System.setProperty("javax.net.ssl.trustStoreType", KEY_STORE_TYPE);
        System.setProperty("javax.net.ssl.trustStorePassword",
 DEFAULT_KEY_STORE_PASSWORD);
    }

    /**
     * This method returns the path of the Key Store File needed for the SSL
 verification during the IAM Database Authentication to
     * the db instance.
     * @return
     * @throws Exception
     */
    private static String createKeyStoreFile() throws Exception {
        return createKeyStoreFile(createCertificate()).getPath();
    }

    /**
     *  This method generates the SSL certificate
```

```
     * @return
     * @throws Exception
     */
    private static X509Certificate createCertificate() throws Exception {
        CertificateFactory certFactory = CertificateFactory.getInstance("X.509");
        URL url = new File(SSL_CERTIFICATE).toURI().toURL();
        if (url == null) {
            throw new Exception();
        }
        try (InputStream certInputStream = url.openStream()) {
            return (X509Certificate) certFactory.generateCertificate(certInputStream);
        }
    }

    /**
     * This method creates the Key Store File
     * @param rootX509Certificate - the SSL certificate to be stored in the KeyStore
     * @return
     * @throws Exception
     */
    private static File createKeyStoreFile(X509Certificate rootX509Certificate) throws
Exception {
        File keyStoreFile = File.createTempFile(KEY_STORE_FILE_PREFIX,
KEY_STORE_FILE_SUFFIX);
        try (FileOutputStream fos = new FileOutputStream(keyStoreFile.getPath())) {
            KeyStore ks = KeyStore.getInstance(KEY_STORE_TYPE, KEY_STORE_PROVIDER);
            ks.load(null);
            ks.setCertificateEntry("rootCaCertificate", rootX509Certificate);
            ks.store(fos, DEFAULT_KEY_STORE_PASSWORD.toCharArray());
        }
        return keyStoreFile;
    }

    /**
     * This method clears the SSL properties.
     * @throws Exception
     */
    private static void clearSslProperties() throws Exception {
           System.clearProperty("javax.net.ssl.trustStore");
           System.clearProperty("javax.net.ssl.trustStoreType");
           System.clearProperty("javax.net.ssl.trustStorePassword");
    }
```

```
}
```

Se desideri connetterti a un'istanza database tramite un proxy, consulta Connessione a un proxy mediante autenticazione IAM.

Connessione al cluster tramite Autenticazione IAM e AWS SDK for Python (Boto3)

Puoi connetterti a un'istanza database RDS for MariaDB, MySQL o PostgreSQL con la AWS SDK for Python (Boto3) come descritto di seguito.

Prerequisiti

Di seguito sono riportati i prerequisiti per la connessione al di istanzaDB utilizzando l'autenticazione IAM:

- Abilitazione e disabilitazione dell'autenticazione database IAM
- Creazione e utilizzo di una policy IAM per l'accesso al database IAM
- Creazione di un account database tramite l'autenticazione IAM

Inoltre, assicurarsi che le librerie importate nel codice di esempio esistano nel sistema.

Esempi

Gli esempi di codice utilizzano i profili per le credenziali condivise. Per informazioni sulla specifica delle credenziali, vedere Credenziali nella documentazione di AWS SDK for Python (Boto3).

Il seguente esempio di codice mostra come generare un token di autenticazione e utilizzarlo per eseguire la connessione a un'istanza del database.

Per eseguire questo codice di esempio è necessario AWS SDK for Python (Boto3), disponibile sul sito AWS.

Modifica i valori delle variabili seguenti in base alle esigenze.

- `ENDPOINT`: l'endpoint dell'istanza cui vuoi accedere
- `PORT` – Numero di porta usato per la connessione al cluster
- `USER` – L'account database cui vuoi accedere.
- `REGION`: la regione AWS in cui è in esecuzione l'istanza

- `DBNAME` – Database a cui accedere.
- `SSLCERTIFICATE` - Percorso completo del certificato SSL per Amazon RDS

  Per `ssl_ca`, specificare un certificato SSL. Per scaricare un certificato SSL consulta

> **Note**
>
> Non è possibile utilizzare un record DNS Route 53 personalizzato anziché l'endpoint dell'istanza database per generare il token di autenticazione.

Questo codice si connette a un'istanza database MariaDB o MySQL.

Prima di eseguire questo codice, installa il driver PyMySQL seguendo le istruzioni in Python Package Index (Indice dei pacchetti Python).

```
import pymysql
import sys
import boto3
import os

ENDPOINT="mysqldb.123456789012.us-east-1.rds.amazonaws.com"
PORT="3306"
USER="jane_doe"
REGION="us-east-1"
DBNAME="mydb"
os.environ['LIBMYSQL_ENABLE_CLEARTEXT_PLUGIN'] = '1'

#gets the credentials from .aws/credentials
session = boto3.Session(profile_name='default')
client = session.client('rds')

token = client.generate_db_auth_token(DBHostname=ENDPOINT, Port=PORT, DBUsername=USER,
 Region=REGION)

try:
    conn =  pymysql.connect(host=ENDPOINT, user=USER, passwd=token, port=PORT,
 database=DBNAME, ssl_ca='SSLCERTIFICATE')
    cur = conn.cursor()
    cur.execute("""SELECT now()""")
    query_results = cur.fetchall()
```

```
    print(query_results)
except Exception as e:
    print("Database connection failed due to {}".format(e))
```

Questo codice si connette a un'istanza database PostgreSQL.

Prima di eseguire questo codice, installapsycopg2seguendo le istruzioni in Documentazione di Psycopg.

```
import psycopg2
import sys
import boto3
import os

ENDPOINT="postgresmydb.123456789012.us-east-1.rds.amazonaws.com"
PORT="5432"
USER="jane_doe"
REGION="us-east-1"
DBNAME="mydb"

#gets the credentials from .aws/credentials
session = boto3.Session(profile_name='RDSCreds')
client = session.client('rds')

token = client.generate_db_auth_token(DBHostname=ENDPOINT, Port=PORT, DBUsername=USER,
 Region=REGION)

try:
    conn = psycopg2.connect(host=ENDPOINT, port=PORT, database=DBNAME, user=USER,
 password=token, sslrootcert="SSLCERTIFICATE")
    cur = conn.cursor()
    cur.execute("""SELECT now()""")
    query_results = cur.fetchall()
    print(query_results)
except Exception as e:
    print("Database connection failed due to {}".format(e))
```

Se desideri connetterti a un'istanza database tramite un proxy, consulta Connessione a un proxy mediante autenticazione IAM.

Risoluzione dei problemi di identità e accesso in Amazon RDS

Utilizza le informazioni seguenti per diagnosticare e risolvere i problemi comuni che possono verificarsi durante l'utilizzo di Amazon RDS e IAM.

Argomenti

Non sono autorizzato a eseguire un'operazione in Amazon RDS

Se la AWS Management Console indica che non sei autorizzato a eseguire un'operazione, devi contattare l'amministratore per ricevere assistenza. L'amministratore è colui che ti ha fornito le credenziali di accesso.

L'errore di esempio seguente si verifica quando l'utente `mateojackson` cerca di utilizzare la console per visualizzare i dettagli relativi a un *`widget`*, ma non dispone di autorizzazioni `rds:`*`GetWidget`*.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 rds:GetWidget on resource: my-example-widget
```

In questo caso, Mateo richiede al suo amministratore di aggiornare le sue policy per poter accedere alla risorsa *`my-example-widget`* utilizzando l'operazione `rds:`*`GetWidget`*.

Non sono autorizzato a eseguire: iam:PassRole

Se ricevi un errore che indica che non sei autorizzato a eseguire l'operazione `iam:PassRole`, devi contattare il tuo amministratore per ricevere assistenza. L'amministratore è colui che ti ha fornito le credenziali di accesso. Richiedi a tale persona di aggiornare le tue policy per poter passare un ruolo a Amazon RDS.

Alcuni servizi AWS consentono di passare un ruolo esistente a tale servizio, invece di creare un nuovo ruolo del servizio o ruolo collegato ai servizi. Per eseguire questa operazione, è necessario disporre delle autorizzazioni per passare il ruolo al servizio.

L'errore di esempio seguente si verifica quando un utente denominato `marymajor` cerca di utilizzare la console per eseguire un'operazione in Amazon RDS. Tuttavia, l'operazione richiede che il servizio

disponga delle autorizzazioni concesse da un ruolo di servizio. Mary non dispone di autorizzazioni per passare il ruolo al servizio.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

In questo caso, Mary chiede all'amministratore di aggiornare la sue policy per poter eseguire l'operazione `iam:PassRole`.

Voglio consentire a persone esterne al mio account AWS di accedere alle mie risorse Amazon RDS

È possibile creare un ruolo con il quale utenti in altri account o persone esterne all'organizzazione possono accedere alle tue risorse. È possibile specificare chi è attendibile per l'assunzione del ruolo. Per servizi che supportano policy basate su risorse o liste di controllo accessi (ACL), utilizza tali policy per concedere alle persone l'accesso alle tue risorse.

Per ulteriori informazioni, consulta gli argomenti seguenti:

- Per capire se Amazon RDS supporta queste funzionalità, consulta Funzionamento di Amazon RDS con IAM.
- Per informazioni su come garantire l'accesso alle risorse negli account AWS che possiedi, consulta Fornire l'accesso a un utente IAM in un altro account AWS che si possiede nella Guida per l'utente di IAM.
- Per capire come fornire l'accesso alle risorse ad account AWS di terze parti, consulta Concessione dell'accesso agli account AWS di proprietà di terze parti nella Guida per l'utente di IAM.
- Per capire come fornire l'accesso tramite la federazione delle identità, consulta Fornire accesso a utenti autenticati esternamente (Federazione delle identità) nella Guida per l'utente di IAM.
- Per informazioni sulle differenze tra l'utilizzo di ruoli e policy basate su risorse per l'accesso multi-account, consultare Differenza tra i ruoli IAM e le policy basate su risorse nella Guida per l'utente di IAM.

Registrazione e monitoraggio in Amazon RDS

Il monitoraggio è importante per garantire l'affidabilità, la disponibilità e le prestazioni di Amazon RDS e delle soluzioni AWS. È consigliabile raccogliere dati di monitoraggio da tutte le parti della soluzione

AWS per eseguire più facilmente il debug di guasti in più punti nel caso si verifichino. AWS fornisce diversi strumenti per il monitoraggio delle risorse Amazon RDS e la risposta a potenziali incidenti:

CloudWatch Allarmi Amazon

Utilizzando Amazon CloudWatch alarms, controlli una singola metrica per un periodo di tempo specificato. Se la metrica supera una determinata soglia, viene inviata una notifica a un argomento o una policy di Amazon SNS. AWS Auto Scaling CloudWatch gli allarmi non richiamano azioni perché si trovano in uno stato particolare. È necessario invece cambiare lo stato e mantenerlo per un numero di periodi specificato.

AWS CloudTrailLog di

CloudTrail fornisce un registro delle azioni intraprese da un utente, un ruolo o un AWS servizio in Amazon RDS . CloudTrail acquisisce tutte le chiamate API per Amazon RDS come eventi, incluse le chiamate dalla console e le chiamate di codice alle operazioni dell'API Amazon RDS. Utilizzando le informazioni raccolte da CloudTrail, puoi determinare la richiesta effettuata ad Amazon RDS Aurora, l'indirizzo IP da cui è stata effettuata la richiesta, chi ha effettuato la richiesta, quando è stata effettuata e dettagli aggiuntivi. Per ulteriori informazioni, consulta Monitoraggio delle chamate API di Amazon RDS in AWS CloudTrail.

Enhanced Monitoring

Amazon RDS fornisce parametri in tempo reale per il sistema operativo sul quale è in esecuzione il cluster di . Puoi visualizzare i parametri per il tuo di istanze DB utilizzando la console o utilizzare l'output JSON di Enhanced Monitoring di Amazon CloudWatch Logs in un sistema di monitoraggio a tua scelta. Per ulteriori informazioni, consulta Monitoraggio dei parametri del sistema operativo con il monitoraggio avanzato.

Performance Insights di Amazon RDS

Performance Insights analizza le funzionalità di monitoraggio esistenti Amazon RDS per illustrare le prestazioni del database e aiutare ad analizzare eventuali problemi che lo riguardano. Con il pannello di controllo di Performance Insights, è possibile visualizzare il carico del database e filtrare il carico in base alle attese, alle istruzioni SQL, agli host o agli utenti. Per ulteriori informazioni, consulta Monitoraggio del carico DB con Performance Insights su Amazon RDS.

Log di database

Puoi visualizzare, scaricare e controllare i log di database tramite AWS Management Console, AWS CLI o l'API RDS. Per ulteriori informazioni, consulta Monitoraggio dei file di log di Amazon RDS.

Raccomandazioni Amazon RDS

Amazon RDS offre consigli automatici per le risorse di database. Queste raccomandazioni forniscono consigli sulle best practice analizzandola configurazione, l'utilizzo e i dati di prestazione del cluster di . Per ulteriori informazioni, consulta Visualizzazione e risposta ai consigli di RDS.

Notifiche di eventi Amazon RDS

Amazon RDS utilizza Amazon Simple Notification Service (Amazon SNS) per fornire una notifica quando si verifica un evento Amazon RDS. Queste notifiche possono essere in qualsiasi forma supportata da Amazon SNS per una regione AWS, ad esempio un'e-mail, un SMS o una chiamata a un endpoint HTTP. Per ulteriori informazioni, consulta Utilizzo della notifica degli eventi di Amazon RDS.

AWS Trusted Advisor

Trusted Advisor sfrutta le best practice acquisite servendo centinaia di migliaia di clienti AWS. Trusted Advisor controlla l'ambiente AWS, quindi fornisce suggerimenti nel caso in cui vi siano opportunità di risparmio, di miglioramento delle prestazioni e della disponibilità dei sistemi o di risoluzione dei problemi di sicurezza. Tutti i clienti AWS hanno accesso a cinque controlli di Trusted Advisor. I clienti che hanno sottoscritto un piano di supporto Business o Enterprise possono visualizzare tutti i controlli di Trusted Advisor.

Trusted Advisor dispone dei seguenti controlli correlati a Amazon RDS:

- Istanze database Amazon RDS inattive
- Rischio accesso gruppo di sicurezza Amazon RDS
- Backup Amazon RDS
- Multi-AZ Amazon RDS

Per ulteriori informazioni su questi controlli, consulta best practice Trusted Advisor (Controlli).

Per ulteriori informazioni sul monitoraggio di Amazon RDS, consulta Monitoraggio di parametri in un'istanza Amazon RDS.

Convalida della conformità per Amazon RDS

Revisori di terze parti valutano la sicurezza e la conformità di Amazon RDS come parte di più programmi di conformità di AWS. Questi includono SOC, PCI, FedRAMP, HIPAA e altri.

Per un elenco di servizi AWS che rientrano nell'ambito di programmi di conformità specifici, consultare Servizi AWS coperti dal programma di compliance. Per informazioni generali, consultare Programmi per la conformità di AWS.

Puoi scaricare i report di audit di terze parti utilizzando AWS Artifact. Per ulteriori informazioni, consulta l'argomento Download dei rapporti in AWS Artifact.

La responsabilità per la conformità quando utilizzi Amazon RDS è determinata dalla riservatezza dei dati, dagli obiettivi di conformità dell'azienda e dalle normative vigenti. AWS fornisce le risorse seguenti per semplificare la conformità:

- Guide Quick Start per la sicurezza e conformità: queste guide all'implementazione illustrano considerazioni relative all'architettura e forniscono fasi per l'implementazione di ambienti di base incentrati sulla sicurezza e sulla conformità su AWS.
- Architecting for HIPAA Security and Compliance on Amazon Web Services (Architettare per la sicurezza e la conformità HIPAA su Amazon Web Services): questo whitepaper descrive come le aziende possono utilizzare AWS per creare applicazioni conformi alla normativa HIPAA.
- Risorse per la conformità AWS: questa raccolta di cartelle di lavoro e guide potrebbe essere utile per il settore e la posizione.
- AWS Config: questo servizio AWS valuta il livello di conformità delle configurazioni delle risorse con pratiche interne, linee guida e regolamenti di settore.
- AWS Security Hub: questo Servizio AWS fornisce una visione completa dello stato di sicurezza all'interno di AWS. La Centrale di sicurezza utilizza i controlli di sicurezza per valutare le risorse AWS e verificare la conformità agli standard e alle best practice del settore della sicurezza. Per un elenco dei servizi e dei controlli supportati, consulta la pagina Documentazione di riferimento sui controlli della Centrale di sicurezza.

Resilienza in Amazon RDS

L'infrastruttura globale di AWS è basata su regioni AWS e zone di disponibilità. AWS Le Regioni forniscono più zone di disponibilità fisicamente separate e isolate che sono connesse tramite reti altamente ridondanti, a bassa latenza e velocità effettiva elevata. Con le zone di disponibilità, è possibile progettare e gestire le applicazioni e database che eseguono il failover automatico tra zone di disponibilità senza interruzioni. Le zone di disponibilità sono più disponibili, tolleranti ai guasti e scalabili rispetto alle infrastrutture a data center singolo o multiplo.

Per ulteriori informazioni sulle regioni e le zone di disponibilità AWS, consulta Infrastruttura globale di AWS.

Oltre all'infrastruttura globale AWS, Amazon RDS offre numerose funzionalità per supportare la resilienza dei dati e le esigenze di backup.

Backup e ripristino

Amazon RDS crea e salva i backup automatici dell'istanza database. Amazon RDS crea uno snapshot dei volumi di storage dell'istanza database, eseguendo il backup dell'intera istanza database anziché dei singoli database.

Amazon RDS crea e salva backup automatici dell'istanza database durante la finestra di backup dell'istanza database. Amazon RDS salva i backup automatici dell'istanza database in base al tempo di conservazione del backup specificato. Se necessario, puoi ripristinare il tuo database a un point-in-time specifico durante il tempo di conservazione del backup. È inoltre possibile eseguire manualmente il backup dell'istanza database creando snapshot DB.

Se l'istanza database di origine fallisce, è possibile creare un'istanza database eseguendo il ripristino dallo snapshot DB come soluzione di ripristino di emergenza.

Per ulteriori informazioni, consulta Backup, ripristino ed esportazione dei dati.

Replica

Amazon RDS usa la funzionalità di replica integrata dei motori di database MariaDB, MySQL, Oracle e PostgreSQL per creare un tipo speciale di istanza database denominato replica di lettura da un'istanza database di origine. Gli aggiornamenti applicati all'istanza database di origine vengono copiati in modo asincrono nella replica di lettura. Puoi ridurre il carico sull'istanza database di origine effettuando il routing le query di lettura dalle applicazioni alla replica di lettura. Tramite le repliche

di lettura puoi aumentare orizzontalmente in modo elastico la capacità oltre i vincoli di una singola istanza database per carichi di lavoro di database particolarmente gravosi in lettura. Puoi promuovere una replica di lettura a un'istanza standalone come soluzione di ripristino di emergenza in caso di errore dell'istanza database di origine. Per alcuni motori DB, Amazon RDS supporta inoltre altre opzioni di replica.

Per ulteriori informazioni, consulta Uso delle repliche di lettura dell'istanza database.

Failover

Amazon RDS offre disponibilità elevata e supporto per il failover per le istanze database tramite le implementazioni Multi-AZ. Amazon RDS utilizza varie tecnologie differenti per garantire il supporto per il failover. Le implementazioni Multi-AZ per le istanze database Oracle, PostgreSQL, MySQL e MariaDB utilizzano la tecnologia di failover di Amazon. Le istanze database SQL Server utilizzano il mirroring del database (DBM) di SQL Server.

Per ulteriori informazioni, consulta Configurazione e gestione di un'implementazione multi-AZ.

Sicurezza dell'infrastruttura in Amazon RDS

In quanto servizio gestito, Amazon Relational Database Service è protetto dalla sicurezza della rete globale AWS. Per informazioni sui servizi di sicurezza AWS e su come AWS protegge l'infrastruttura, consulta la pagina Sicurezza del cloud AWS. Per progettare l'ambiente AWS utilizzando le best practice per la sicurezza dell'infrastruttura, consulta la pagina Protezione dell'infrastruttura nel Pilastro della sicurezza di AWS Well-Architected Framework.

Utilizzare le chiamate API pubblicate di AWS per accedere ad Amazon RDS tramite la rete. I clienti devono supportare quanto segue:

- Transport Layer Security (TLS). È richiesto TLS 1.2 ed è consigliato TLS 1.3.
- Suite di cifratura con Perfect Forward Secrecy (PFS), ad esempio Ephemeral Diffie-Hellman (DHE) o Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). La maggior parte dei sistemi moderni, come Java 7 e versioni successive, supporta tali modalità.

Inoltre, le richieste devono essere firmate utilizzando un ID chiave di accesso e una chiave di accesso segreta associata a un principale IAM. In alternativa, è possibile utilizzare AWS Security Token Service (AWS STS) per generare le credenziali di sicurezza temporanee per sottoscrivere le richieste.

Inoltre, Amazon RDS offre funzionalità che contribuiscono a supportare la sicurezza dell'infrastruttura.

Gruppi di sicurezza

I gruppi di sicurezza controllano l'accesso del traffico in entrata e in uscita di un'istanza database. Per impostazione predefinita, l'accesso alla rete è disattivato per un'istanza database. Puoi specificare delle norme in un gruppo di sicurezza che consente l'accesso da un intervallo di indirizzi IP, dalla porta o da un gruppo di sicurezza. Una volta configurate le regole in ingresso, si applicano le stesse regole a tutte le istanze database con associazione a tale gruppo di sicurezza.

Per ulteriori informazioni, consulta Controllo dell'accesso con i gruppi di sicurezza.

Public accessibility (Accesso pubblico)

Quando si avvia un'istanza database all'interno di un cloud privato virtuale (VPC) in base al servizio Amazon VPC, è possibile attivare o disattivare l'accessibilità pubblica per tale istanza database. Per stabilire se l'istanza database creata ha un nome DNS che si risolve in un indirizzo IP pubblico,

utilizza il parametro Public accessibility (Accessibilità pubblica). Questo parametro consente di stabilire se esiste un accesso pubblico all'istanza database. È possibile modificare un'istanza database per attivare o disattivare l'accessibilità pubblica modificando il parametro Public accessibility (Accessibilità pubblica).

Per ulteriori informazioni, consulta Nascondere istanze database in un VPC da Internet.

> **Note**
>
> Se l'istanza del DB si trova in un VPC ma non è accessibile pubblicamente, puoi anche utilizzare una connessione AWS Site-to-Site VPN o una connessione AWS Direct Connect per accedervi da una rete privata. Per ulteriori informazioni, consulta Riservatezza del traffico Internet.

API Amazon RDS ed endpoint VPC dell'interfaccia (AWS PrivateLink)

È possibile stabilire una connessione privata tra gli endpoint VPC e l'API Amazon RDS creando un endpoint VPC di interfaccia. Endpoint di interfaccia con tecnologia AWS PrivateLink.

AWS PrivateLink consente di accedere in modo privato alle operazioni delle API di Amazon RDS senza un gateway Internet, un dispositivo NAT, una connessione VPN o una connessione. AWS Direct Connect Le istanze database nel VPC non necessitano di indirizzi IP pubblici per comunicare con gli endpoint dell'API Amazon RDS per avviare, modificare o terminare le istanze database. Inoltre, le istanze database non richiedono indirizzi IP pubblici per utilizzare le operazioni dell'API RDS disponibili. Il traffico di rete tra il VPC e Amazon RDS non esce dalla rete Amazon.

Ogni endpoint dell'interfaccia è rappresentato da una o più interfacce di rete elastiche nelle sottoreti. Per maggiori informazioni sulle interfacce di rete elastiche, consulta le interfacce di rete elastiche nella Guida dell'utente di Amazon EC2.

Per ulteriori informazioni sugli endpoint VPC, consulta Interface VPC endpoints () nella Amazon VPC User AWS PrivateLink Guide. Per informazioni sulle operazioni API RDS, consulta Documentazione di riferimento delle API di Amazon RDS.

Non è necessaria un'interfaccia endpoint VPC per connettersi a un'istanza database. Per ulteriori informazioni, consulta Scenari per accedere a un'istanza database in un VPC.

Considerazioni sugli endpoint VPC

Prima di impostare un endpoint VPC di interfaccia per endpoint Amazon RDS API, verificare di esaminare le proprietà e le limitazioni degli endpoint di interfaccia in Guida per l'utente di Amazon VPC.

Tutte le operazioni API RDS rilevanti per la gestione delle risorse Amazon RDS sono disponibili dal VPC utilizzando AWS PrivateLink.

Le policy degli endpoint VPC sono supportate per gli endpoint dell'API RDS. Per impostazione predefinita, l'accesso completo alle operazioni API RDS è consentito attraverso l'endpoint. Per ulteriori informazioni, consulta Controllo degli accessi ai servizi con endpoint VPC in Guida per l'utente di Amazon VPC.

Disponibilità

L'API Amazon RDS attualmente supporta gli endpoint VPC nelle seguenti regioni: AWS

- Stati Uniti orientali (Ohio)
- Stati Uniti orientali (Virginia settentrionale)
- Stati Uniti occidentali (California settentrionale)
- Stati Uniti occidentali (Oregon)
- Africa (Città del Capo)
- Asia Pacific (Hong Kong)
- Asia Pacific (Mumbai)
- Asia Pacific (Osaka)
- Asia Pacific (Seul)
- Asia Pacifico (Singapore)
- Asia Pacifico (Sydney)
- Asia Pacifico (Tokyo)
- Canada (Centrale)
- Canada occidentale (Calgary)
- Cina (Pechino)
- China (Ningxia)
- Europa (Francoforte)
- Europa (Zurigo)
- Europa (Irlanda)
- Europe (London)
- Europe (Paris)
- Europe (Stockholm)
- Europa (Milano)
- Israele (Tel Aviv)
- Medio Oriente (Bahrein)
- Sud America (San Paolo)
- AWS GovCloud (Stati Uniti orientali)
- AWS GovCloud (Stati Uniti occidentali)

Creazione di un endpoint VPC di interfaccia per l'API Amazon RDS

Puoi creare un endpoint VPC per l'API Amazon RDS utilizzando la console Amazon VPC o il (). AWS Command Line Interface AWS CLI Per ulteriori informazioni, consulta Creazione di un endpoint dell'interfaccia nella Guida per l'utente di Amazon VPC.

Crea un endpoint VPC per l'API Amazon RDS utilizzando il nome del servizio `com.amazonaws.region.rds`.

Escludendo AWS le regioni in Cina, se abiliti il DNS privato per l'endpoint, puoi effettuare richieste API ad Amazon RDS con l'endpoint VPC utilizzando ad esempio il nome DNS predefinito per la regione. AWS `rds.us-east-1.amazonaws.com` Per le AWS regioni Cina (Pechino) e Cina (Ningxia), puoi effettuare richieste API con l'endpoint VPC utilizzando e, rispettivamente. `rds-api.cn-north-1.amazonaws.com.cn` `rds-api.cn-northwest-1.amazonaws.com.cn`

Per ulteriori informazioni, consulta Accesso a un servizio tramite un endpoint dell'interfaccia in Guida per l'utente di Amazon VPC.

Creazione di una policy di endpoint VPC per l'API Amazon RDS

Puoi allegare una policy di endpoint all'endpoint VPC che controlla l'accesso all'API Amazon RDS. Questo codice specifica le informazioni riportate di seguito:

- Il principale che può eseguire operazioni.
- Le azioni che possono essere eseguite.
- Le risorse sui cui si possono eseguire operazioni.

Per ulteriori informazioni, consulta Controllo degli accessi ai servizi con endpoint VPC in Guida per l'utente di Amazon VPC.

Ad esempio, policy di endpoint VPC per le operazioni API Amazon RDS

Di seguito è riportato un esempio di una policy endpoint per l'API Amazon RDS. Se collegato a un endpoint, questa policy concede l'accesso alle operazioni API Amazon RDS riportate per tutte le entità su tutte le risorse.

```
{
   "Statement":[
      {
```

```
         "Principal":"*",
         "Effect":"Allow",
         "Action":[
            "rds:CreateDBInstance",
            "rds:ModifyDBInstance",
            "rds:CreateDBSnapshot"
         ],
         "Resource":"*"
      }
   ]
}
```

Esempio: policy degli endpoint VPC che nega tutti gli accessi da un account specificato AWS

La seguente politica degli endpoint VPC nega all' AWS account `123456789012` tutti gli accessi alle risorse che utilizzano l'endpoint. La policy consente tutte le operazioni da altri account.

```
{
  "Statement": [
    {
      "Action": "*",
      "Effect": "Allow",
      "Resource": "*",
      "Principal": "*"
    },
    {
      "Action": "*",
      "Effect": "Deny",
      "Resource": "*",
      "Principal": { "AWS": [ "123456789012" ] }
     }
   ]
}
```

Best practice relative alla sicurezza di Amazon RDS

Usa gli account AWS Identity and Access Management (IAM) per controllare l'accesso alle operazioni dell'API Amazon RDS, in particolare le operazioni che creano, modificano o eliminano risorse Amazon RDS . Tali risorse includono i cluster di , i gruppi di sicurezza e i gruppi di parametri. Utilizza anche IAM per controllare le operazioni che eseguono operazioni amministrative comuni come il backup e il ripristino di cluster di .

- Crea un singolo utente per ogni persona che gestisce le risorse Amazon RDS, incluso te stesso. Non utilizzare credenziali AWS root per gestire le risorse Amazon RDS .
- Assegna a ciascun utente un set minimo di autorizzazioni richieste per eseguire le proprie mansioni.
- Utilizza gruppi IAM per gestire in modo efficace le autorizzazioni per più utenti.
- Ruota periodicamente le credenziali IAM.
- Configura AWS Secrets Manager per ruotare automaticamente i segreti per Amazon RDS Amazon . Per ulteriori informazioni, consulta Rotating your AWS Secrets Manager secrets nella Guida per l'utente.AWS Secrets Manager Puoi anche recuperare le credenziali da programmaticamente. AWS Secrets Manager Per ulteriori informazioni, consulta Recupero del valore segreto nella Guida per l'utente di AWS Secrets Manager .

Per ulteriori informazioni sulla sicurezza di Amazon RDS, consulta Sicurezza in Amazon RDS. Per ulteriori informazioni su IAM, consulta AWS Identity and Access Management. Per informazioni sulle best practice di IAM, consulta Best practice di IAM.

AWS Security Hub utilizza controlli di sicurezza per valutare le configurazioni delle risorse e gli standard di sicurezza per aiutarvi a rispettare vari quadri di conformità. Per ulteriori informazioni sull'utilizzo di Security Hub per valutare le risorse RDS, consulta i controlli di Amazon Relational Database Service nella Guida per AWS Security Hub l'utente.

Puoi monitorare l'uso di RDS in relazione alle best practice sulla sicurezza utilizzando Centrale di sicurezza. Per ulteriori informazioni, consulta What is? AWS Security Hub .

Usa l'API AWS Management Console AWS CLI, the o RDS per modificare la password per il tuo utente principale. Se usi uno strumento diverso, ad esempio un client SQL, per modificare la password dell'utente master, i privilegi per l'utente potrebbero venire revocati involontariamente.

Controllo dell'accesso con i gruppi di sicurezza

I gruppi di sicurezza VPC controllano l'accesso che il traffico ha in entrata e in uscita su un'istanza database. Per impostazione predefinita, l'accesso alla rete è disattivato per un'istanza database. Puoi specificare delle norme in un gruppo di sicurezza che consente l'accesso da un intervallo di indirizzi IP, dalla porta o da un gruppo di sicurezza. Una volta configurate le regole in ingresso, si applicano le stesse regole a tutte le istanze database con associazione a tale gruppo di sicurezza. Puoi specificare fino a 20 norme in un gruppo di sicurezza.

Panoramica dei gruppi di sicurezza VPC

Ogni regola del gruppo di sicurezza VPC consente a un'origine specifica di accedere a un'istanza database in un VPC con associazione al gruppo di sicurezza VPC specifico. L'origine può essere una serie di indirizzi (ad esempio, 203.0.113.0/24) oppure un altro gruppo di sicurezza VPC. Specificando un gruppo di sicurezza VPC come origine, consenti il traffico in entrata da tutte le istanze (in genere i server dell'applicazione) che usano il gruppo di sicurezza VPC. I gruppi di sicurezza VPC possono avere regole che gestiscono sia il traffico in entrata che in uscita. Tuttavia, le regole del traffico in uscita in genere non si applicano alle istanze database. Le regole del traffico in uscita si applicano solo se l'istanza database funge da client. Ad esempio, le regole del traffico in uscita si applicano a un'istanza database di Oracle con collegamenti database in uscita. Per creare gruppi di sicurezza VPC, è necessario utilizzare l'API Amazon EC2 o l'opzione Security Group (Gruppo di sicurezza) nella console VPC.

Quando si creano regole per il gruppo di sicurezza VPC che consentono l'accesso alle istanze nel VPC, è necessario specificare una porta per ciascun intervallo di indirizzi per i quali la regola consente l'accesso. Ad esempio, se si desidera abilitare l'accesso SSH (Secure Shell) alle istanze nel VPC, devi creare una regola che consenta l'accesso alla porta TCP 22 per l'intervallo di indirizzi specificato.

È possibile configurare più gruppi di sicurezza VPC che consentono l'accesso a porte diverse per istanze diverse nel VPC. Ad esempio, è possibile creare un gruppo di sicurezza VPC che consenta l'accesso alla porta TCP 80 per i server web nel VPC. Quindi è possibile creare un altro gruppo di sicurezza VPC che consenta l'accesso alla porta TCP 3306 per le istanze database RDS for MySQL nel VPC.

Per ulteriori informazioni sui gruppi di sicurezza VPC, consulta la pagina Gruppi di sicurezza nella Guida per l'utente di Amazon Virtual Private Cloud.

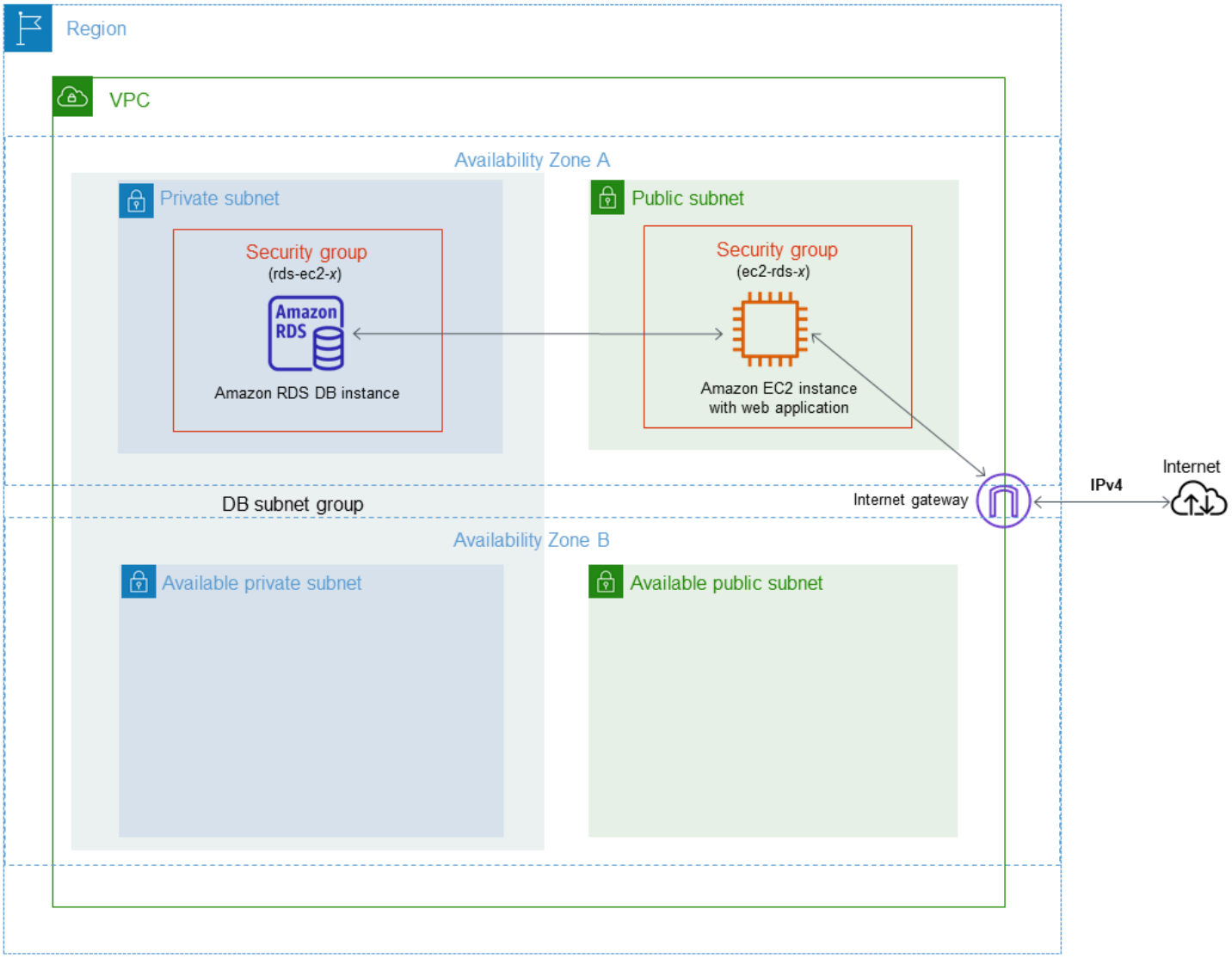
> **Note**
>
> Se il di istanze DB si trova in un VPC ma non è accessibile pubblicamente, puoi anche utilizzare una connessione VPN AWS da sito a sito AWS Direct Connect o una connessione per accedervi da una rete privata. Per ulteriori informazioni, consulta Riservatezza del traffico Internet.

Scenario del gruppo di sicurezza

Un uso comune di un'istanza database in un VPC è quello di condividere dati con un server di applicazione in esecuzione in un'istanza Amazon EC2 nello stesso VPC, al quale si accede tramite un'applicazione client esterna al VPC. Per questo scenario, si utilizzano le pagine RDS e VPC sulla AWS Management Console oppure le azioni API RDS ed EC2 per creare le istanze e i gruppi di sicurezza necessari:

1. Creare un gruppo di sicurezza VPC (ad esempio, `sg-0123ec2example`) e definire le regole in entrata che utilizzano l'indirizzo IP dell'applicazione client usato nell'indirizzo IP dell'applicazione del client come origine. Questo gruppo di sicurezza consente all'applicazione del client di collegarsi alle istanze EC2 in una VPC che utilizza questo gruppo di sicurezza.
2. Creare un'istanza EC2 per l'applicazione e aggiungere l'istanza EC2 al gruppo di sicurezza VPC (`sg-0123ec2example`) creato nel passaggio precedente.
3. Creare un secondo gruppo di sicurezza VPC (ad esempio, `sg-6789rdsexample`) e creare una nuova regola specificando il gruppo di sicurezza VPC creato nel passaggio 1 (`sg-0123ec2example`) come l'origine.
4. Creare una nuova istanza database e aggiungere l'istanza database al gruppo di sicurezza VPC (`sg-6789rdsexample`) creato nel passaggio precedente. Quando crei l'istanza database, utilizza lo stesso numero di porta di quello specificato per la regola del gruppo di sicurezza VPC (`sg-6789rdsexample`) creato nel passaggio 3.

Il seguente diagramma mostra questo scenario.



Per istruzioni dettagliate sulla configurazione di un VPC per questo scenario, consulta Tutorial: Creazione di un Amazon VPC da utilizzare con un'istanza database (solo IPv4). Per ulteriori informazioni sull'utilizzo di un VPC, consulta VPC di Amazon VPC e Amazon RDS.

Creazione di un gruppo di sicurezza VPC

Puoi creare un gruppo di sicurezza VPC per un'istanza database tramite la console VPC. Per informazioni sulla creazione di un gruppo di sicurezza, consulta le pagine Fornisci accesso alla istanza database nel VPC creando un gruppo di sicurezza e Gruppi di sicurezza nella Guida per l'utente di Amazon Virtual Private Cloud.

Associazione di un gruppo di sicurezza a un istanza database

Puoi associare un gruppo di sicurezza a un'istanza DB utilizzando Modify sulla console RDS, l'API `ModifyDBInstance` Amazon RDS o il `modify-db-instance` AWS CLI comando.

Il seguente esempio CLI associa un gruppo di sicurezza VPC specifico e rimuove i gruppi di sicurezza DB dall'istanza DB.

```
aws rds modify-db-instance --db-instance-identifier dbName --vpc-security-group-ids sg-ID
```

Per ulteriori informazioni sulla modifica di un'istanza di database, consulta Modifica di un'istanza database Amazon RDS. Per le considerazioni sui gruppi di sicurezza quando si ripristina un'istanza database da uno snapshot DB, consulta Considerazioni relative al gruppo di sicurezza.

> **Note**
>
> Nella console RDS vengono visualizzati diversi nomi di regole dei gruppi di sicurezza per il database se il valore della porta è configurato su un valore non predefinito.

Per le istanze RDS for Oracle DB, è possibile associare gruppi di sicurezza aggiuntivi compilando l'impostazione delle opzioni del gruppo di sicurezza per le opzioni Oracle Enterprise Manager Database Express (OEM), Oracle Management Agent for Enterprise Manager Cloud Control (OEM Agent) e Oracle Secure Sockets Layer. In questo caso, entrambi i gruppi di sicurezza associati all'istanza DB e le impostazioni delle opzioni si applicano all'istanza DB. Per ulteriori informazioni su questi gruppi di opzioni, consulta Oracle Enterprise ManagerOracle Management Agent per Enterprise Manager Cloud Control, eOracle Secure Sockets Layer.

Privilegi dell'account utente master

Quando viene creato un nuovo oggetto di tipo istanza database, l'utente master predefinito usato ottiene determinati privilegi per tale istanza database. Non è possibile modificare il nome utente master dopo aver creato l'istanza database.

> **Important**
>
> Si consiglia di non utilizzare l'utente master direttamente nelle applicazioni. Rispetta piuttosto la best practice di utilizzare un utente del database creato con i privilegi minimi richiesti per l'applicazione.

> **Note**
>
> Se si cancellano per errore le autorizzazioni per l'utente master è possibile ripristinarle modificando l'instance database e impostando una nuova password per l'utente master. Per ulteriori informazioni sulla modifica di un'istanza database, consulta Modifica di un'istanza database Amazon RDS.

La tabella seguente mostra i privilegi e i ruoli di database ottenuti dall'utente master per ciascuno dei motori di database.

Motore del database	Privilegio del sistema	Ruolo di database
RDS per Db2	L'utente principale viene assegnato al `masterdba` gruppo e gli viene assegnato il. `master_user_role` SYSMON, DBADM con DATAACCESS ANDACCCESSCTRL ,BINDADD, CONNECTCREATETAB ,CREATE_SECURE_OBJECT ,EXPLAIN,IMPLICIT_SCHEMA ,LOAD,SQLADM, WLMADM	DBA, DBA_RESTRICTED , DEVELOPER , ROLE_NULLID_PACKAGES , ROLE_PROCEDURES , ROLE_TABLESPACES
RDS per MariaDB	SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, RELOAD, PROCESS, REFERENCES , INDEX, ALTER, SHOW DATABASES , CREATE TEMPORARY TABLES, LOCK TABLES, EXECUTE, REPLICATION CLIENT , CREATE VIEW, SHOW VIEW, CREATE ROUTINE,	—

Motore del database	Privilegio del sistema	Ruolo di database
	`ALTER ROUTINE`, `CREATE USER`, `EVENT`, `TRIGGER`, `REPLICATION SLAVE`	
RDS per MySQL 8.0.36 e versioni successive	`SELECT`, `INSERT`, `UPDATE`, `DELETE`, `CREATE`, `DROP`, `RELOAD`, `PROCESS`, `REFERENCES` , `INDEX`, `ALTER`, `SHOW DATABASES` , `CREATE TEMPORARY TABLES`, `LOCK TABLES`, `EXECUTE`, `REPLICATION SLAVE`, `REPLICATION CLIENT` , `CREATE VIEW`, `SHOW VIEW`, `CREATE ROUTINE`, `ALTER ROUTINE`, `CREATE USER`, `EVENT`, `TRIGGER`, `CREATE ROLE`, `DROP ROLE`, `APPLICATION_PASSWORD_ADMIN` , `ROLE_ADMIN` , `SET_USER_ID` , `XA_RECOVER_ADMIN`	`rds_superuser_role` Per ulteriori informazioni su `rds_superuser_role` , consulta Privilegio basato sui ruoli.
Versioni RDS per MySQL precedenti alla 8.0.36	`SELECT`, `INSERT`, `UPDATE`, `DELETE`, `CREATE`, `DROP`, `RELOAD`, `PROCESS`, `REFERENCES` , `INDEX`, `ALTER`, `SHOW DATABASES` , `CREATE TEMPORARY TABLES`, `LOCK TABLES`, `EXECUTE`, `REPLICATION CLIENT` , `CREATE VIEW`, `SHOW VIEW`, `CREATE ROUTINE`, `ALTER ROUTINE`, `CREATE USER`, `EVENT`, `TRIGGER`, `REPLICATION SLAVE`	—

Motore del database	Privilegio del sistema	Ruolo di database
RDS per PostgreSQL.	`CREATE ROLE`, `CREATE DB`, `PASSWORD VALID UNTIL INFINITY`, `CREATE EXTENSION` , `ALTER EXTENSION` , `DROP EXTENSION` , `CREATE TABLESPACE` , `ALTER <OBJECT> OWNER`, `CHECKPOINT` , `PG_CANCEL_BACKEND( )` , `PG_TERMINATE_BACKEND()` , `SELECT PG_STAT_REPLICATION` , `EXECUTE PG_STAT_STATEMENTS_RESET()` , `OWN POSTGRES_FDW_HANDLER()` , `OWN POSTGRES_FDW_VALIDATOR()` , `OWN POSTGRES_FDW` , `EXECUTE PG_BUFFERCACHE_PAGES()` , `SELECT PG_BUFFERCACHE`	`RDS_SUPERUSER` Per ulteriori informazioni su RDS_SUPERUSER, consulta Informazioni su ruoli e autorizzazioni di PostgreSQL.
RDS per Oracle	`ADMINISTER DATABASE TRIGGER` , `ALTER DATABASE LINK`, `ALTER PUBLIC DATABASE LINK`, `AUDIT SYSTEM`, `CHANGE NOTIFICATION` , `DROP ANY DIRECTORY` , `EXEMPT ACCESS POLICY`, `EXEMPT IDENTITY POLICY`, `EXEMPT REDACTION POLICY`, `FLASHBACK ANY TABLE`, `GRANT ANY OBJECT PRIVILEGE` , `RESTRICTED SESSION` , `SELECT ANY TABLE`, `UNLIMITED TABLESPACE`	`DBA` **Note** Il `DBA` ruolo è esente dai seguenti privilegi: `ALTER DATABASE`, `ALTER SYSTEM`, `CREATE ANY DIRECTORY` , `CREATE EXTERNAL JOB`, `CREATE PLUGGABLE DATABASE`, `GRANT ANY PRIVILEGE` , `GRANT ANY ROLE`, `READ ANY FILE GROUP`

Motore del database	Privilegio del sistema	Ruolo di database
Amazon RDS for Microsoft SQL Server	`ADMINISTER BULK OPERATIONS` , `ALTER ANY CONNECTION` , `ALTER ANY CREDENTIAL` , `ALTER ANY EVENT SESSION`, `ALTER ANY LINKED SERVER`, `ALTER ANY LOGIN`, `ALTER ANY SERVER AUDIT`, `ALTER ANY SERVER ROLE`, `ALTER SERVER STATE`, `ALTER TRACE`, `CONNECT SQL`, `CREATE ANY DATABASE`, `VIEW ANY DATABASE`, `VIEW ANY DEFINITION` , `VIEW SERVER STATE`, `ALTER ON ROLE SQLAgentOperatorRole`	`DB_OWNER` (ruolo a livello di database), `PROCESSADMIN` (ruolo a livello di server), `SETUPADMIN` (ruolo a livello di server), `SQLAgentUserRole` (ruolo a livello di database)

Utilizzo di ruoli collegati ai servizi per Amazon RDS

Amazon RDS utilizza i ruoli collegati ai servizi AWS Identity and Access Management (IAM). Un ruolo collegato ai servizi è un tipo univoco di ruolo IAM collegato direttamente a Amazon RDS. I ruoli collegati ai servizi sono definiti automaticamente da Amazon RDS e includono tutte le autorizzazioni richieste dal servizio per eseguire chiamate agli altri servizi AWS per tuo conto.

Un ruolo collegato ai servizi semplifica l'uso di Amazon RDS perché non sarà più necessario aggiungere manualmente le autorizzazioni. Amazon RDS definisce le autorizzazioni dei ruoli collegati ai servizi e, salvo diversamente definito, solo Amazon RDS può assumere il ruolo. Le autorizzazioni definite includono la policy di trust e la policy delle autorizzazioni. Una policy delle autorizzazioni specifica non può essere collegata a un'altra entità IAM.

È possibile eliminare i ruoli solo dopo aver eliminato le risorse correlate. Questa procedura protegge le risorse di Amazon RDS perché impedisce la rimozione involontaria delle autorizzazioni di accesso alle risorse.

Per informazioni sugli altri servizi che supportano i ruoli collegati ai servizi, consultare i servizi AWS che funzionano con IAM e cercare i servizi che riportano Yes (Sì) nella colonna Service-Linked Role (Ruolo associato ai servizi). Scegli Yes (Sì) in corrispondenza di un link per visualizzare la documentazione relativa al ruolo collegato ai servizi per tale servizio.

Autorizzazioni del ruolo collegato ai servizi per Amazon RDS

Amazon RDS usa il ruolo collegato al servizio denominato AWSServiceRoleForRDS per consentire ad Amazon RDS di chiamare i servizi AWS per conto delle tue istanze di database.

Ai fini dell'assunzione del ruolo, il ruolo collegato ai servizi AWSServiceRoleForRDS considera attendibili i seguenti servizi:

- `rds.amazonaws.com`

A questo ruolo collegato ai servizi è collegata un policy di autorizzazione denominata `AmazonRDSServiceRolePolicy` che concede le autorizzazioni per operare nell'account. La policy delle autorizzazioni del ruolo consente ad Amazon RDS di eseguire le seguenti operazioni sulle risorse specificate:

Per ulteriori informazioni su questa policy, incluso il documento sulla policy JSON, consulta AmazonRDSServiceRolePolicy nella Guida di riferimento sulle policy gestite da AWS.

> **Note**
>
> Per consentire a un'entità IAM (ad esempio un utente, un gruppo o un ruolo) di creare, modificare o eliminare un ruolo collegato ai servizi, devi configurare le autorizzazioni. Se viene visualizzato il messaggio di errore seguente:
> Unable to create the resource. (Impossibile creare la risorsa. Verify that you have permission to create service linked role. (Verifica di possedere le autorizzazioni necessarie per creare un ruolo collegato ai servizi.) Otherwise wait and try again later. (In caso contrario, attendi e riprova più tardi.
> Accertati che le seguenti autorizzazioni siano abilitate:
>
> ```
> {
>     "Action": "iam:CreateServiceLinkedRole",
>     "Effect": "Allow",
>     "Resource": "arn:aws:iam::*:role/aws-service-role/rds.amazonaws.com/
> AWSServiceRoleForRDS",
>     "Condition": {
>         "StringLike": {
>             "iam:AWSServiceName":"rds.amazonaws.com"
>         }
>     }
> }
> ```
>
> Per ulteriori informazioni, consulta Autorizzazioni del ruolo collegato ai servizi nella Guida per l'utente di IAM.

Creazione di un ruolo collegato ai servizi per Amazon RDS

Non devi creare manualmente un ruolo collegato ai servizi. Quando si crea un'istanza database, Amazon RDS crea nuovamente il ruolo collegato al servizio per conto dell'utente.

> **Important**
>
> Se usavi il servizio Amazon RDS prima del 1 dicembre 2017, data da cui è disponibile il supporto dei ruoli collegati ai servizi, allora Amazon RDS ha creato il ruolo AWSServiceRoleForRDS nel tuo account. Per ulteriori informazioni, consulta Un nuovo ruolo appare nell'account AWS.

Se elimini questo ruolo collegato ai servizi e quindi devi ricrearlo di nuovo, puoi utilizzare lo stesso processo per ricreare il ruolo nel tuo account. Quando si crea un'istanza di database, Amazon RDS crea nuovamente il ruolo collegato al servizio per tuo conto.

Modifica di un ruolo collegato ai servizi per Amazon RDS

Amazon RDS non consente di modificare il ruolo collegato ai servizi AWSServiceRoleForRDS. Dopo aver creato un ruolo collegato ai servizi, non potrai modificarne il nome perché varie entità potrebbero farvi riferimento. È possibile tuttavia modificarne la descrizione utilizzando IAM. Per ulteriori informazioni, consulta Modifica di un ruolo collegato ai servizi nella Guida per l'utente di IAM.

Eliminazione di un ruolo collegato ai servizi per Amazon RDS

Se non è più necessario utilizzare una caratteristica o un servizio che richiede un ruolo collegato ai servizi, ti consigliamo di eliminare quel ruolo. In questo modo non sarà più presente un'entità non utilizzata che non viene monitorata e gestita attivamente. Tuttavia, prima di poter eliminare il ruolo collegato al servizio, devi eliminare tutte le istanze database.

Pulizia di un ruolo collegato ai servizi

Prima di utilizzare IAM per eliminare un ruolo collegato ai servizi, devi innanzitutto verificare che il ruolo non abbia sessioni attive ed eliminare tutte le risorse utilizzate dal ruolo.

Per verificare se il ruolo collegato ai servizi dispone di una sessione attiva nella console IAM

1. Accedi alla AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione della console IAM seleziona Roles (Ruoli). Quindi, scegli il nome (non la casella di controllo) del ruolo AWSServiceRoleForRDS.
3. Nella pagina Summary (Riepilogo) per il ruolo selezionato, scegli la scheda Access Advisor (Consulente accessi).
4. Nella scheda Access Advisor (Consulente accessi), esamina l'attività recente per il ruolo collegato ai servizi.

> **Note**
>
> Se non si ha la certezza che Amazon RDS stia utilizzando il ruolo AWSServiceRoleForRDS, è possibile provare a eliminarlo. Se il servizio sta utilizzando il ruolo, l'eliminazione non andrà a buon fine e potrai visualizzare le regioni AWS in cui

il ruolo viene utilizzato. Se il ruolo è in uso, prima di poterlo eliminare dovrai attendere il termine della sessione. Non puoi revocare la sessione per un ruolo collegato ai servizi.

Se desideri rimuovere il ruolo AWSServiceRoleForRDS, devi prima eliminare tutti gli oggetti di tipo istanza database.

Eliminazione di tutte le istanze

Utilizza una di queste procedure per eliminare ogni istanza.

Per eliminare un'istanza (console)

1. Apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, scegliere Databases (Database).
3. Scegliere l'istanza da eliminare.
4. In Actions (Azioni), selezionare Delete (Elimina).
5. Se viene visualizzato il messaggio Create final Snapshot? (Creare snapshot finale?), scegliere Yes (Sì) o No.
6. Se si sceglie Yes (Sì) nella fase precedente, in Final snapshot name (Nome snapshot finale) immettere il nome dell'ultimo snapshot.
7. Scegliere Delete (Elimina).

Per eliminare un'istanza (CLI)

Consulta `delete-db-instance` in Riferimento ai comandi AWS CLI.

Per eliminare un'istanza (API)

Consulta `DeleteDBInstance` nella Amazon RDS API Reference.

Per eliminare il ruolo collegato al servizio AWSServiceRoleForRDS, puoi usare la console IAM, la CLI IAM o l'API IAM. Per ulteriori dettagli, consulta Eliminazione di un ruolo collegato al servizio nella Guida per l'utente di IAM.

Autorizzazioni del ruolo collegato ai servizi per Amazon RDS Custom

Amazon RDS Custom usa il ruolo collegato al servizio denominato `AWSServiceRoleForRDSCustom` per consentire a RDS Custom di chiamare i servizi AWS per conto delle istanze e dei cluster database.

Ai fini dell'assunzione del ruolo, il ruolo collegato ai servizi AWSServiceRoleForRDSCustom considera attendibili i seguenti servizi:

- `custom.rds.amazonaws.com`

A questo ruolo collegato ai servizi è collegata un policy di autorizzazione denominata `AmazonRDSCustomServiceRolePolicy` che concede le autorizzazioni per operare nell'account. La policy delle autorizzazioni del ruolo consente ad di eseguire le seguenti operazioni sulle risorse specificate:

Per ulteriori informazioni su questa policy, incluso il documento sulla policy JSON, consulta AmazonRDSCustomServiceRolePolicy nella Guida di riferimento sulle policy gestite da AWS.

La creazione, la modifica o l'eliminazione del ruolo collegato al servizio per RDS Custom funziona come per Amazon RDS. Per ulteriori informazioni, consulta Autorizzazioni del ruolo collegato ai servizi per Amazon RDS.

> **Note**
>
> Per consentire a un'entità IAM (ad esempio un utente, un gruppo o un ruolo) di creare, modificare o eliminare un ruolo collegato ai servizi, devi configurare le autorizzazioni. Se viene visualizzato il messaggio di errore seguente:
> Unable to create the resource. (Impossibile creare la risorsa. Verify that you have permission to create service linked role. (Verifica di possedere le autorizzazioni necessarie per creare un ruolo collegato ai servizi.) Otherwise wait and try again later. (In caso contrario, attendi e riprova più tardi.
> Accertati che le seguenti autorizzazioni siano abilitate:
>
> ```
> {
>     "Action": "iam:CreateServiceLinkedRole",
>     "Effect": "Allow",
>     "Resource": "arn:aws:iam::*:role/aws-service-role/custom.rds.amazonaws.com/
> AmazonRDSCustomServiceRolePolicy",
> ```

```
    "Condition": {
        "StringLike": {
            "iam:AWSServiceName":"custom.rds.amazonaws.com"
        }
    }
}
```

Per ulteriori informazioni, consulta Autorizzazioni del ruolo collegato ai servizi nella Guida per l'utente di IAM.

VPC di Amazon VPC e Amazon RDS

Amazon Virtual Private Cloud (Amazon VPC) consente di avviare le risorse AWS, come le istanze database Amazon RDS, in un cloud privato virtuale (VPC).

Quando utilizzi un VPC, hai il controllo completo sull'ambiente virtuale di rete. Puoi scegliere il tuo intervallo di indirizzi IP, creare sottoreti e configurare liste di routing e di controllo accessi. Non è previsto alcun costo aggiuntivo per eseguire l'istanza database in Amazon VPC.

Gli account hanno un VPC predefinito. Tutte le nuove istanze database vengono create nel VPC predefinito, salvo diversamente specificato.

Argomenti

Di seguito vengono descritte le funzionalità VPC relative alle istanze database Amazon RDS. Per ulteriori informazioni su Amazon VPC, consulta Guida alle operazioni di base di Amazon VPC e Guida per l'utente di Amazon VPC.

Uso di un'istanza database in un VPC

L'istanza database deve essere all'interno di un cloud privato virtuale (VPC). Un VPC è una rete virtuale isolata a livello logico da altre reti virtuali in AWS Cloud. Amazon VPC ti consente di avviare le risorse AWS, ad esempio un'istanza database Amazon RDS o un'istanza Amazon EC2, in un VPC. Il VPC può essere un VPC predefinito fornito con l'account o uno creato da te. Tutti i VPC sono associati all'account AWS.

Il VPC predefinito ha tre sottoreti che è possibile usare per isolare le risorse all'interno del VPC. Il VPC predefinito ha anche un gateway Internet che può essere usato per fornire l'accesso alle risorse all'interno del VPC dall'esterno del VPC.

Per un elenco di scenari relativi alle istanze database Amazon RDS in un VPC e al suo esterno, consulta Scenari per accedere a un'istanza database in un VPC.

Argomenti

Nei seguenti tutorial puoi imparare a creare un VPC da usare per uno scenario Amazon RDS comune:

Uso di un'istanza database in un VPC

Ecco alcuni consigli per l'uso di un'istanza database in un VPC:

- Il VPC deve avere almeno due sottoreti. Queste sottoreti devono trovarsi in due zone di disponibilità diverse nella Regione AWS in cui si desidera implementare l'istanza database. Una sottorete è un segmento dell'intervallo di indirizzi IP di un VPC che è possibile specificare e utilizzare per raggruppare le istanze database in base alle esigenze operative e di sicurezza.

  Per le implementazioni Multi-AZ, la definizione di una sottorete per due o più zone di disponibilità in una Regione AWS consente ad Amazon RDS di creare, se necessario, un nuovo standby in un'altra zona di disponibilità. Assicurati di effettuare questa operazione anche per le implementazioni Single-AZ, in modo da poterle eventualmente convertire in implementazioni Multi-AZ in un secondo momento.

  > **Note**
  >
  > Il gruppo di sottorete DB per un'area locale può avere solo una sottorete.

- Se desideri che l'istanza database nel VPC sia accessibile a livello pubblico, verifica di aver attivato gli attributi VPC DNS hostnames (Nomi host DNS) e DNS resolution (Risoluzione DNS).

- Il VPC deve includere un gruppo di sottoreti database creato da te. Puoi creare un gruppo di sottoreti di database, specificando le sottoreti create. Amazon RDS sceglie una sottorete e un indirizzo IP all'interno di quel gruppo di sottoreti da associare all'istanza database. L'istanza database utilizza la zona di disponibilità che contiene la sottorete.
- Il VPC deve avere un gruppo di sicurezza VPC che consente l'accesso all'istanza database.

  Per ulteriori informazioni, consulta Scenari per accedere a un'istanza database in un VPC.
- I blocchi CIDR in ciascuna delle sottoreti devono essere sufficientemente grandi da contenere gli indirizzi IP di riserva per Amazon RDS da utilizzare durante le attività di manutenzione, inclusi il failover e il dimensionamento delle risorse di calcolo. Ad esempio, un intervallo come 10.0.0.0/24 e 10.0.1.0/24 è in genere sufficientemente ampio.
- Un VPC può avere un attributo di tenancy di istanza o predefinito o dedicato. Tutti i VPC predefiniti hanno l'attributo di tenancy di istanza impostato su predefinito e un VPC predefinito può supportare qualsiasi classe di istanza database.

  Se scegli di includere l'istanza database in un VPC dedicato in cui l'attributo di locazione dell'istanza è impostato su dedicato, la classe dell'istanza database dell'istanza database deve essere uno dei tipi di istanza dedicata Amazon EC2. Ad esempio, l'istanza dedicata EC2 r5.large corrisponde alla classe di istanza database db.r5.large. Per informazioni sulla tenancy di istanza in un VPC, consulta Istanze dedicate nella Guida per l'utente Amazon Elastic Compute Cloud.

  Per ulteriori informazioni sui tipi di istanze che possono essere in un'istanza dedicata, consulta Istanze dedicate Amazon EC2 nella pagina dei prezzi EC2.

  > **Note**
  >
  > Quando si imposta l'attributo di locazione dell'istanza su dedicato per un'istanza database, non è garantita l'esecuzione dell'istanza database su un host dedicato.

- Quando un gruppo di opzioni è assegnato a un'istanza database, tale gruppo è associato al VPC dell'istanza database. Questo collegamento significa che non puoi utilizzare il gruppo di opzioni assegnato a un'istanza database se tenti di ripristinare l'istanza database in un diverso VPC.
- Se ripristini un'istanza database in un VPC diverso, assicurati di assegnare il gruppo di opzioni predefinito all'istanza database, assegnare un gruppo di opzioni che sia collegato a quel VPC oppure creare un nuovo gruppo di opzioni e assegnarlo all'istanza database. Tieni presente che con le opzioni permanenti, come Oracle TDE, quando ripristini un'istanza database in un VPC diverso devi creare un nuovo gruppo di opzioni che includa l'opzione permanente.

Utilizzo di gruppi di sottoreti database

Le sottoreti sono segmenti di un intervallo di indirizzi IP di un VPC che si designa per raggruppare le risorse in base alle esigenze operative e di sicurezza. Un gruppo di sottoreti DB è una raccolta di sottoreti (generalmente private) creata in un VPC e che è possibile indicare per le istanze database. Un gruppo di sottoreti DB ti consente di specificare un determinato VPC quando crei istanze database usando la AWS CLI oppure l'API RDS. Se utilizzi la console, puoi scegliere il VPC e i gruppi di sottorete che desideri usare.

Ogni gruppo di sottoreti database deve avere almeno due zone di disponibilità in una determinata Regione AWS. Quando si crea un'istanza database in un VPC, è necessario selezionare anche un gruppo di sottoreti DB. Dal gruppo di sottoreti DB, Amazon RDS sceglie una sottorete e un indirizzo IP all'interno di essa da associare istanza database. Il database utilizza la zona di disponibilità che contiene la sottorete.

Se l'istanza database primaria di un'implementazione Multi-AZ ha esito negativo, Amazon RDS può promuovere l'istanza di standby corrispondente e successivamente creare una nuova istanza di standby utilizzando un indirizzo IP della sottorete in una delle altre zone di disponibilità.

Le sottoreti di un gruppo di sottoreti DB sono pubbliche o private. Le sottoreti sono pubbliche o private, a seconda della configurazione impostata per gli elenchi di controllo dell'accesso alla rete (ACL) e le tabelle di routing. Affinché un'istanza database possa essere accessibile a livello pubblico, tutte le sottoreti nel gruppo di sottoreti database devono essere pubbliche. Se una sottorete associata a un'istanza database accessibile pubblicamente cambia da pubblica a privata, ciò può avere ripercussioni sulla disponibilità dell'istanza database.

Per creare un gruppo di sottoreti DB che supporti la modalità dual-stack, assicurati che a ogni sottorete aggiunta al gruppo sia associata un blocco CIDR Internet Protocol versione 6 (IPv6). Per ulteriori informazioni, consulta Assegnazione di indirizzi IP in Amazon RDS e Migrazione a IPv6 nella Guida per l'utente di Amazon VPC.

> **Note**
>
> Il gruppo di sottorete DB per un'area locale può avere solo una sottorete.

Quando Amazon RDS crea un'istanza database in un VPC, assegna un'interfaccia di rete all'istanza database usando un indirizzo IP dal gruppo di sottoreti del database. Tuttavia, consigliamo vivamente

di usare il nome del sistema dei nomi di dominio (DNS) per collegare l'istanza database, perché l'indirizzo IP sottostante varia durante un failover.

> **Note**
>
> Per ogni istanza database in esecuzione in un VPC, assicurati di riservare almeno un indirizzo in ogni sottorete nel gruppo di sottoreti DB per l'utilizzo da parte di Amazon RDS per le operazioni di ripristino.

Sottoreti condivise

Puoi creare un'istanza database in un VPC condiviso.

Alcune considerazioni da tenere presente durante l'utilizzo di VPC condivisi:

- È possibile spostare un'istanza database da una sottorete VPC condivisa a una sottorete VPC non condivisa e viceversa.
- I membri di un VPC condiviso devono creare un gruppo di sicurezza nel VPC per consentire loro di creare un'istanza database.
- I proprietari e i membri di un VPC condiviso possono accedere al database utilizzando le query SQL. Tuttavia, solo il creatore di una risorsa può effettuare chiamate API sulla risorsa.

Assegnazione di indirizzi IP in Amazon RDS

Gli indirizzi IP permettono alle risorse nel VPC di comunicare tra loro e con le risorse su Internet. Amazon RDS supporta entrambi i protocolli di indirizzamento, IPv4 e IPv6. Per impostazione di default, Amazon RDS e Amazon VPC utilizzano il protocollo di indirizzamento IPv4. Non puoi disattivare questo comportamento. Quando crei un VPC, assicurati di specificare un blocco CIDR IPv4 (un intervallo di indirizzi IPv4 privati). Puoi scegliere di assegnare un blocco CIDR IPv6 al VPC e alle sottoreti e di assegnare gli indirizzi IPv6 di tale blocco a instanze database presenti nella sottorete.

Il supporto del protocollo IPv6 espande il numero di indirizzi IP supportati. L'utilizzo del protocollo IPv6 consente di disporre di un numero sufficiente di indirizzi per adeguarsi alla futura espansione di Internet. Le risorse RDS nuove ed esistenti possono utilizzare indirizzi IPv4 e IPv6 all'interno del VPC. La configurazione, la protezione e la traduzione del traffico di rete tra i due protocolli utilizzati in

diverse parti di un'applicazione può causare sovraccarico operativo. Puoi standardizzare il protocollo IPv6 per le risorse Amazon RDS per semplificare la configurazione di rete.

Argomenti

Indirizzi IPv4

Quando crei un VPC, devi specificare un intervallo di indirizzi IPv4 per il VPC sotto forma di un blocco (CIDR), ad esempio `10.0.0.0/16`. Un gruppo di sottoreti DB definisce l'intervallo di indirizzi IP in questo blocco CIDR che può essere usato da istanze database. Questi indirizzi IP possono essere privati o pubblici.

Un indirizzo IPv4 privato è un indirizzo IP non raggiungibile tramite Internet. Puoi utilizzare indirizzi IPv4 privati per la comunicazione tra istanze database e altre risorse, ad esempio istanze Amazon EC2, nello stesso VPC. Ogni istanza database dispone di un indirizzo IP privato per la comunicazione nel VPC.

Un indirizzo IP pubblico è un indirizzo IPv4 raggiungibile tramite Internet. Puoi utilizzare gli indirizzi pubblici per la comunicazione tra istanze database e risorse su Internet, ad esempio un client SQL. Puoi controllare se instanze database ricevono un indirizzo IP pubblico.

Per un tutorial che mostra come creare un VPC solo con indirizzi IPv4 privati da usare con uno scenario Amazon RDS comune, consulta Tutorial: Creazione di un Amazon VPC da utilizzare con un'istanza database (solo IPv4).

Indirizzi IPv6

Puoi anche scegliere di associare un blocco CIDR IPv6 al VPC e alle sottoreti e di assegnare gli indirizzi IPv6 del blocco alle risorse presenti nel VPC. Ogni indirizzo IPv6 è univoco a livello globale.

Il blocco CIDR IPv6 per il VPC è assegnato automaticamente dal pool di indirizzi IPv6 di Amazon. Non è possibile scegliere l'intervallo in modo autonomo.

Quando ti connetti a un indirizzo IPv6, assicurati che siano soddisfatte le seguenti condizioni:

- Il client è configurato in modo che sia consentito il traffico tra client e database su IPv6.

- I gruppi di sicurezza RDS utilizzati dall'istanza database sono configurati correttamente in modo che sia consentito il traffico tra client e database su IPv6.
- Lo stack del sistema operativo client consente il traffico sull'indirizzo IPv6 e i driver e le librerie del sistema operativo sono configurati per scegliere l'endpoint dell'istanza database di default corretto (IPv4 o IPv6).

Per ulteriori informazioni su IPv6, consulta l'argomento relativo all'assegnazione di indirizzi IP nella Guida per l'utente di Amazon VPC.

Modalità dual-stack

Quando istanze database possono comunicare mediante entrambi i protocolli di indirizzamento IPv4 e IPv6, significa che sono in esecuzione in modalità dual-stack. Pertanto, le risorse possono comunicare con istanze database su IPv4, IPv6 o entrambi. RDS disabilita l'accesso al gateway Internet per gli endpoint IPv6 di istanze database private in modalità dual-stack per garantire che gli endpoint IPv6 siano privati e accessibili solo dall'interno del VPC.

Argomenti

- Modalità dual-stack e gruppi di sottoreti database
- Utilizzo di istanze database in modalità dual-stack
- Modifica delle istanze database solo IPv4 per l'utilizzo della modalità dual-stack
- Disponibilità di regioni e versioni
- Limitazioni per istanze database di rete dual-stack

Per un tutorial che mostra come creare un VPC con indirizzi IPv4 e IP6 da usare con uno scenario Amazon RDS comune, consulta Tutorial: Creazione di un VPC per l'utilizzo con un'istanza database (modalità dual-stack).

Modalità dual-stack e gruppi di sottoreti database

Per utilizzare la modalità dual-stack, assicurati che ogni sottorete nel gruppo di sottoreti database associato a istanze database sia associata a un blocco CIDR IPv6. Per soddisfare questo requisito, puoi creare un nuovo gruppo di sottoreti database o modificare un gruppo di sottoreti database esistente. Quando istanze database sono in modalità dual-stack, i client possono connettersi normalmente. Assicurati che i firewall di sicurezza client e i gruppi di sicurezza delle istanze database RDS siano configurati in modo accurato per consentire il traffico su IPv6. Per connettersi, i client utilizzano l'endpoint dell'istanza database. Le applicazioni client possono specificare il protocollo

preferito per la connessione a un database. In modalità dual-stack, l'istanza database rileva il protocollo di rete preferito dal client (IPv4 o IPv6) e utilizza tale protocollo per la connessione.

Se un gruppo di sottoreti database smette di supportare la modalità dual-stack a causa dell'eliminazione della sottorete o dell'annullamento dell'associazione con il blocco CIDR, si può verificare una situazione di stato di rete incompatibile per le istanze database associate al gruppo di sottoreti database. Inoltre, non puoi utilizzare il gruppo di sottoreti database quando crei una nuova istanza database in modalità dual-stack.

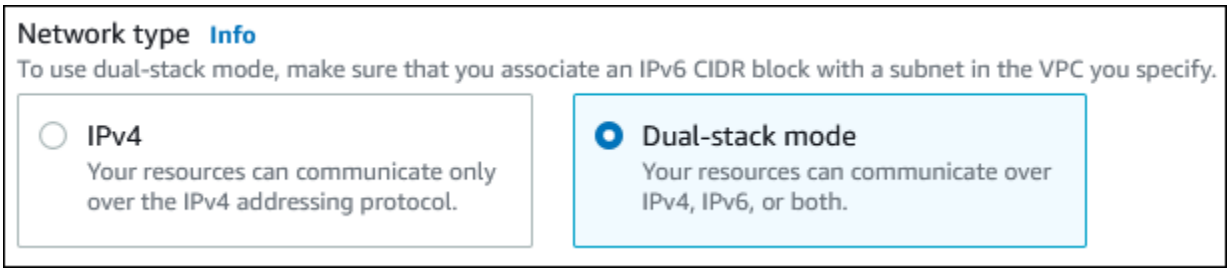
Per determinare se un gruppo di sottoreti database supporta la modalità dual-stack utilizzando la AWS Management Console, visualizzare il tipo di rete nella pagina dei dettagli del gruppo di sottoreti database. Per determinare se un gruppo di sottoreti DB supporta la modalità dual-stack utilizzando ilAWS CLI, esegui il describe-db-subnet-groupscomando e visualizza `SupportedNetworkTypes` nell'output.

Le repliche di lettura vengono trattate come istanze database indipendenti e possono avere un tipo di rete diverso dall'istanza database principale. Se si modifica il tipo di rete dell'istanza database principale di una replica di lettura, tale modifica non interessa la replica di lettura. Quando si ripristina un'istanza database, è possibile ripristinarla su qualsiasi tipo di rete supportato.

Utilizzo di istanze database in modalità dual-stack

Quando crei o modifichi un'istanza database, puoi specificare la modalità dual-stack per consentire alle risorse di comunicare con l'istanza su IPv4, IPv6 o entrambi.

Quando utilizzi la AWS Management Console per creare o modificare un'istanza database, è possibile specificare la modalità dual-stack nella sezione Tipo di rete. L'immagine seguente mostra la sezione Network type (Tipo di rete) nella console.



Quando utilizzi la AWS CLI per creare o modificare un'istanza database, per utilizzare la modalità dual-stack imposta l'ozpione `--network-type` su `DUAL`. Quando usi l'API RDS per creare o modificare un'istanza database, per utilizzare la modalità dual-stack imposta il parametro `NetworkType` su `DUAL`. Quando modifichi il tipo di rete di un'istanza database, è possibile che si verifichi un periodo di inattività. Se la modalità dual-stack non è supportata dalla versione

del motore del database o dal gruppo di sottoreti database in uso, viene restituito l'errore `NetworkTypeNotSupported`.

Per ulteriori informazioni sulla creazione di un'istanza database, consulta Creazione di un'istanza database Amazon RDS. Per ulteriori informazioni sulla modifica di un'istanza database, consulta Modifica di un'istanza database Amazon RDS.

Per determinare se un'istanza database è in modalità dual-stack utilizzando la console, visualizza l'opzione Network type (Tipo di rete) nella scheda Connectivity & security (Connettività e sicurezza) per l'istanza database in questione.

Modifica delle istanze database solo IPv4 per l'utilizzo della modalità dual-stack

È possibile modificare un'istanza database solo IPv4 per utilizzare la modalità dual-stack. A tale scopo, devi modificare il tipo di rete dell'istanza database. La modifica potrebbe comportare tempi di inattività.

Ti consigliamo di modificare il tipo di rete delle istanze database Amazon RDS durante una finestra di manutenzione. L'impostazione predefinita del tipo di rete delle nuove istanze sulla modalità dual stack non è al momento supportata. È possibile impostare il tipo di rete manualmente utilizzando il comando `modify-db-instance` .

Prima di modificare un'istanza database per utilizzare la modalità dual-stack, assicurati che il gruppo di sottoreti DB supporti la modalità dual-stack. Se il gruppo di sottoreti DB associato all'istanza database non supporta la modalità dual-stack, specifica un gruppo di sottoreti database diverso che supporta tale modalità quando modifichi l'istanza database. La modifica del gruppo di sottoreti di database di un'istanza database può causare tempi di inattività.

Se si modifica il gruppo di sottoreti di database di un'istanza database prima di modificare l'istanza database per l'utilizzo della modalità Dual stack, assicurati che il gruppo di sottoreti di database sia valido per l'istanza database prima e dopo la modifica.

Per RDS per PostgreSQL, RDS per MySQL, RDS per Oracle e RDS per le istanze Single-AZ di MariadB, si consiglia di eseguire il comando con il solo parametro impostato per modificare la rete in modalità dual-stack. modify-db-instance`--network-typeDUAL` L'aggiunta di altri parametri al parametro `--network-type` nella stessa chiamata API potrebbe causare tempi di inattività. Per modificare più parametri, assicurati che la modifica del tipo di rete sia stata completata correttamente prima di inviare un'altra richiesta `modify-db-instance` con altri parametri.

Le modifiche al tipo di rete per le istanze DB RDS per PostgreSQL, RDS per MySQL, RDS per Oracle e RDS per MariaDB Multi-AZ causano un breve periodo di inattività e attivano un failover se si

utilizza il parametro solo o se si combinano i parametri in un comando. `--network-type` modify-db-instance

Le modifiche del tipo di rete per le istanze database Single-AZ o Multi-AZ RDS per SQL Server causano tempi di inattività se si utilizza solo il parametro `--network-type` o se si combinano più parametri in un comando `modify-db-instance`. Le modifiche del tipo di rete causano il failover in un'istanza Multi-AZ SQL Server.

Se non riesci a connetterti all'istanza database dopo la modifica, assicurati che i firewall di sicurezza e le tabelle di routing client e database siano configurati in modo accurato per consentire il traffico verso il database sulla rete selezionata (IPv4 o IPv6). Potrebbe anche essere necessario modificare i parametri del sistema operativo, le librerie o i driver per connettersi mediante un indirizzo IPv6.

Quando modifichi un'istanza database per usare la modalità dual-stack non possono essere presenti modifiche in sospeso da implementazioni Single-AZ a implementazioni multi-AZ o viceversa.

Per modificare un'istanza database solo IPv4 per utilizzare la modalità dual-stack

1. Modificare un gruppo di sottoreti database per supportare la modalità dual-stack o creare un gruppo di sottoreti database che supporti la modalità dual-stack:

   a. Associazione di un blocco CIDR IPv6 al VPC.

      Per ulteriori informazioni, consulta Come aggiungere un blocco CIDR IPv6 al VPC nella Guida per l'utente di Amazon VPC.

   b. Allegare il blocco CIDR IPv6 a tutte le sottoreti del gruppo di sottoreti database.

      Per ulteriori informazioni, consulta Come aggiungere un blocco CIDR IPv6 alla sottorete nella Guida per l'utente di Amazon VPC.

   c. Verificare che il gruppo di sottoreti database supporti la modalità dual-stack.

      In caso di utilizzo della AWS Management Console, selezionare il gruppo di sottoreti database e assicurarsi che il valore dell'opzione Supported network types (Tipi di rete supportati) sia Dual-IPv4.

      Se utilizzi il, esegui il comando e assicurati che il valore per l'istanza DB siaAWS CLI. describe-db-subnet-groups`SupportedNetworkTypeDual, IPv4`

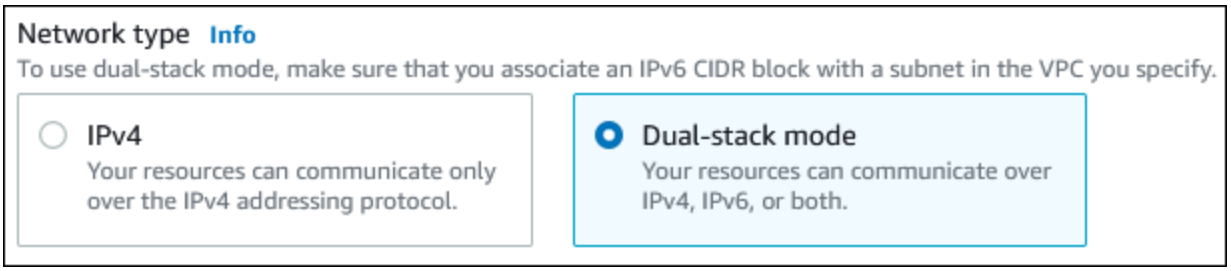
2. Modifica il gruppo di sicurezza associato all'istanza database per consentire le connessioni IPv6 al database o creare un nuovo gruppo di sicurezza che consenta le connessioni IPv6.

Per le istruzioni, vedere Regole del gruppo di sicurezza nella Guida per l'utente di Amazon VPC.

3. Modifica l'istanza database in modo che supporti la modalità dual-stack. A questo scopo, imposta l'opzione Network type (Tipo di rete) su Dual-stack mode (Modalità dual-stack).

   In caso di utilizzo della console, accertati che le impostazioni seguenti siano corrette:

   - Tipo di rete–Dual-stack mode (Modalità dual-stack)

   - DB subnet group (Gruppo di sottoreti DB): gruppo di sottoreti database configurato in un passaggio precedente
   - Security group (Gruppo di sicurezza): gruppo di sicurezza di default configurato in un passaggio precedente

   In caso di utilizzo della AWS CLI, accertarsi che le impostazioni seguenti siano corrette:

   - `--network-type` – `dual`
   - `--db-subnet-group-name`: gruppo di sottoreti database configurato in un passaggio precedente
   - `--vpc-security-group-ids`: gruppo di sicurezza VPC configurato in un passaggio precedente

   Per esempio:

   ```
   aws rds modify-db-instance --db-instance-identifier my-instance --network-type
    "DUAL"
   ```

4. Verifica che l'istanza database supporti la modalità dual-stack.

   Se utilizzi la console, scegli la scheda Connectivity & security (Connettività e sicurezza) per l'istanza database. In quella scheda, assicurati che il valore dell'opzione Network type (Tipo di rete) sia Dual-stack mode (Modalità dual-stack).

Se utilizzi ilAWS CLI, esegui il describe-db-instancescomando e assicurati che il `NetworkType` valore per l'istanza DB sia`dual`.

Esegui il comando `dig` sull'endpoint dell'istanza database per individuare l'indirizzo IPv6 associato.

```
dig db-instance-endpoint AAAA
```

Utilizza l'endpoint dell'istanza database, non l'indirizzo IPv6, per connetterti all'istanza database.

Disponibilità di regioni e versioni

Il supporto varia a seconda delle versioni specifiche di ciascun motore di database e a seconda delle Regioni AWS. Per ulteriori informazioni sulla disponibilità di versioni e regioni con la modalità dual-stack, consulta Regioni e motori DB supportati per la modalità dual-stack in Amazon RDS.

Limitazioni per istanze database di rete dual-stack

Le seguenti limitazioni si applicano alle istanze database di rete dual-stack:

- Le istanze database non possono utilizzare esclusivamente il protocollo IPv6. Possono utilizzare esclusivamente il protocollo IPv4 oppure i protocolli IPv4 e IPv6 (modalità dual-stack).
- Amazon RDS non supporta le sottoreti IPv6 native.
- Le istanze database che utilizzano la modalità dual-stack devono essere di tipo privato. Non possono essere accessibili pubblicamente.
- La modalità dual-stack non supporta le istanza database di classe db.m3 e db.r3.
- Per RDS per SQL Server, le istanze database in modalità dual-stack che utilizzano gli endpoint del listener del gruppo di disponibilità Always On usano solo indirizzi IPv4.
- Non è possibile utilizzare il proxy RDS con istanze database in modalità dual-stack.
- Non è possibile utilizzare la modalità dual-stack con RDS su istanze database AWS Outposts.
- Non è possibile utilizzare la modalità dual-stack con istanze database in una zona locale.

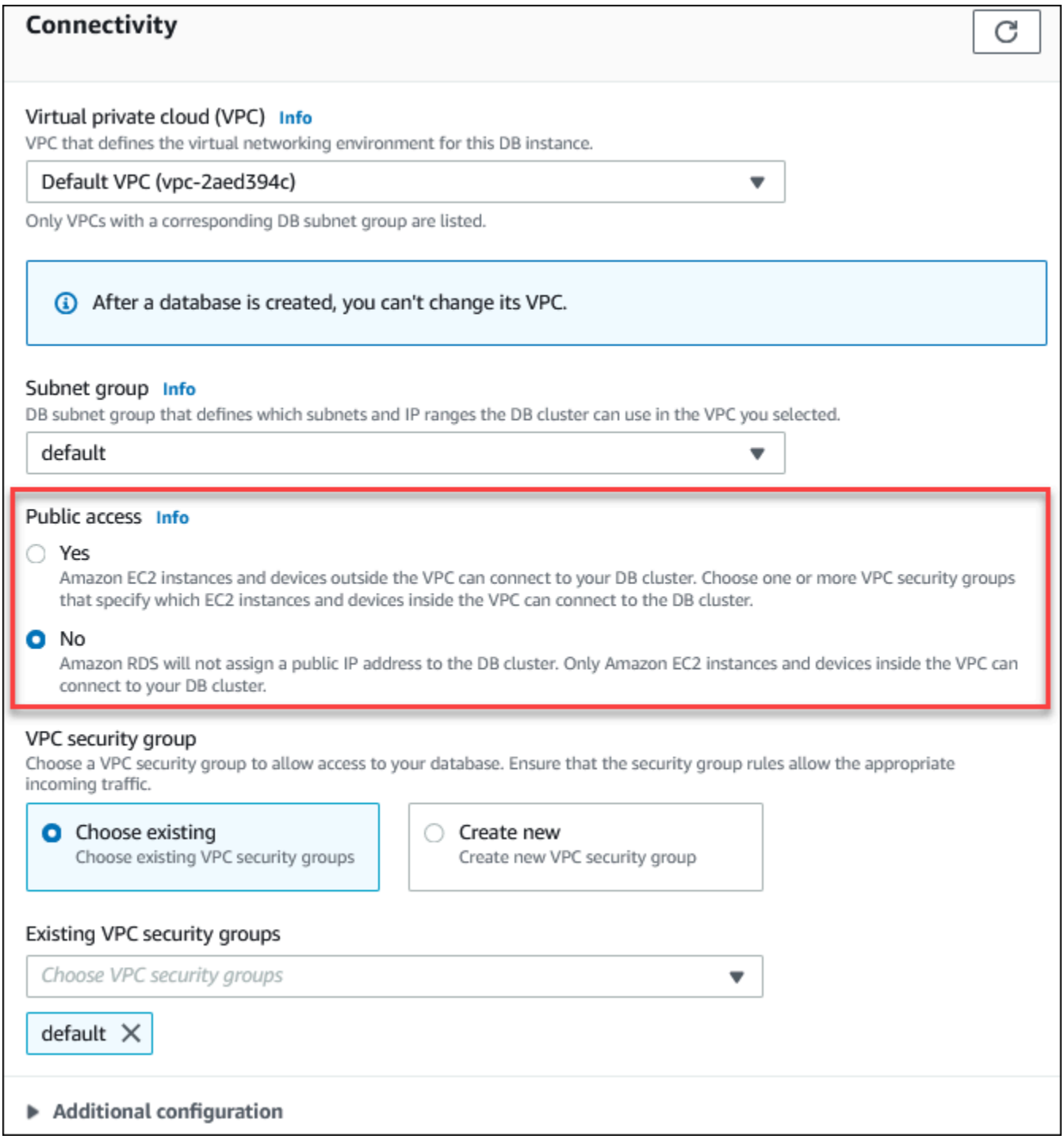
Nascondere istanze database in un VPC da Internet

Uno scenario Amazon RDS comune è quello di avere un VPC in cui disponi di un'istanza EC2 con un'applicazione Web pubblica e un'istanza database con un database che non è pubblicamente

accessibile. Puoi ad esempio creare un VPC che ha una sottorete pubblica e una sottorete privata. Le istanze Amazon EC2 che fungono da server Web possono essere implementate nella sottorete pubblica. L'implementazione di istanze database viene invece eseguita nella sottorete privata. In tale implementazione, solo i server Web hanno accesso alle istanze database. Per un'illustrazione di questo scenario, consulta Un'istanza database in un VPC a cui accede un'istanza EC2 nello stesso VPC.

Quando si avvia un'istanza database all'interno di un VPC, l'istanza database dispone di un indirizzo IP privato per il traffico all'interno del VPC. Questo indirizzo IP privato non è accessibile pubblicamente. Puoi utilizzare l'opzione Public access (Accesso pubblico) per indicare se l'istanza database dispone anche di un indirizzo IP pubblico oltre all'indirizzo IP privato. Se l'istanza database è definito come accessibile pubblicamente, il relativo endpoint DNS utilizza l'indirizzo IP privato dall'interno del VPC. Utilizza invece l'indirizzo IP pubblico dall'esterno del VPC. L'accesso all'istanza database è in ultima analisi controllato dal gruppo di sicurezza in uso. Questo accesso pubblico non è consentito se il gruppo di sicurezza assegnato all'istanza database non include regole in entrata che lo consentono. Per un'istanza database che deve essere accessibile pubblicamente, le sottoreti nel relativo gruppo di sottoreti DB devono disporre di un gateway Internet. Per ulteriori informazioni, consulta Impossibile connettersi all'istanza database di Amazon RDS

Puoi modificare un'istanza database per attivare o disattivare l'accessibilità pubblica modificando l'opzione Public access (Accesso pubblico). Nella figura seguente viene illustrata l'opzione Public access (Accesso pubblico) nella sezione Additional connectivity configuration (Configurazioni di connettività aggiuntiva) . Per impostare l'opzione, apri la sezione Additional connectivity configuration (Configurazioni di connettività aggiuntiva) nella sezione Connectivity (Connettività) .



Per informazioni sulla modifica di un'istanza database per impostare l'opzione Public access (Accesso pubblico), consulta Modifica di un'istanza database Amazon RDS.

Creazione di un'istanza database in un VPC

Le procedure seguenti aiutano a creare un'istanza database in un VPC. Per utilizzare il VPC predefinito, puoi iniziare con il passaggio 2 e utilizzare il VPC e il gruppo di sottoreti DB creati automaticamente. Se desideri creare un VPC aggiuntivo, puoi creare un nuovo VPC.

> **Note**
>
> Se desideri che l'istanza database nel VPC sia pubblicamente accessibile, devi aggiornare le informazioni DNS per il VPC attivando gli attributi VPC DNS hostnames (Nomi host DNS) e DNS resolution (Risoluzione DNS). Per informazioni sull'aggiornamento delle informazioni DNS per un'istanza VPC, consulta Aggiornamento del supporto DNS per il VPC.

Segui questa procedura per creare un'istanza database in un VPC:

- Fase 1. Creazione di un VPC
- Fase 2: creazione di un gruppo di sottoreti database
- Fase 3: creazione di un gruppo di sicurezza VPC
- Passaggio 4: creazione di un'istanza database nel VPC

Fase 1. Creazione di un VPC

Crea un VPC con sottoreti in almeno due zone di disponibilità. Usi queste sottoreti quando crei un gruppo di sottoreti database. Se disponi di un VPC di default, viene creata automaticamente una sottorete in ciascuna zona di disponibilità nella Regione AWS.

Per ulteriori informazioni, consulta Creazione di un VPC con sottoreti pubbliche e private oppure Creazione di un VPC nella Guida per l'utente di Amazon VPC..

Fase 2: creazione di un gruppo di sottoreti database

Un gruppo di sottoreti DB è una raccolta di sottoreti (generalmente private) creata per un VPC e che è possibile definire per le istanze database. Un gruppo di sottoreti DB ti consente di specificare un determinato VPC quando crei istanze database usando la AWS CLI oppure l'API RDS. Se utilizzi la console, puoi scegliere il VPC e le sottoreti che desideri usare. Ogni gruppo di sottoreti database deve avere almeno una sottorete in almeno due zone di disponibilità nella Regione AWS. Come best practice, ogni gruppo di sottoreti database deve disporre di almeno una sottorete per ogni zona di disponibilità nella Regione AWS.

Per le implementazioni Multi-AZ, la definizione di una sottorete per tutte le zone di disponibilità in una Regione AWS consente ad Amazon RDS di creare una nuova replica in standby in un'altra zona di disponibilità, se necessario. È possibile seguire questa best practice anche per le distribuzioni con singola zona di disponibilità, perché in futuro è possibile convertirle in distribuzioni Multi-AZ.

Per un'istanza database che deve essere accessibile pubblicamente, le sottoreti nel gruppo di sottoreti database devono disporre di un gateway Internet. Per ulteriori informazioni sui gateway Internet, consulta Eseguire la connessione a Internet utilizzando un gateway Internet nella Guida per l'utente di Amazon VPC.

> **Note**
>
> Il gruppo di sottorete DB per un'area locale può avere solo una sottorete.

Quando crei un'istanza database in un VPC, puoi selezionare un gruppo di sottoreti DB. Amazon RDS sceglie una sottorete e un indirizzo IP al suo interno da associare all'istanza database. Se non esistono gruppi di sottoreti DB, Amazon RDS crea un gruppo di sottoreti predefinito quando crei un'istanza database. Amazon RDS crea e associa un'interfaccia di rete elastica all'istanza database con tale indirizzo IP. L'istanza database utilizza la zona di disponibilità contenente la sottorete.

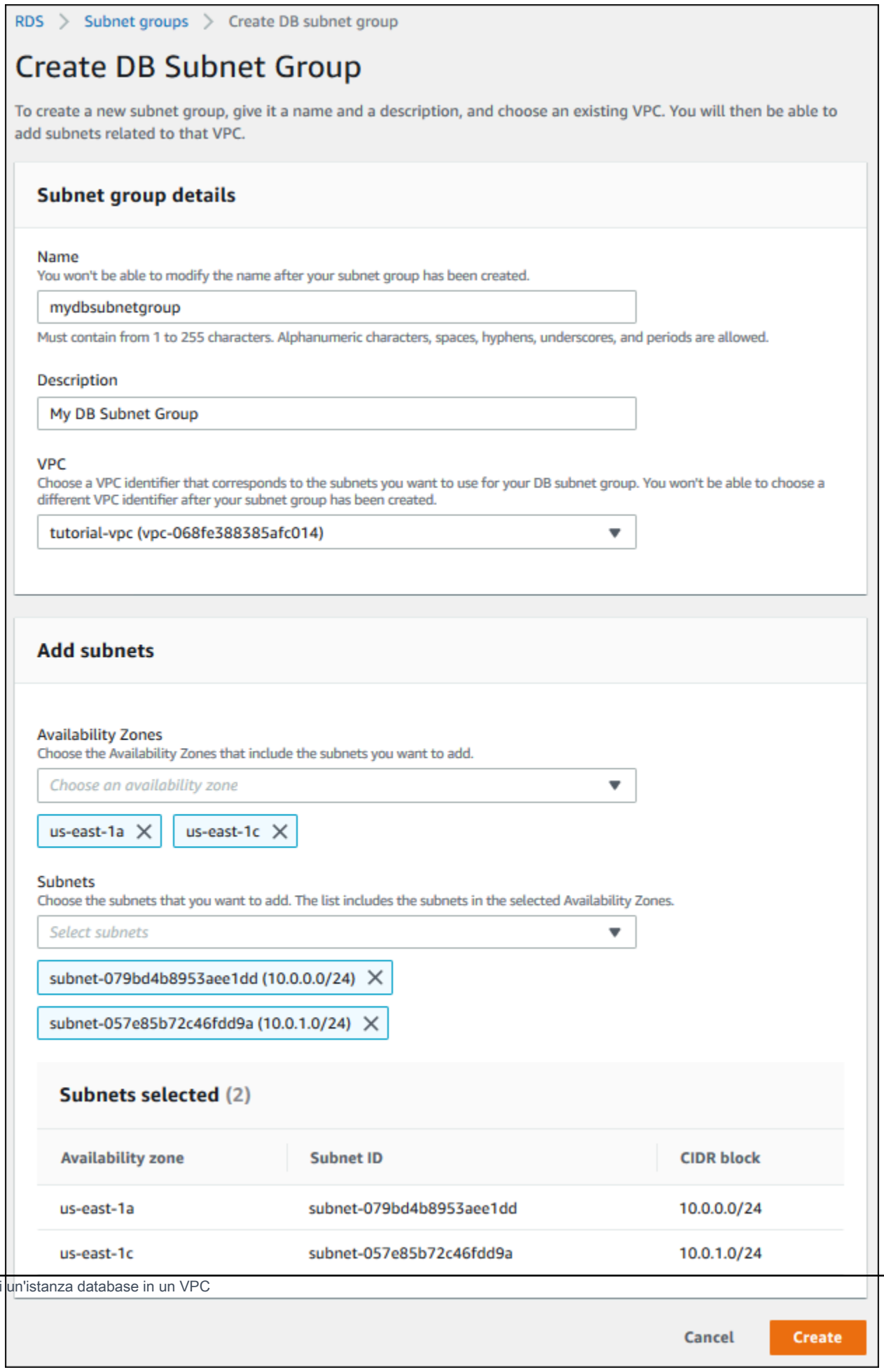
Per le implementazioni Multi-AZ, definire una sottorete per due o più zone di disponibilità in una regione Regione AWS consente ad Amazon RDS di creare un nuovo standby in un'altra zona di disponibilità. Devi effettuare questa operazione anche per le implementazioni Single-AZ, in modo da poterle eventualmente convertire in implementazioni Multi-AZ in un secondo momento.

In questo passaggio, si crea un gruppo di sottoreti database e si aggiungono le sottoreti create per il VPC.

Creare un gruppo di sottoreti database

1. Apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel pannello di navigazione selezionare Subnet groups (Gruppi di sottoreti).
3. Scegli Create DB Subnet Group (Crea gruppo di sottoreti del database).
4. Per Name (Nome), digita il nome del gruppo di sottoreti database.
5. Per Description Descrizione), digita una descrizione per il gruppo di sottoreti database.
6. In VPC, scegli il VPC predefinito o il VPC creato in precedenza.

7. Nella sezione Aggiungi sottoreti, scegliere le zone di disponibilità che includono le sottoreti da Zone di disponibilità, quindi scegliere le sottoreti da Sottoreti.


RDS > Subnet groups > Create DB subnet group
Create DB Subnet Group
To create a new subnet group, give it a name and a description, and choose an existing VPC. You will then be able to add subnets related to that VPC.
Subnet group details
Name
You won't be able to modify the name after your subnet group has been created.
mydbsubnetgroup
Must contain from 1 to 255 characters. Alphanumeric characters, spaces, hyphens, underscores, and periods are allowed.
Description
My DB Subnet Group
VPC
Choose a VPC identifier that corresponds to the subnets you want to use for your DB subnet group. You won't be able to choose a different VPC identifier after your subnet group has been created.
tutorial-vpc (vpc-068fe388385afc014)
Add subnets
Availability Zones
Choose the Availability Zones that include the subnets you want to add.
Choose an availability zone
us-east-1a
us-east-1c
Subnets
Choose the subnets that you want to add. The list includes the subnets in the selected Availability Zones.
Select subnets
subnet-079bd4b8953aee1dd (10.0.0.0/24)
subnet-057e85b72c46fdd9a (10.0.1.0/24)
Subnets selected (2)
Availability zone
Subnet ID
CIDR block
us-east-1a
subnet-079bd4b8953aee1dd
10.0.0.0/24
us-east-1c
subnet-057e85b72c46fdd9a
10.0.1.0/24
Cancel
Create

> **Note**
>
> Se è stata abilitata una zona locale, è possibile scegliere un gruppo zone di disponibilità nella pagina Create DB subnet group (Crea gruppo di sottorete DB). In questo caso, scegliere Availability Zone group (Gruppo zona di disponibilità), Availability Zones (Zone di disponibilità) e Subnet (Sottorete).

8. Scegliere Create (Crea).

   Il nuovo gruppo di sottoreti database viene visualizzato nell'elenco dei gruppi di sottoreti database sulla console RDS. Puoi scegliere il gruppo di sottoreti database per visualizzare i dettagli, comprese tutte le sottoreti associate al gruppo, nel riquadro dei dettagli nella parte inferiore della finestra.

Fase 3: creazione di un gruppo di sicurezza VPC

Prima di creare un'istanza database, devi creare un gruppo di sicurezza VPC da associare tale istanza database. Se non crei un gruppo di sicurezza VPC, puoi utilizzare il gruppo di sicurezza predefinito quando crei un'istanza database. Per istruzioni su come creare un gruppo di sicurezza per l'istanza database, consulta Creazione di un gruppo di sicurezza VPC per un'istanza database privata oppure Controlla il traffico verso le risorse utilizzando gruppi di sicurezza nella Guida per l'utente di Amazon VPC.

Passaggio 4: creazione di un'istanza database nel VPC

In questo passaggio, si crea un'istanza database e si utilizza il nome VPC, il gruppo di sottoreti DB e il gruppo di sicurezza VPC creato nel passaggio precedente.

> **Note**
>
> Se desideri che l'istanza database nel VPC sia pubblicamente accessibile, devi abilitare gli attributi VPC DNS hostnames (Nomi host DNS) e DNS resolution (Risoluzione DNS). Per ulteriori informazioni, consulta Attributi DNS per il VPC nella Guida per l'utente di Amazon VPC.

Per informazioni dettagliate su come creare un'istanza database, consulta Creazione di un'istanza database Amazon RDS.

Quando richiesto nella sezione Connectivity (Connettività), inserisci il nome VPC, il gruppo di sottoreti DB e il gruppo di sicurezza VPC.

Aggiornamento del VPC per un'istanza database

Puoi utilizzare la AWS Management Console per spostare l'istanza database in un VPC diverso.

Per ulteriori informazioni sulla modifica di un'istanza di database, consulta Modifica di un'istanza database Amazon RDS. Nella sezione Connectivity (Connettività) della pagina di modifica, immetti il nuovo gruppo di sottoreti in DB subnet group (Gruppo di sottoreti DB). Il nuovo gruppo di sottoreti deve essere un gruppo di sottoreti in un nuovo VPC.

Non è possibile modificare il VPC per un'istanza database se si applicano le seguenti condizioni:

- L'istanza database si trova in più zone di disponibilità. Puoi convertire l'istanza database in una singola zona di disponibilità, spostarla in un nuovo VPC e quindi convertirla in un'istanza database Multi-AZ. Per ulteriori informazioni, consulta Configurazione e gestione di un'implementazione multi-AZ.
- L'istanza database contiene una o più repliche di lettura. Puoi rimuovere le repliche di lettura, spostare l'istanza database in un nuovo VPC e quindi aggiungere nuovamente le repliche di lettura. Per ulteriori informazioni, consulta Uso delle repliche di lettura dell'istanza database.
- L'istanza database è una replica di lettura. Puoi promuovere la replica di lettura e quindi spostare l'istanza database autonoma su un nuovo VPC. Per ulteriori informazioni, consulta Promozione di una replica di lettura a istanza database standalone.
- Il gruppo di sottoreti nel VPC di destinazione non ha sottoreti nella zona di disponibilità dell'istanza database. Puoi aggiungere sottoreti nella zona di disponibilità dell'istanza database al gruppo di sottoreti di database e quindi spostare l'istanza database nel nuovo VPC. Per ulteriori informazioni, consulta Utilizzo di gruppi di sottoreti database.

Scenari per accedere a un'istanza database in un VPC

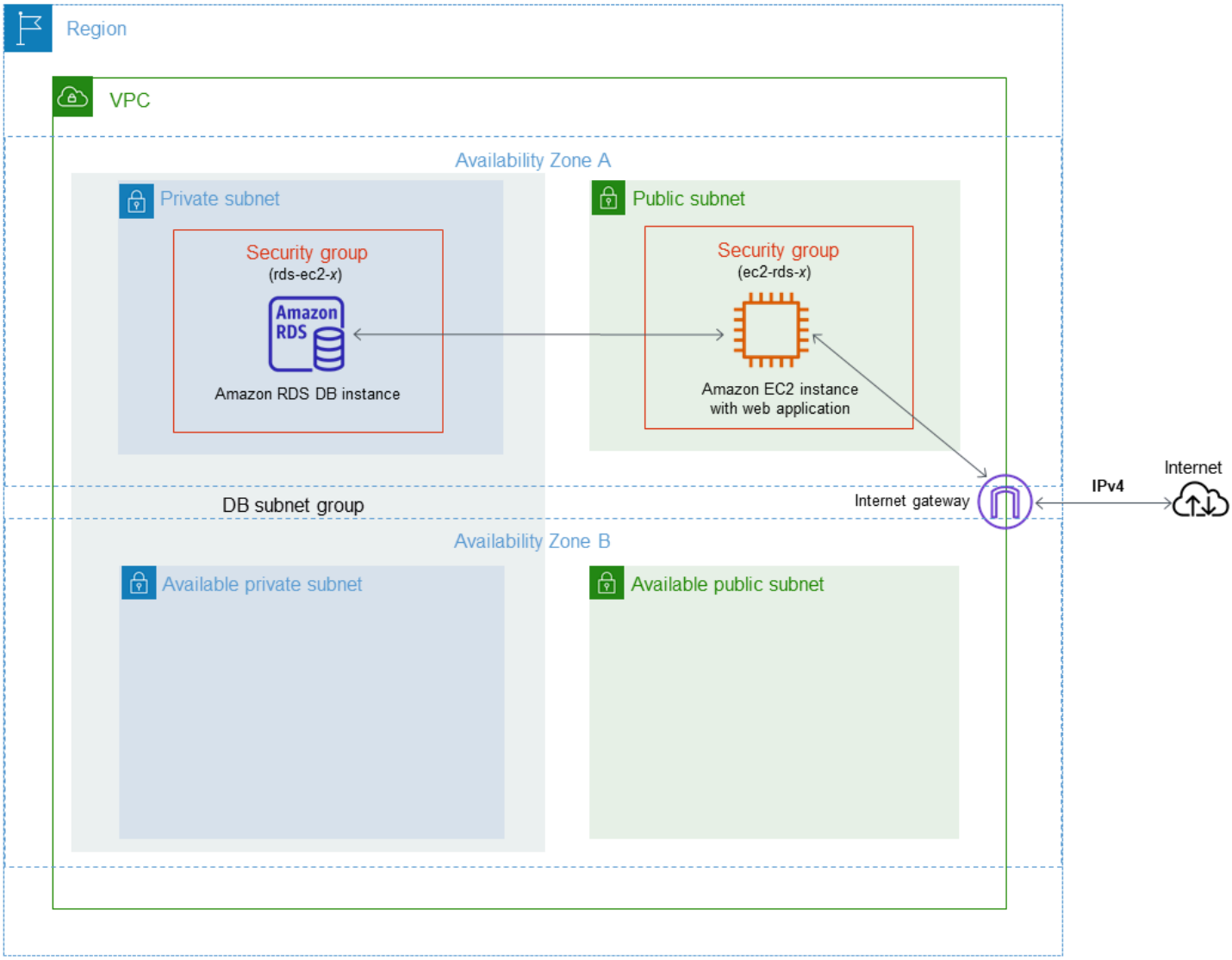
Amazon RDS supporta i seguenti scenari per accedere a un'istanza database in un VPC:

- Un'istanza EC2 nello stesso VPC
- Un'istanza EC2 in un VPC diverso
- Un'applicazione client tramite internet
- Una rete privata

Un'istanza database in un VPC a cui accede un'istanza EC2 nello stesso VPC

Un uso comune di un'istanza database in un VPC è quello di condividere dati con un server di applicazione in esecuzione in un'istanza EC2 nello stesso VPC.

Il seguente diagramma mostra questo scenario.



Il modo più semplice per gestire l'accesso tra istanze EC2 e istanze database nello stesso VPC consiste nel fare quanto segue:

- Creare un gruppo di sicurezza VPC in cui si troveranno le istanze database. Questo gruppo di sicurezza può essere usato per limitare l'accesso alle istanze database. Ad esempio, puoi creare una regola personalizzata per questo gruppo di sicurezza. Ciò potrebbe consentire l'accesso TCP usando la porta assegnata all'istanza database al momento della creazione della stessa e un indirizzo IP utilizzato per accedere all'istanza database per lo sviluppo o per altri scopi.
- Crea un gruppo di sicurezza VPC in cui si troveranno le istanze EC2 (server Web e client). Questo gruppo di sicurezza può, se necessario, consentire l'accesso all'istanza EC2 da Internet tramite la tabella di routing del VPC. Ad esempio, può impostare regole in questo gruppo di sicurezza per consentire l'accesso TCP all'istanza EC2 sulla porta 22.

- Creare regole personalizzate nel gruppo di sicurezza per le istanze database che consentono connessioni dal gruppo di sicurezza creato per le istanze EC2. Queste regole potrebbero consentire a qualsiasi membro del gruppo di sicurezza di accedere alle istanze database.

È disponibile una sottorete pubblica e privata aggiuntiva in una zona di disponibilità separata. Un gruppo di sottoreti DB RDS richiede una sottorete in almeno due zone di disponibilità. La sottorete aggiuntiva semplifica il passaggio a un'implementazione Multi-AZ di un'istanza DB in futuro.

Per una dimostrazione che mostri come creare un VPC con sottoreti pubbliche e private per questo scenario, consulta Tutorial: Creazione di un Amazon VPC da utilizzare con un'istanza database (solo IPv4).

> **Tip**
>
> Quando crei un'istanza database, puoi configurare automaticamente la connettività di rete tra un'istanza Amazon EC2 e un'istanza database. Per ulteriori informazioni, consulta .

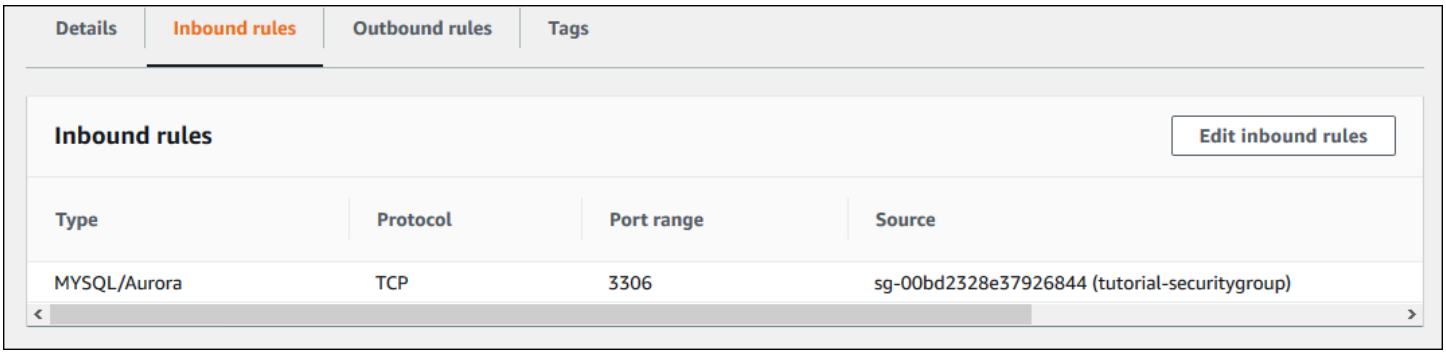
Per creare una regola in un gruppo di sicurezza VPC che consente delle connessioni da un altro gruppo di sicurezza, esegui la procedura seguente:

1. Accedere ad AWS Management Console e aprire la console Amazon VPC all'indirizzo https://console.aws.amazon.com/vpc.
2. Fai clic su Security Groups (Gruppi di sicurezza) nel pannello di navigazione.
3. Scegli o crea un gruppo di sicurezza per il quale desideri concedere l'accesso ai membri di un altro gruppo di sicurezza. Nello scenario precedente, questo è il gruppo di sicurezza utilizzato per le istanze database. Seleziona la scheda Regole in entrata, quindi scegli Modifica regola.
4. Nella scheda Modifica regole in entrata, seleziona Aggiungi regola.
5. Per Tipo, scegli la voce che corrisponde alla porta utilizzata durante la creazione dell'istanza database, ad esempio MySQL/Aurora.
6. Nella casella Origine iniziare a digitare l'ID del gruppo di sicurezza, che elenca i gruppi di sicurezza corrispondenti. Scegli il gruppo di sicurezza con i membri che desideri abbiano accesso alle risorse protette da questo gruppo di sicurezza. Nello scenario precedente, questo è il gruppo di sicurezza utilizzato per le istanze EC2.
7. Se necessario, ripeti i passaggi per il protocollo TCP creando una regola con All TCP (Tutti i TCP) come Tipo e il gruppo di sicurezza nella casella Source (Origine). Se desideri usare

il protocollo UDP, crea una regola con All UDP (Tutti i UDP) come Type (Tipo) e il gruppo di sicurezza nella casella Source (Origine).

8. Scegliere Save rules (Salva regole).

Nella schermata seguente viene illustrata una regola in entrata con un gruppo di sicurezza per la relativa origine.

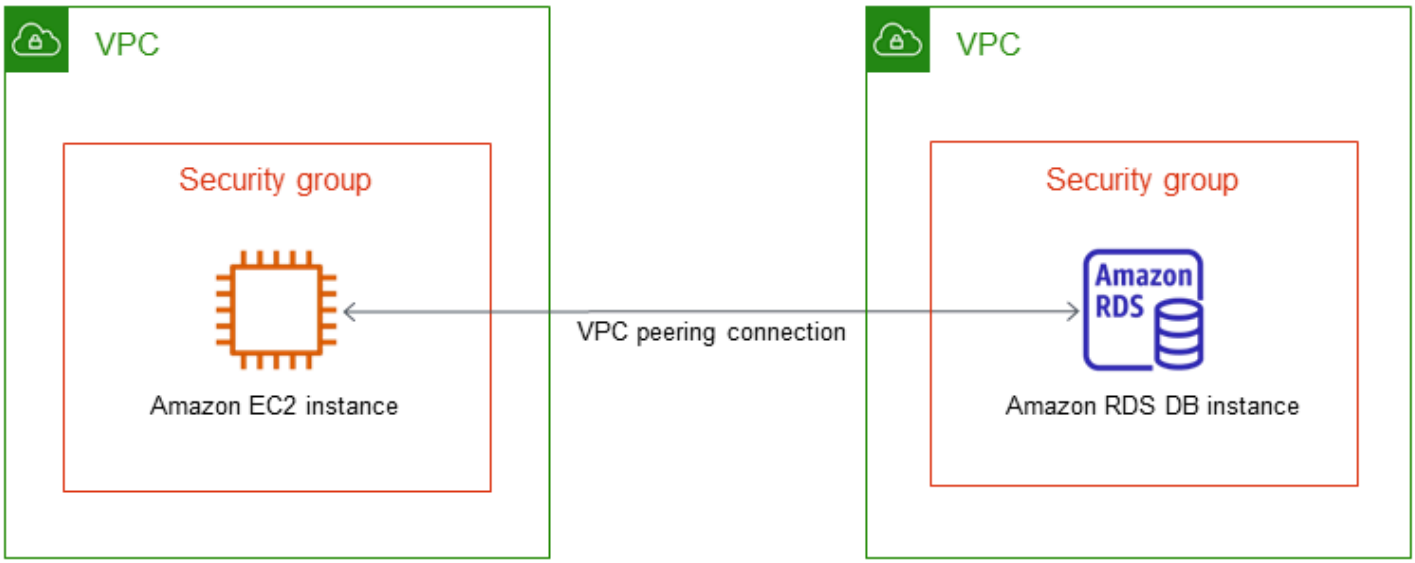


Per ulteriori informazioni sulla connessione all'istanza database dall'istanza EC2, consulta Connessione a un'istanza database Amazon RDS .

Un'istanza database in un VPC a cui accede un'istanza EC2 in un VPC diverso

Quando le istanze database si trova in un VPC diverso dall'istanza EC2 che si sta utilizzando per accedervi, puoi utilizzare il peering VPC per accedere all'istanza database.

Il seguente diagramma mostra questo scenario.

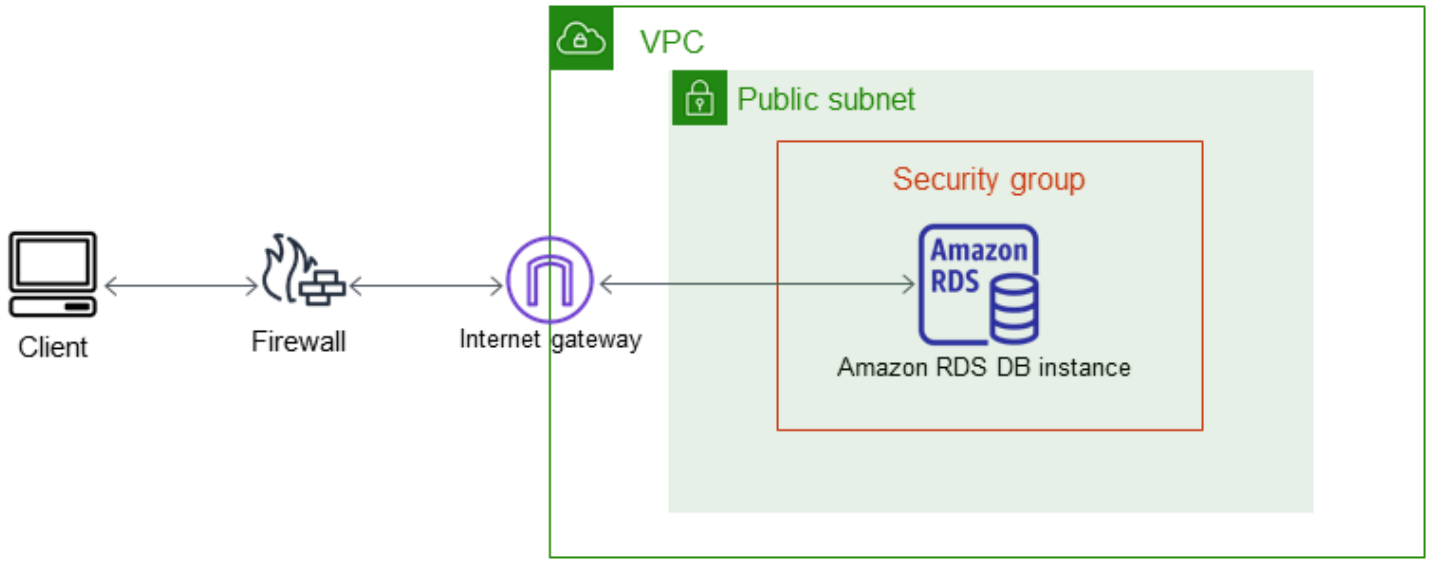


Una connessione di peering di VPC è una connessione di rete tra due VPC che consentono di instradare il traffico tra loro utilizzando degli indirizzi IP privati. Le risorse in uno qualsiasi dei VPC possono comunicare tra loro come se fossero nella stessa rete. Puoi anche creare una connessione di peering VPC tra VPC, con un VPC in un altro account AWS o con un VPC in una Regione AWS diversa. Per ulteriori informazioni su VPC in peering, consulta Peering di VPC nella Guida per l'utente di Amazon Virtual Private Cloud.

Un'istanza database in un VPC a cui accede un'applicazione client tramite Internet

Per accedere a istanze database in un VPC da un'applicazione client tramite Internet, configura un VPC con una sottorete pubblica singola e un gateway Internet per abilitare la comunicazione in Internet.

Il seguente diagramma mostra questo scenario.



È consigliabile utilizzare la seguente configurazione:

- Un VPC di dimensione /16 (ad esempio, CIDR: 10.0.0.0/16). Questa dimensione fornisce indirizzi 65.536 indirizzi IP privati.
- Una sottorete di dimensione /24 (ad esempio, CIDR: 10.0.0.0/24). Questa dimensione fornisce 256 indirizzi IP privati.
- Un'istanza database Amazon RDS che è in associazione al VPC e alla sottorete. Amazon RDS assegna un indirizzo IP nella sottorete all'istanza database.
- Un gateway Internet che collega il VPC a Internet e agli altri prodotti AWS.

- Un gruppo di sicurezza associato all'istanza database. Le regole in entrata del gruppo di sicurezza consentono all'applicazione client di accedere all'istanza database.

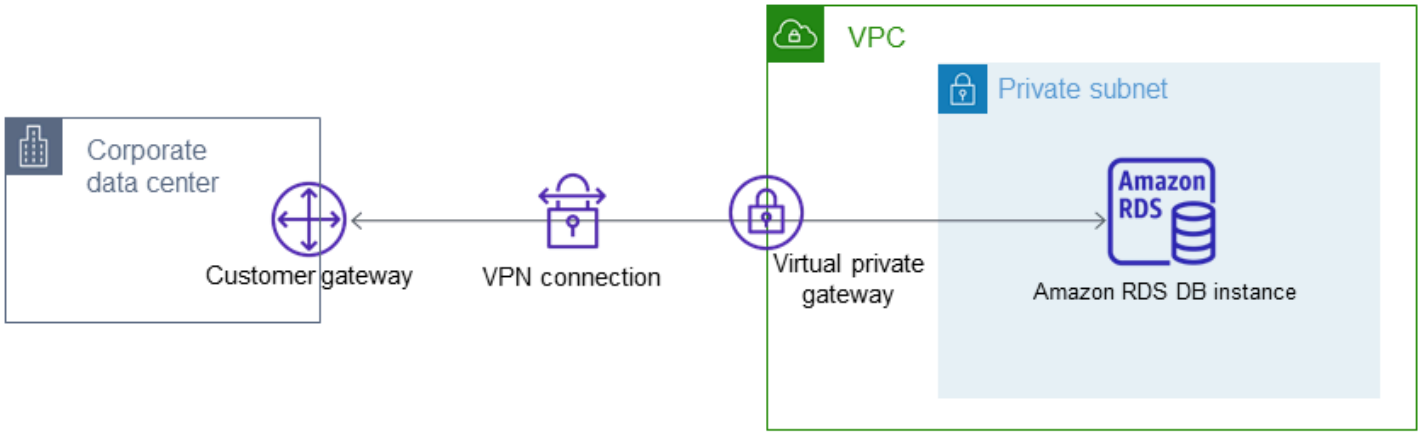
Per informazioni su come creare un'istanza database in un VPC, consulta Creazione di un'istanza database in un VPC.

Un'istanza database in un VPC a cui si accede da una rete privata

Se l'istanza database non è accessibile pubblicamente, sono disponibili le seguenti opzioni per consentire l'accesso da una rete privata:

- Una connessione Site-to-Site VPNAWS. Per ulteriori informazioni, consulta Che cos'è AWS Site-to-Site VPN?
- Una connessione AWS Direct Connect. Per ulteriori informazioni, consulta Che cos'è AWS Direct Connect?
- Una connessione AWS Client VPN. Per ulteriori informazioni, consulta Che cos'è AWS Client VPN?

Il diagramma seguente mostra uno scenario con una connessione Site-to-Site VPN AWS.

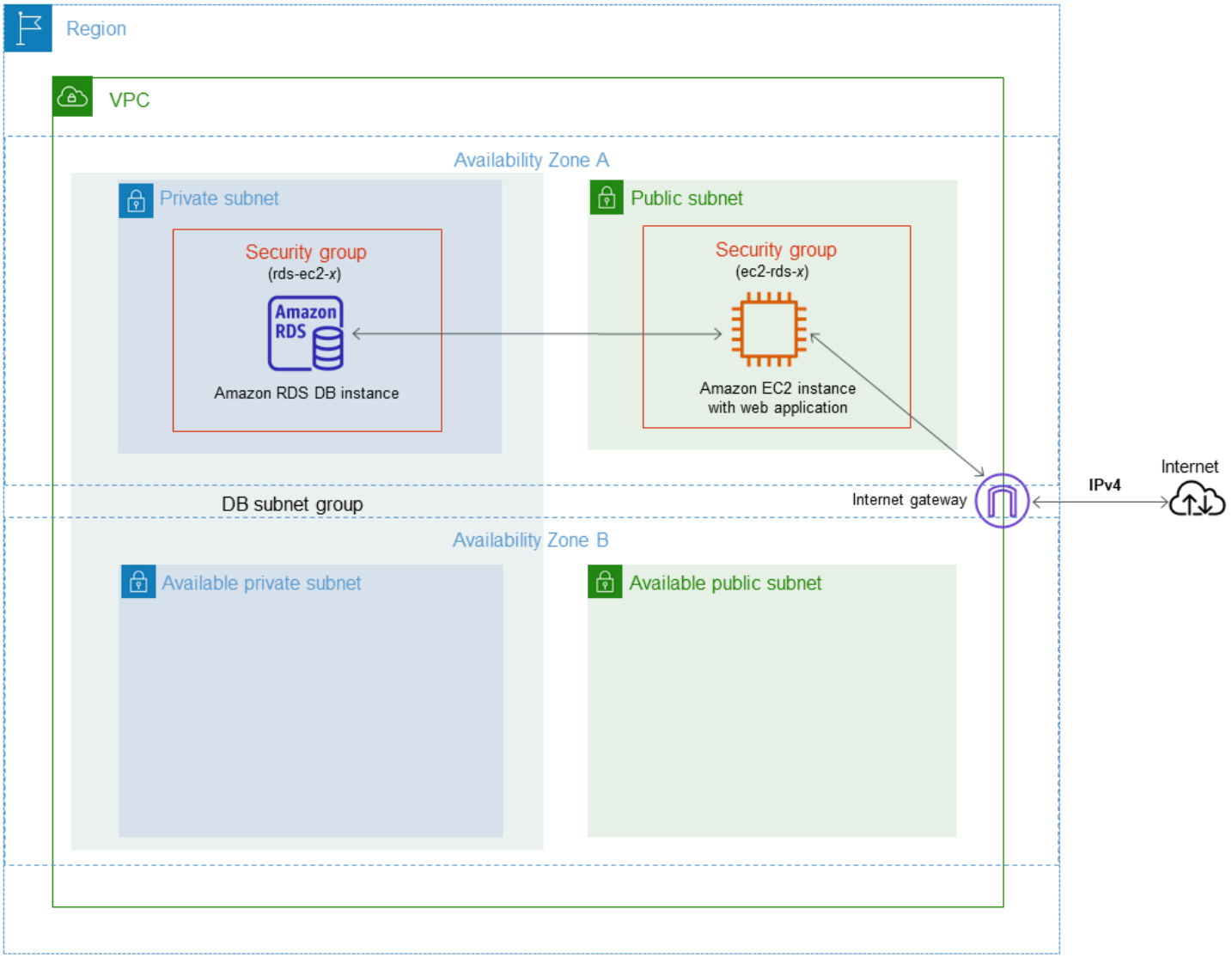


Per ulteriori informazioni, consulta Riservatezza del traffico Internet.

Tutorial: Creazione di un Amazon VPC da utilizzare con un'istanza database (solo IPv4)

Uno scenario comune include un'istanza database in un cloud privato virtuale (VPC) basato sul servizio Amazon VPC. Questo VPC condivide i dati con un server Web in esecuzione nello stesso VPC. In questo tutorial eseguirai la creazione del VPC in questo scenario.

Il seguente diagramma mostra questo scenario. Per informazioni su altri scenari, consulta Scenari per accedere a un'istanza database in un VPC.



L'istanza database deve essere disponibile solo per il server Web e non per la rete Internet pubblica. Pertanto, occorre creare un VPC con sottoreti pubbliche e private. Il server Web è ospitato nella sottorete pubblica, in modo da poter raggiungere l'Internet pubblico. L'istanza database è ospitata

in una sottorete privata. Il server Web può connettersi all'istanza database perché è ospitato nello stesso VPC. Tuttavia, l'istanza database non è disponibile per la rete Internet pubblica e ciò garantisce una maggiore sicurezza.

Questo tutorial configura una sottorete pubblica e privata aggiuntiva in una zona di disponibilità separata. Queste sottoreti non vengono utilizzate dal tutorial. Un gruppo di sottoreti DB RDS richiede una sottorete in almeno due zone di disponibilità. La sottorete aggiuntiva semplifica il passaggio a un'implementazione Multi-AZ di un'istanza DB in futuro.

In questa esercitazione viene descritta la configurazione di un VPC per istanze DB Amazon RDS. Per un'esercitazione che illustra come creare un server Web per questo scenario VPC, consulta Tutorial: creazione di un server Web e un'istanza database Amazon RDS. Per ulteriori informazioni su Amazon VPC, consulta Guida alle operazioni di base di Amazon VPC e Guida per l'utente di Amazon VPC.

> **Tip**
>
> Quando crei un'istanza database, puoi configurare automaticamente la connettività di rete tra un'istanza Amazon EC2 e un'istanza database. La configurazione di rete è simile a quella descritta in questo tutorial. Per ulteriori informazioni, consulta Configurazione della connettività di rete automatica con un'istanza EC2.

Creazione di un VPC con sottoreti pubbliche e private

Utilizza la procedura seguente per creare un VPC con sottoreti pubbliche e private.

Per creare un VPC e sottoreti

1. Accedere alla console Amazon VPC all'indirizzo https://console.aws.amazon.com/vpc/.
2. Nell'angolo superiore destro della AWS Management Console scegliere la regione in cui creare il VPC. In questo esempio viene utilizzata la regione Stati Uniti occidentali (Oregon).
3. Nell'angolo in alto a sinistra, scegli VPC Dashboard (Pannello di controllo VPC). Per iniziare a creare un VPC, scegli Create VPC (Crea VPC).
4. In Resources to create (Risorse da creare) nell'area VPC settings (Impostazioni VPC), scegli VPC and more (VPC e altro).
5. In VPC settings (Impostazioni VPC) restanti, imposta i seguenti valori:

- Name tag auto-generation (Generazione automatica del tag del nome): **`tutorial`**
- IPv4 CIDR block (Blocco CIDR IPv4): **`10.0.0.0/16`**
- IPv6 CIDR block (Blocco IPv6 CIDR): No IPv6 CIDR Block (Nessun blocco IPv6 CIDR)
- Tenancy (Locazione): Default
- Number of Availability Zones (AZs) (Numero di zone di disponibilità): 2
- Customize AZs (Personalizza le zone di disponibilità): mantieni i valori predefiniti.
- Number of public subnet (Numero di sottoreti pubbliche): 2
- Number of private subnets (Numero di sottoreti private): 2
- Customize subnets CIDR blocks (Personalizza blocchi CIDR delle sottoreti): mantieni i valori predefiniti.
- NAT gateways ($) (Gateway NAT): nessuna
- VPC endpoints (Endpoint VPC): nessuno
- DNS options (Opzioni DNS): mantieni i valori predefiniti.

> **Note**
>
> Amazon RDS richiede almeno due sottoreti in due zone di disponibilità diverse per supportare le implementazioni Multi-AZ di un'istanza DB. In questo tutorial viene creata crea un'implementazione Single-AZ, ma il requisito semplifica la conversione in un'implementazione Multi-AZ di un'istanza DB in futuro.

6. Seleziona Create VPC (Crea VPC).

Creazione di un gruppo di sicurezza VPC per un server Web pubblico

È possibile a questo punto aggiungere un gruppo di sicurezza per l'accesso pubblico. Per connetterti alle istanze EC2 nel VPC, aggiungi regole in entrata al gruppo di sicurezza VPC. Queste consentono al traffico di connettersi da Internet.

Per creare un gruppo di sicurezza VPC

1. Aprire la console Amazon VPC all'indirizzo https://console.aws.amazon.com/vpc/.
2. Scegliere VPC Dashboard (Pannello di controllo VPC), Security Groups (Gruppi di sicurezza) e quindi Create security group (Crea gruppo di sicurezza).

3. Nella pagina Create security group (Crea gruppo di sicurezza) impostare questi valori:

   - Security group name: (Nome del gruppo di sicurezza: **`tutorial-securitygroup`**
   - Descrizione: **`Tutorial Security Group`**
   - VPC: scegliere il VPC creato in precedenza, ad esempio vpc-***`identifier`*** (tutorial-vpc)

4. Aggiungere regole in entrata al gruppo di sicurezza

   a. Determina l'indirizzo IP da utilizzare per connettersi alle istanze EC2 nel VPC utilizzando Secure Shell (SSH). Per determinare l'indirizzo IP pubblico, in una finestra o una scheda del browser diversa, è possibile utilizzare il servizio all'indirizzo https://checkip.amazonaws.com. Un esempio di indirizzo IP è `203.0.113.25/32`.

      In molti casi, è possibile eseguire la connessione tramite un fornitore di servizi Internet (ISP) o con la protezione di un firewall senza un indirizzo IP statico. In tal caso, trova l'intervallo di indirizzi IP utilizzati dai computer client.

      > **Warning**
      >
      > Se utilizzi `0.0.0.0/0` per l'accesso SSH, consenti a tutti gli indirizzi IP di accedere alle istanze pubbliche utilizzando SSH. Questo approccio è accettabile per un breve periodo di tempo in un ambiente di test, ma non è sicuro per gli ambienti di produzione. In produzione, autorizza solo un determinato indirizzo IP o un intervallo di indirizzi per accedere alle istanze utilizzando SSH.

   b. Nella sezione Regole in entrata, scegliere Aggiungi regola.

   c. Imposta i seguenti valori per la nuova regola in entrata per consentire l'accesso SSH all'istanza Amazon EC2. In tal caso, è possibile eseguire la connessione all'istanza Amazon EC2 per installare il server Web e altre utility. Puoi connetterti all'istanza EC2 anche per caricare contenuto per il server Web.

      - Tipo: **`SSH`**
      - Origine: indirizzo IP o intervallo ottenuto nella fase 1, ad esempio **`203.0.113.25/32`**.

   d. Scegli Aggiungi regola.

   e. Imposta i seguenti valori per la nuova regola in entrata per consentire l'accesso HTTP al server Web:

      - Tipo: **`HTTP`**

- Origine: **0.0.0.0/0**

5. Per creare il gruppo di sicurezza, scegli Create security group (Crea gruppo di sicurezza).

   Prendi nota dell'ID del gruppo di sicurezza perché sarà necessario in seguito in questo tutorial.

Creazione di un gruppo di sicurezza VPC per un'istanza database privata

Per mantenere l'istanza database privata, crea un secondo gruppo di sicurezza per l'accesso privato. Per connetterti alle istanze database private nel VPC, aggiungi regole in entrata al gruppo di sicurezza VPC per consentire il traffico solo dal server Web.

Per creare un gruppo di sicurezza VPC

1. Aprire la console Amazon VPC all'indirizzo https://console.aws.amazon.com/vpc/.
2. Scegliere VPC Dashboard (Pannello di controllo VPC), Security Groups (Gruppi di sicurezza) e quindi Create security group (Crea gruppo di sicurezza).
3. Nella pagina Create security group (Crea gruppo di sicurezza) impostare questi valori:
   - Security group name: (Nome del gruppo di sicurezza: **tutorial-db-securitygroup**
   - Descrizione: **Tutorial DB Instance Security Group**
   - VPC: scegliere il VPC creato in precedenza, ad esempio vpc-***identifier*** (tutorial-vpc)
4. Aggiungere regole in entrata al gruppo di sicurezza
   a. Nella sezione Regole in entrata, scegliere Aggiungi regola.
   b. Impostare i valori seguenti per la nuova regola in entrata per consentire il traffico MySQL sulla porta 3306 dall'istanza Amazon EC2. In tal caso, puoi connetterti dal server Web all'istanza database ed eseguire l'archiviazione e il recupero dei dati dall'applicazione Web nel database.
      - Tipo: **MySQL/Aurora**
      - Source (Origine): l'identificatore del gruppo di sicurezza tutorial-securitygroup creato in precedenza in questo tutorial, ad esempio sg-9edd5cfb.
5. Per creare il gruppo di sicurezza, scegli Create security group (Crea gruppo di sicurezza).

Per creare un gruppo di sottoreti del database

Un gruppo di sottoreti DB è una raccolta di sottoreti creata in un VPC e che è possibile indicare per le istanze database. Un gruppo di sottoreti DB consente di specificare un determinato VPC quando si creano istanze database.

Creare un gruppo di sottoreti database

1. Identifica le sottoreti private per il database nel VPC.

   a. Accedere alla console Amazon VPC all'indirizzo https://console.aws.amazon.com/vpc/.

   b. Scegli VPC Dashboard (Pannello di controllo VPC), quindi seleziona Subnets (Sottoreti).

   c. Prendi nota degli ID sottorete delle sottoreti denominati tutorial-subnet-private1-us-west-2a e tutorial-subnet-private2-us-west-2b.

      Gli ID sottorete sono necessari quando si crea il gruppo di sottoreti DB.

2. Apri la console di Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.

   Assicurarsi di connettersi alla console Amazon RDS e non alla console Amazon VPC.

3. Nel pannello di navigazione selezionare Subnet groups (Gruppi di sottoreti).

4. Scegli Create DB Subnet Group (Crea gruppo di sottoreti del database).

5. Nella pagina Create DB subnet group (Crea gruppo di sottoreti del database) impostare questi valori in Subnet group details (Dettagli gruppi di sottoreti):

   - Nome: **`tutorial-db-subnet-group`**
   - Descrizione: **`Tutorial DB Subnet Group`**
   - VPC: tutorial-vpc (vpc-***`identifier`***)

6. Nella sezione Aggiungi sottoreti, scegliere Zone di disponibilità e Sottoreti.

   Per questo tutorial, scegli us-west-2a e us-west-2b per l'opzione Availability Zones (Zone di disponibilità). In Subnets (Sottoreti), scegli le sottoreti private identificate nella fase precedente.

7. Seleziona Crea.

   Il nuovo gruppo di sottoreti database viene visualizzato nell'elenco dei gruppi di sottoreti database sulla console RDS. Puoi scegliere il gruppo di sottoreti DB per visualizzare i dettagli nel riquadro dei dettagli nella parte inferiore della finestra. Questi dettagli includono tutte le sottoreti associate al gruppo.

> **Note**
>
> Se questo VPC è stato creato per il completamento di Tutorial: creazione di un server Web e un'istanza database Amazon RDS, creare l'istanza database seguendo le istruzioni riportate in Creazione di un'istanza database Amazon RDS .

Eliminazione del VPC

Dopo aver creato il VPC e altre risorse per questo tutorial, è possibile eliminarle se non sono più necessarie.

> **Note**
>
> Se hai aggiunto risorse nel VPC creato per questo tutorial, potrebbe essere necessario eliminarle prima di poter eliminare il VPC. Ad esempio, queste risorse potrebbero includere istanze Amazon EC2 o istanze database Amazon RDS. Per ulteriori informazioni, consulta Eliminazione del VPC nella Guida per l'utente di Amazon VPC.

Per eliminare un VPC e le risorse correlate

1. Eliminare il gruppo di sottoreti di database.
    a. Apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
    b. Nel pannello di navigazione selezionare Subnet groups (Gruppi di sottoreti).
    c. Selezionare il gruppo di sottoreti di database che si desidera eliminare, ad esempio tutorial-db-subnet-group.
    d. Scegliere Delete (Elimina) e quindi scegliere Delete (Elimina) nella finestra di conferma.
2. Nota l'ID VPC.
    a. Accedere alla console Amazon VPC all'indirizzo https://console.aws.amazon.com/vpc/.
    b. ScegliereVPC Dashboard (Pannello di controllo VPC) e quindi scegliere VPC.
    c. Nell'elenco, identifica il VPC creato, ad esempio tutorial-vpc.
    d. Prendi nota del valore ID VPC del VPC creato. L'ID VPC è richiesto nei passaggi successivi.
3. Eliminare i gruppi di sicurezza.

a. Accedere alla console Amazon VPC all'indirizzo https://console.aws.amazon.com/vpc/.
b. Scegliere VPC Dashboard (Pannello di controllo VPC) e quindi scegliere Security Groups (Gruppi di sicurezza).
c. Seleziona il gruppo di sicurezza per l'istanza database di Amazon RDS, ad esempio tutorial-db-securitygroup.
d. In Actions (Operazioni), scegli Delete security groups (Elimina gruppi di sicurezza) e quindi scegli Delete (Elimina) nella pagina di conferma.
e. Sulla pagina Security Groups (Gruppi di sicurezza), selezionare il gruppo di sicurezza per l'istanza Amazon EC2, ad esempio tutorial-securitygroup.
f. In Actions (Operazioni), scegli Delete security groups (Elimina gruppi di sicurezza) e quindi scegli Delete (Elimina) nella pagina di conferma.

4. Eliminare il VPC.

a. Accedere alla console Amazon VPC all'indirizzo https://console.aws.amazon.com/vpc/.
b. ScegliereVPC Dashboard (Pannello di controllo VPC) e quindi scegliere VPC.
c. Selezionare il VPC che si desidera eliminare, ad esempio tutorial-vpc.
d. In Actions (Operazioni), scegliere Delete VPC (Elimina VPC).

   Nella pagina di conferma vengono visualizzate altre risorse associate al VPC che verranno eliminate, incluse le subnet associate.
e. Nella pagina di conferma, immetti **`delete`** e scegliere Delete (Elimina).

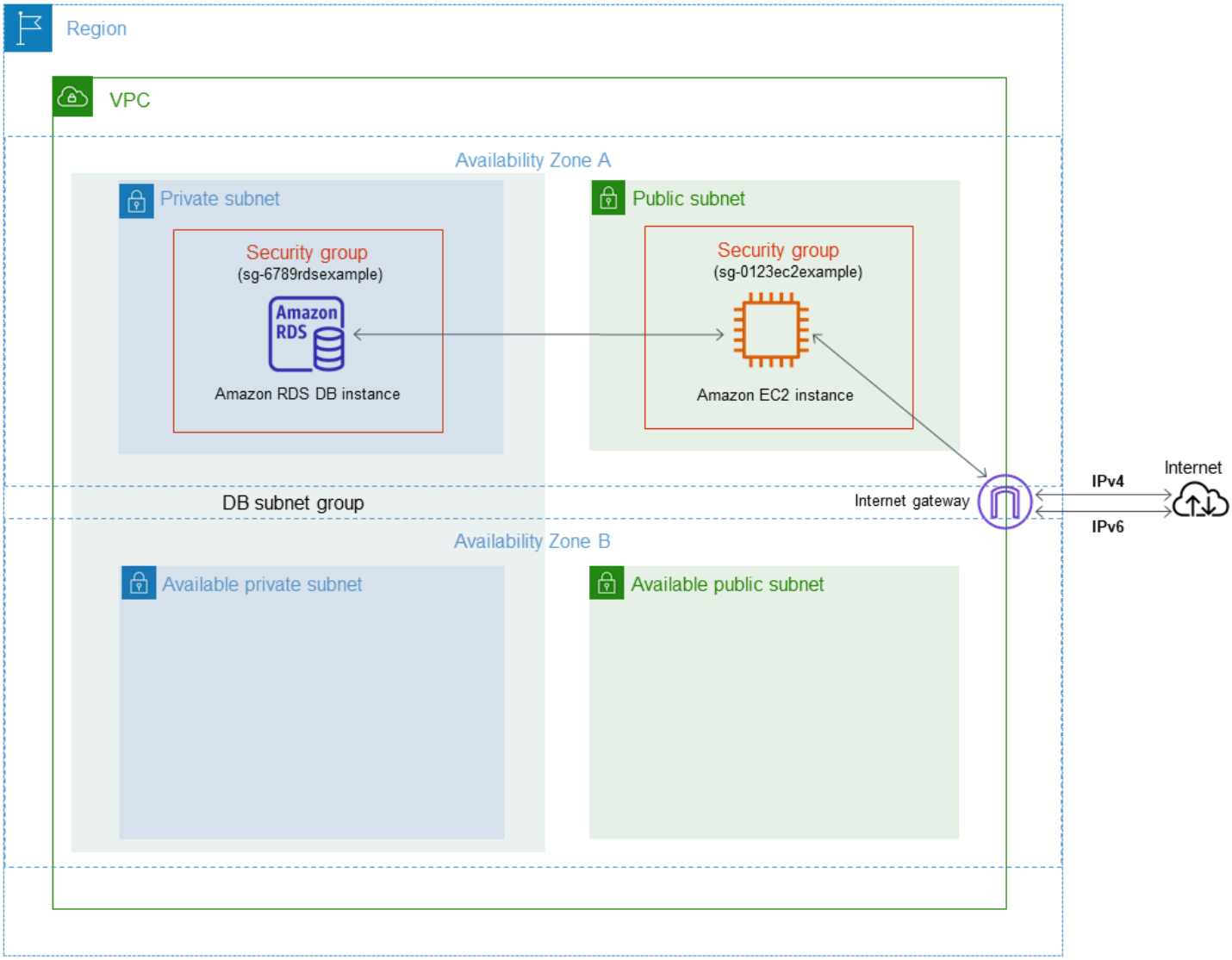
Tutorial: Creazione di un VPC per l'utilizzo con un'istanza database (modalità dual-stack)

Uno scenario comune include un'istanza database in un cloud privato virtuale (VPC) basato sul servizio Amazon VPC. Questo VPC condivide i dati con un'istanza Amazon EC2 pubblica in esecuzione nello stesso VPC.

In questo tutorial, si crea il VPC per questo scenario che funziona con un database in esecuzione in modalità dual-stack. Modalità dual-stack per abilitare la connessione tramite il protocollo di indirizzamento IPv6. Per ulteriori informazioni sull'indirizzamento IP, consulta Assegnazione di indirizzi IP in Amazon RDS.

Le istanze di rete dual-stack sono supportate nella maggior parte delle regioni. Per ulteriori informazioni, consulta Disponibilità di regioni e versioni. Per esaminare le limitazioni della modalità dual-stack, consulta Limitazioni per istanze database di rete dual-stack.

Il seguente diagramma mostra questo scenario.



Per informazioni su altri scenari, consulta Scenari per accedere a un'istanza database in un VPC.

L'istanza database deve essere disponibile solo per l'istanza Amazon EC2 e non per la rete Internet pubblica. Pertanto, occorre creare un VPC con sottoreti pubbliche e private. Il server Web è ospitato nella sottorete pubblica, in modo da poter raggiungere l'Internet pubblico. L'istanza database è ospitata in una sottorete privata. L'istanza Amazon EC2 può connettersi all'istanza database perché è ospitata nello stesso VPC. Tuttavia, l'istanza database non è disponibile per la rete Internet pubblica, fornendo una maggiore sicurezza.

Questo tutorial configura una sottorete pubblica e privata aggiuntiva in una zona di disponibilità separata. Queste sottoreti non vengono utilizzate dal tutorial. Un gruppo di sottoreti DB RDS richiede una sottorete in almeno due zone di disponibilità. La sottorete aggiuntiva semplifica il passaggio a un'implementazione Multi-AZ di un'istanza DB in futuro.

Per creare un'istanza database che utilizza la modalità dual-stack, specificare Dual-stack mode (Modalità dual-stack) per l'opzione Tipo di rete. Si può inoltre modificare un'istanza database usando la stessa impostazione. Per ulteriori informazioni, consulta Creazione di un'istanza database Amazon RDS e Modifica di un'istanza database Amazon RDS.

In questa esercitazione viene descritta la configurazione di un VPC per istanze DB Amazon RDS. Per ulteriori informazioni su Amazon VPC, consulta la Guida per l'utente di Amazon VPC.

Creazione di un VPC con sottoreti pubbliche e private

Utilizza la procedura seguente per creare un VPC con sottoreti pubbliche e private.

Per creare un VPC e sottoreti

1. Accedere alla console Amazon VPC all'indirizzo https://console.aws.amazon.com/vpc/.
2. Nell'angolo in alto a destra della AWS Management Console scegliere la regione in cui creare il VPC. Questo esempio utilizza la regione Stati Uniti orientali (Ohio).
3. Nell'angolo in alto a sinistra, scegli VPC Dashboard (Pannello di controllo VPC). Per iniziare a creare un VPC, scegli Create VPC (Crea VPC).
4. In Resources to create (Risorse da creare) nell'area VPC settings (Impostazioni VPC), scegli VPC and more (VPC e altro).
5. Nei campi VPC settings (Impostazioni VPC) restanti, imposta i seguenti valori:

    - Name tag auto-generation (Generazione automatica del tag del nome): **`tutorial-dual-stack`**
    - IPv4 CIDR block (Blocco CIDR IPv4): **`10.0.0.0/16`**
    - IPv6 CIDR block (Blocco CIDR IPv6): blocco CIDR IPv6 fornito da Amazon
    - Tenancy (Locazione): Default
    - Number of Availability Zones (AZs) (Numero di zone di disponibilità): 2
    - Customize AZs (Personalizza le zone di disponibilità): mantieni i valori predefiniti.
    - Number of public subnet (Numero di sottoreti pubbliche): 2
    - Number of private subnets (Numero di sottoreti private): 2
    - Customize subnets CIDR blocks (Personalizza blocchi CIDR delle sottoreti): mantieni i valori predefiniti.
    - NAT gateways ($) (Gateway NAT): nessuna
    - Egress only internet gateway (Gateway Internet solo in uscita): No

- VPC endpoints (Endpoint VPC): nessuno
- DNS options (Opzioni DNS): mantieni i valori predefiniti.

> **Note**
>
> Amazon RDS richiede almeno due sottoreti in due zone di disponibilità diverse per supportare le implementazioni Multi-AZ di un'istanza DB. In questo tutorial viene creata crea un'implementazione Single-AZ, ma il requisito semplifica la conversione in un'implementazione Multi-AZ di un'istanza DB in futuro.

6. Seleziona Crea VPC.

Per creare un gruppo di sicurezza VPC per un'istanza Amazon EC2 pubblica

È possibile a questo punto aggiungere un gruppo di sicurezza per l'accesso pubblico. Per connettersi alle istanze EC2 nel VPC, aggiungi regole in entrata al gruppo di sicurezza VPC per consentire il traffico da Internet.

Per creare un gruppo di sicurezza VPC

1. Aprire la console Amazon VPC all'indirizzo https://console.aws.amazon.com/vpc/.
2. Scegliere VPC Dashboard (Pannello di controllo VPC), Security Groups (Gruppi di sicurezza) e quindi Create security group (Crea gruppo di sicurezza).
3. Nella pagina Create security group (Crea gruppo di sicurezza) impostare questi valori:
   - Security group name: (Nome del gruppo di sicurezza: **`tutorial-dual-stack-securitygroup`**
   - Descrizione: **`Tutorial Dual-Stack Security Group`**
   - VPC: scegli il VPC creato in precedenza, ad esempio vpc-***`identifier`*** (tutorial-dual-stack-vpc)
4. Aggiungere regole in entrata al gruppo di sicurezza
   a. Determina l'indirizzo IP da utilizzare per connettersi alle istanze EC2 nel VPC utilizzando Secure Shell (SSH).

Un esempio di indirizzo Internet Protocol versione 4 (IPv4) è `203.0.113.25/32`.
Un esempio di intervalli di indirizzi Internet Protocol versione 6 (IPv6) è `2001:db8:1234:1a00::/64`.

In molti casi, è possibile eseguire la connessione tramite un fornitore di servizi Internet (ISP) o con la protezione di un firewall senza un indirizzo IP statico. In tal caso, trova l'intervallo di indirizzi IP utilizzati dai computer client.

> **Warning**
>
> Se utilizzi `0.0.0.0/0` per IPv4 o `::0` per IPv6, consenti a tutti gli indirizzi IP di accedere alle istanze pubbliche utilizzando SSH. Questo approccio è accettabile per un breve periodo di tempo in un ambiente di test, ma non è sicuro per gli ambienti di produzione. In produzione, è preferibile autorizzare l'accesso alle istanze solo a indirizzo IP o a un intervallo di indirizzi specifico.

b. Nella sezione Regole in entrata, scegliere Aggiungi regola.

c. Imposta i seguenti valori per la nuova regola in entrata per consentire l'accesso Secure Shell (SSH) all'istanza Amazon EC2. In questo caso, è possibile connettersi all'istanza EC2 per installare client SQL e altre applicazioni. Specifica un indirizzo IP per poter accedere all'istanza EC2:

- Tipo: **`SSH`**
- Origine: indirizzo IP o intervallo ottenuto nel passaggio a. Un esempio di indirizzo IP IPv4 è **`203.0.113.25/32`**. Un esempio di indirizzo IP IPv6 è **`2001:DB8::/32`**.

5. Per creare il gruppo di sicurezza, scegli Create security group (Crea gruppo di sicurezza).

   Prendi nota dell'ID del gruppo di sicurezza perché sarà necessario in seguito in questo tutorial.

Creazione di un gruppo di sicurezza VPC per un'istanza database privata

Per mantenere l'istanza database privata, crea un secondo gruppo di sicurezza per l'accesso privato. Per connetterti alle istanze database private nel VPC, aggiungi le regole in entrata al gruppo di sicurezza VPC. Queste consentono il traffico solo dall'istanza Amazon EC2.

Per creare un gruppo di sicurezza VPC

1. Aprire la console Amazon VPC all'indirizzo https://console.aws.amazon.com/vpc/.
2. Scegliere VPC Dashboard (Pannello di controllo VPC), Security Groups (Gruppi di sicurezza) e quindi Create security group (Crea gruppo di sicurezza).
3. Nella pagina Create security group (Crea gruppo di sicurezza) impostare questi valori:
   - Security group name: (Nome del gruppo di sicurezza: **`tutorial-dual-stack-db-securitygroup`**
   - Descrizione: **`Tutorial Dual-Stack DB Instance Security Group`**
   - VPC: scegli il VPC creato in precedenza, ad esempio vpc-***`identifier`*** (tutorial-dual-stack-vpc)
4. Aggiungere regole in entrata al gruppo di sicurezza
   a. Nella sezione Regole in entrata, scegliere Aggiungi regola.
   b. Impostare i valori seguenti per la nuova regola in entrata per consentire il traffico MySQL sulla porta 3306 dall'istanza Amazon EC2. In tal modo, puoi eseguire la connessione dall'istanza EC2 all'istanza database ed Questa operazione consente di inviare dati dall'istanza EC2 al database.
      - Tipo: MySQL/Aurora
      - Source (Origine): l'identificatore del gruppo di sicurezza tutorial-dual-stack-securitygroup creato in precedenza in questo tutorial, ad esempio sg-9edd5cfb.
5. Per creare il gruppo di sicurezza, scegliere Crea gruppo di sicurezza.

Per creare un gruppo di sottoreti del database

Un gruppo di sottoreti DB è una raccolta di sottoreti creata in un VPC e che è possibile indicare per le istanze database. L'utilizzo di un gruppo di sottoreti DB consente di specificare un determinato VPC durante la creazione di istanze database. Per creare un gruppo di sottoreti database compatibile con `DUAL`, tutte le sottoreti devono essere compatibili con `DUAL`. Per essere compatibile con `DUAL`, a una sottorete deve essere associato un CIDR IPv6.

Creare un gruppo di sottoreti database

1. Identifica le sottoreti private per il database nel VPC.

a. Accedere alla console Amazon VPC all'indirizzo https://console.aws.amazon.com/vpc/.
b. Scegli VPC Dashboard (Pannello di controllo VPC), quindi seleziona Subnets (Sottoreti).
c. Prendi nota degli ID sottorete delle sottoreti denominati tutorial-dual-stack-subnet-private1-us-west-2a e tutorial-dual-stack-subnet-private2-us-west-2b.

   Gli ID sottorete saranno necessari quando si crea il gruppo di sottoreti DB.

2. Apri la console di Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.

   Assicurarsi di connettersi alla console Amazon RDS e non alla console Amazon VPC.
3. Nel pannello di navigazione selezionare Subnet groups (Gruppi di sottoreti).
4. Scegli Create DB Subnet Group (Crea gruppo di sottoreti del database).
5. Nella pagina Create DB subnet group (Crea gruppo di sottoreti del database) impostare questi valori in Subnet group details (Dettagli gruppi di sottoreti):

   - Nome: **`tutorial-dual-stack-db-subnet-group`**
   - Descrizione: **`Tutorial Dual-Stack DB Subnet Group`**
   - VPC: tutorial-dual-stack-vpc (vpc-***`identifier`***)
6. Nella sezione Add subnets (Aggiungi sottoreti), scegli i valori desiderati per le opzioni Availability Zones (Zone di disponibilità) e Subnets (Sottoreti).

   Per questo tutorial, scegli us-east-2b e us-east-2c per l'opzione Availability Zones (Zone di disponibilità). In Subnets (Sottoreti), scegli le sottoreti private identificate nella fase precedente.
7. Seleziona Crea.

Il nuovo gruppo di sottoreti database viene visualizzato nell'elenco dei gruppi di sottoreti database sulla console RDS. È possibile scegliere il gruppo di sottoreti database per visualizzarne i dettagli. I dettagli includono i protocolli di indirizzamento supportati e tutte le sottoreti associate al gruppo e il tipo di rete supportato dal gruppo di sottoreti database.

Creare un'istanza Amazon EC2 in modalità dual-stack

Per creare un'istanza Amazon EC2, segui le istruzioni riportate in Avvio di un'istanza tramite la nuova procedura guidata di avvio dell'istanza nella Guida per l'utente per istanze Linux di Amazon EC2.

Nella pagina Configure Instance Details (Configura i dettagli dell'istanza), imposta questi valori e lascia gli altri valori come predefiniti:

- Rete: scegli un VPC esistente con sottoreti sia pubbliche che private, ad esempio tutorial-dual-stack-vpc (vpc-*identifier*) creato in Creazione di un VPC con sottoreti pubbliche e private.
- Subnet (Sottorete): scegli una sottorete pubblica esistente, ad esempio subnet-***identifier*** | tutorial-dual-stack-subnet-public1-us-east-2a | us-east-2a creata in Per creare un gruppo di sicurezza VPC per un'istanza Amazon EC2 pubblica.
- Auto-assign Public IP (Assegna automaticamente IP pubblico): scegli Enable (Abilita).
- Auto-assign IPv6 IP (Assegna automaticamente IP IPv6): scegli Enable (Abilita).
- Firewall (security groups) (Firewall (gruppi di sicurezza)): scegli Select an existing security group (Seleziona un gruppo di sicurezza esistente).
- Gruppi di sicurezza comuni: scegli un gruppo di sicurezza esistente, ad esempio il `tutorial-securitygroup` creato in Per creare un gruppo di sicurezza VPC per un'istanza Amazon EC2 pubblica. Assicurarsi che il gruppo di sicurezza scelto includa regole in ingresso per Secure Shell (SSH) e l'accesso HTTP.

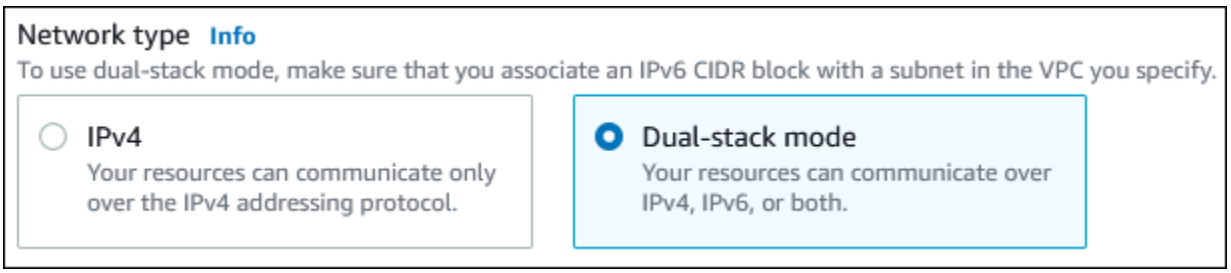
Creazione di un'istanza database in modalità dual-stack

In questo passaggio, viene creata un'istanza database che viene eseguito in modalità dual-stack.

Per creare un'istanza database.

1. Accedi alla AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nell'angolo in alto a destra della console, scegli la Regione AWS in cui desideri creare l'istanza database. Questo esempio utilizza la regione Stati Uniti orientali (Ohio).
3. Nel riquadro di navigazione, scegliere Databases (Database).
4. Scegliere Create database (Crea database).
5. Nella pagina Create database (Crea database), assicurati che l'opzione Standard create (Creazione standard) sia selezionata, quindi scegli il tipo di motore DB MySQL.
6. Nella sezione Connettività, imposta i seguenti valori:

    - Network type (Tipo di rete): scegli Dual-stack mode (Modalità dual-stack).



- Virtual private cloud (VPC) Cloud privato virtuale (VPC): scegli un VPC esistente con sottoreti sia pubbliche che private, ad esempio tutorial-dual-stack-vpc (vpc-*identifier*) creato in Creazione di un VPC con sottoreti pubbliche e private.

  Il VPC deve avere sottoreti in diverse zone di disponibilità.
- DB subnet group (Gruppo di sottoreti DB): scegli un gruppo di sottoreti DB per il VPC, ad esempio tutorial-dual-stack-db-subnet-group creato in Per creare un gruppo di sottoreti del database.
- Public access (Accesso pubblico): scegli No.
- VPC security group (firewall) (Gruppo di sicurezza VPC (firewall)): seleziona Choose existing (Sceglie esistente).
- Gruppi di sicurezza VPC esistenti: scegli un gruppo di sicurezza VPC esistente che sia configurato per accesso privato, ad esempio tutorial-dual-stack-db-securitygroup creato in Creazione di un gruppo di sicurezza VPC per un'istanza database privata.

  Rimuovere gli altri gruppi di sicurezza, come quello predefinito, selezionando la X associata a esso.
- Availability Zone (Zona di disponibilità): scegli us-west-2a.

  Per evitare il traffico tra AZ, assicurati che l'istanza database e l'istanza EC2 si trovino nella stessa zona di disponibilità.

7. Per le restanti sezioni, specifica le impostazioni dell'istanza database. Per informazioni su ciascuna impostazione, consulta Impostazioni per istanze database.

Connessione all'istanza Amazon EC2 e all'istanza database

Dopo aver creato l'istanza Amazon EC2 e l'istanza database in modalità dual-stack, puoi eseguire la connessione utilizzando il protocollo IPv6. Per connetterti a un'istanza Amazon EC2 utilizzando il protocollo IPv6, segui le istruzioni riportate in Connessione all'istanza di Linux nella Guida per l'utente di Amazon EC2 per le istanze Linux.

Per connetterti alla tua istanza database RDS per MySQL dall'istanza Amazon EC2, segui le istruzioni in Connessione di un'istanza database MySQL.

Eliminazione del VPC

Dopo aver creato il VPC e altre risorse per questo tutorial, è possibile eliminarle se non sono più necessarie.

Se hai aggiunto risorse nel VPC creato per questo tutorial, potrebbe essere necessario eliminarle prima di poter eliminare il VPC. Esempi di risorse sono le istanze Amazon EC2 o le istanze database. Per ulteriori informazioni, consulta Eliminazione del VPC nella Guida per l'utente di Amazon VPC.

Per eliminare un VPC e le risorse correlate

1. Eliminare il gruppo di sottoreti database:
    a. Apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
    b. Nel pannello di navigazione selezionare Subnet groups (Gruppi di sottoreti).
    c. Seleziona il gruppo di sottoreti database da eliminare, ad esempio tutorial-db-subnet-group.
    d. Scegliere Delete (Elimina) e quindi scegliere Delete (Elimina) nella finestra di conferma.
2. Annotare l'ID VPC:
    a. Accedere alla console Amazon VPC all'indirizzo https://console.aws.amazon.com/vpc/.
    b. ScegliereVPC Dashboard (Pannello di controllo VPC) e quindi scegliere VPC.
    c. Nell'elenco, individua il VPC creato, ad esempio tutorial-dual-stack-vpc.
    d. Annotare il valore ID VPC del VPC creato. Sarà necessario usare questo ID VPC nei passaggi successivi.
3. Eliminare i gruppi di sicurezza:
    a. Accedere alla console Amazon VPC all'indirizzo https://console.aws.amazon.com/vpc/.
    b. Scegliere VPC Dashboard (Pannello di controllo VPC) e quindi scegliere Security Groups (Gruppi di sicurezza).
    c. Selezionare il gruppo di sicurezza per l'istanza database di Amazon RDS, ad esempio tutorial-dual-stack-db-securitygroup.
    d. In Actions (Operazioni), scegli Delete security groups (Elimina gruppi di sicurezza) e quindi scegli Delete (Elimina) nella pagina di conferma.

e. Nella pagina Security Groups (Gruppi di sicurezza), seleziona il gruppo di sicurezza per l'istanza Amazon EC2, ad esempio tutorial-dual-stack-securitygroup..

f. In Actions (Operazioni), scegli Delete security groups (Elimina gruppi di sicurezza) e quindi scegli Delete (Elimina) nella pagina di conferma.

4. Elliminare il gateway NAT:

a. Accedere alla console Amazon VPC all'indirizzo https://console.aws.amazon.com/vpc/.

b. Scegliere VPC Dashboard (Pannello di controllo VPC) e quindi scegliere NAT Gateways (Gateway NAT).

c. Seleziona il gateway NAT del VPC creato. Utilizzare l'ID VPC per identificare il gateway NAT corretto.

d. In Actions (Operazioni), scegli Delete NAT gateway (Elimina gateway NAT).

e. Nella pagina di conferma, immetti **`delete`** e scegliere Delete (Elimina).

5. Eliminare il VPC

a. Accedere alla console Amazon VPC all'indirizzo https://console.aws.amazon.com/vpc/.

b. ScegliereVPC Dashboard (Pannello di controllo VPC) e quindi scegliere VPC.

c. Seleziona il VPC da eliminare, ad esempio tutorial-dual-stack-vpc.

d. In Actions (Operazioni), scegliere Delete VPC (Elimina VPC).

Nella pagina di conferma vengono visualizzate altre risorse associate al VPC che verranno eliminate, incluse le subnet associate.

e. Nella pagina di conferma, immetti **`delete`** e scegliere Delete (Elimina).

6. Rilasciare gli indirizzi IP elastici:

a. Apri la console Amazon EC2 all'indirizzo https://console.aws.amazon.com/ec2/.

b. Scegliere EC2 Dashboard (Pannello di controllo EC2) e quindi scegliere Elastic IPs (IP elastici).

c. Seleziona l'indirizzo IP elastico da rilasciare.

d. In Actions (Operazioni), scegli Release Elastic IP addresses (Rilascia indirizzi IP elastici).

e. Nella pagina di conferma scegli Release (Rilasciare).

Lo spostamento di un'istanza database non in un VPC all'interno di un VPC

Alcune istanze database della legacy sulla piattaforma EC2-Classic non si trovano in VPC. Se l'istanza database non si trova in un VPC, puoi ora usare AWS Management Console per spostare con facilità l'istanza database in un VPC. Prima di poter spostare un'istanza database non in un VPC all'interno di un VPC, è necessario creare il VPC.

EC2-Classic è stato ritirato il 15 agosto 2022. Se non hai eseguito la migrazione da EC2-Class ic a un VPC, ti consigliamo di eseguirla il prima possibile. Per ulteriori informazioni, consultare Eseguire la migrazione da EC2-Classic a un VPC nella Guida per l'utente di Amazon EC2 e il blog Il networking EC2-Classic va in pensione: ecco come prepararsi.

Important

Se sei un nuovo cliente Amazon RDS, se non hai mai creato un'istanza database prima o se stai creando un'istanza database in una regione AWS che non hai mai usato prima, in quasi tutti i casi stai usando la piattaforma EC2-VPC e hai un VPC predefinito. Per informazioni sull'utilizzo di istanze database in un VPC, consulta Uso di un'istanza database in un VPC.

Segui questa procedura per creare un VPC per l'istanza database.

- Fase 1. Creazione di un VPC
- Fase 2: creazione di un gruppo di sottoreti database
- Fase 3: creazione di un gruppo di sicurezza VPC

Dopo aver creato il VPC, segui questa procedura per spostare l'istanza database nel VPC.

- Aggiornamento del VPC per un'istanza database

Si consiglia vivamente di creare un backup dell'istanza database immediatamente prima della migrazione. In questo modo è possibile ripristinare i dati in caso di esito negativo della migrazione. Per ulteriori informazioni, consulta Backup, ripristino ed esportazione dei dati.

Le seguenti sono alcune limitazioni che riguardano lo spostamento dell'istanza database nel VPC.

- Classi di istanza database di generazione precedente: le classi di istanza database di generazione precedente potrebbero non essere supportate sulla piattaforma VPC. Quando si sposta un'istanza database in un VPC, scegliere una classe di istanza database db.m3 o db.r3. Dopo aver spostato l'istanza database in un VPC, è possibile dimensionare l'istanza database per utilizzare una classe di istanza database successiva. Per un elenco completo delle classi di istanza supportate da VPC, consulta Tipi di istanza Amazon RDS.
- Multiple-AZ – lo spostamento di un'istanza database Multi-AZ non in un VPC all'interno di un VPC non è attualmente supportato. Per spostare l'istanza database in un VPC, modificare innanzitutto l'istanza database in modo che si tratti di una distribuzione AZ singola. Cambiare l'impostazione Implementazione Multi-AZ su No. Dopo aver spostato l'istanza database in un VPC, modificarla di nuovo per renderlo un'implementazione Multi-AZ. Per ulteriori informazioni, consulta Modifica di un'istanza database Amazon RDS.
- Repliche di lettura: lo spostamento di un'istanza database con repliche di lettura non in un VPC all'interno di un VPC non è attualmente supportato. Per spostare l'istanza database in un VPC, eliminare prima tutte le repliche di lettura. Dopo aver spostato l'istanza database in un VPC, ricreare le repliche di lettura. Per ulteriori informazioni, consulta Uso delle repliche di lettura dell'istanza database.
- Gruppi di opzioni: se si sposta l'istanza database in un VPC e questa utilizza un gruppo di opzioni personalizzato, modificare il gruppo di opzioni associato all'istanza database. I gruppi di opzioni sono specifici della piattaforma e il passaggio a un VPC è un cambiamento nella piattaforma. Per usare un gruppo di opzioni personalizzate in questo caso, assegna il gruppo di opzioni VPC predefinito all'istanza database, assegna un gruppo di opzioni che è utilizzato da altre istanze database nel VPC in cui le stai spostando, oppure crea un nuovo gruppo di opzioni e assegnalo all'istanza database. Per ulteriori informazioni, consulta Uso di gruppi di opzioni.

Alternative per lo spostamento di un'istanza database non in un VPC in un VPC con tempi di inattività minimi

Utilizzando le seguenti alternative, è possibile spostare un'istanza database non in un VPC in un VPC con tempi di inattività minimi. Queste alternative causano un'interruzione minima dell'istanza database di origine e consentono di servire il traffico utenti durante la migrazione. Tuttavia, il tempo necessario per la migrazione a un VPC varia in base alle dimensioni del database e alle caratteristiche del carico di lavoro attivo.

- AWS Database Migration Service (AWS DMS) – AWS DMS consente la migrazione in tempo reale dei dati mantenendo l'istanza database di origine completamente operativa, ma replica solo un

insieme limitato di istruzioni DDL. AWS DMS non propaga elementi quali indici, utenti, privilegi, stored procedure e altre modifiche al database non direttamente correlate ai dati della tabella. Inoltre, AWS DMS non utilizza automaticamente snapshot RDS per la creazione iniziale dell'istanza database e ciò può aumentare il tempo di migrazione. Per ulteriori informazioni, consultare AWS Database Migration Service.

- Ripristino snapshot DB o ripristino point-in-time: è possibile spostare un'istanza database in un VPC ripristinando uno snapshot dell'istanza database o ripristinando un'istanza database in un point-in-time. Per ulteriori informazioni, consulta Ripristino da uno snapshot database e Ripristino a un'ora specifica per un'istanza database.

Quote e vincoli per Amazon RDS

Di seguito, è possibile trovare una descrizione delle quote di risorse e dei vincoli di denominazione per Amazon RDS.

Argomenti

Quote in Amazon RDS

Ogni AWS account dispone di quote, per ogni AWS regione, sul numero di risorse Amazon RDS che è possibile creare. Una volta raggiunta la quota della risorsa, le ulteriori richieste di creazione di tale risorsa restituiranno un errore con un'eccezione.

La tabella seguente elenca le risorse e le relative quote per regione. AWS

Nome	Predefinita	Adattabile	Descrizione
Autorizzazioni per gruppo di sicurezza DB	Ogni regione supportata: 20	No	Numero di autorizzazioni per gruppo di sicurezza DB
Versioni personalizzate del motore	Ogni regione supportata: 40	Sì	Il numero massimo di versioni personalizzate del motore consentite in questo account nella regione corrente
Gruppi di parametri di cluster database	Ogni Regione supportata: 50	No	Il numero massimo di gruppi di parametri del cluster di database

Nome	Predefinita	Adatta e	Descrizione
Cluster database	Ogni regione supportata: 40	Sì	Il numero massimo di cluster Aurora consentiti in questo account nella regione corrente
Istanze DB	Ogni regione supportata: 40	Sì	Il numero massimo di istanze database consentite in questo account nella regione corrente
Gruppi di sottoreti database	Ogni regione supportata: 50	Sì	Il numero massimo di gruppi di sottoreti di database
Dimensione del corpo della richiesta HTTP dell'API dati	Ogni regione supportata: 4 MB	No	La dimensione massima consentita per il corpo della richiesta HTTP.
Numero massimo di coppie segrete del cluster simultanee dell'API dati	Ogni regione supportata: 30	No	Il numero massimo di coppie uniche di cluster e segreti DB Aurora Serverless v1 nelle richieste simultanee di Data API per questo account nella regione corrente. AWS

Nome	Predefinita	Adatta e	Descrizione
Numero massimo di richieste simultane e dell'API dati	Ogni regione supportata: 500	No	Il numero massimo di richieste Data API a un cluster DB Aurora Serverless v1 che utilizzano lo stesso segreto e possono essere elaborate contempor aneamente. Le richieste aggiuntive vengono messe in coda ed elaborate al termine delle richieste in corso.
Dimensione massima del set di risultati dell'API dati	Ogni regione supportata: 1 MB	No	La dimensione massima del set di risultati del database che può essere restituito dall'API dati.
Dimensione massima dell'API dati della stringa di risposta JSON	Ogni regione supportata: 10 megabyte	No	La dimensione massima della stringa di risposta JSON semplificata restituita dall'API dati RDS.
Richieste API dati al secondo	Ogni regione supportata: 1.000 al secondo	No	Il numero massimo di richieste all'API Data al secondo consentito per questo account nella regione corrente. AWS Questa quota si applica solo ai cluster Amazon Aurora Serverless v1.

Nome	Predefinita	Adatta e	Descrizione
Abbonamenti a eventi	Ogni regione supportata: 20	Sì	Il numero massimo di sottoscrizioni di eventi
Ruoli IAM per cluster di database	Ogni regione supportata: 5	Sì	Il numero massimo di ruoli IAM associati a un cluster di database
Ruoli IAM per istanza database	Ogni regione supportata: 5	Sì	Il numero massimo di ruoli IAM associati a un'istanza database
Snapshot del cluster di database manuale	Ogni regione supportata: 100	Sì	Il numero massimo di snapshot manuali del cluster di database
Snapshot manuali dell'istanza database	Ogni regione supportata: 100	Sì	Il numero massimo di snapshot manuali di istanza database
Gruppi di opzioni	Ogni regione supportata: 20	Sì	Il numero massimo di gruppi di opzioni
Gruppi di parametri	Ogni regione supportata: 50	Sì	Il numero massimo di gruppi di parametri
Proxy	Ogni regione supportata: 20	Sì	Il numero massimo di proxy consentito in questo account nella regione corrente AWS
Lettura delle repliche per primario	Ogni regione supportata: 15	Sì	Il numero massimo di repliche di lettura per istanza database primario. Questa quota non può essere modificat a per Amazon Aurora.

Nome	Predefinita	Adattabile	Descrizione
Istanze database riservate	Ogni regione supportata: 40	Sì	Il numero massimo di istanze DB riservate consentite in questo account nella regione corrente AWS
Regole per gruppo di sicurezza	Ogni regione supportata: 20	No	Il numero massimo di regole per gruppo di sicurezza DB
Gruppi di sicurezza	Ogni regione supportata: 25	Sì	Il numero massimo di gruppi di sicurezza DB
Gruppi di sicurezza (VPC)	Ogni regione supportata: 5	No	Il numero massimo di gruppi di sicurezza DB per Amazon VPC
Sottoreti per gruppo di sottoreti del database	Ogni regione supportata: 20	No	Il numero massimo di sottoreti per gruppo di sottoreti di database
Tag per risorsa	Ogni regione supportata: 50	No	Il numero massimo di tag per risorsa Amazon RDS
Totale storage per tutte le istanze database	Ogni regione supportata: 100.000 GB	Sì	Lo spazio di archiviazione massimo totale (in GB) per tutte le istanze database di Amazon RDS aggiunte insieme. Questa quota non si applica ad Amazon Aurora, che ha un volume cluster massimo di 128 TiB per ogni cluster DB.

> **Note**
>
> Per impostazione predefinita, puoi avere fino a 40 istanze database. Le istanze database RDS, le istanze database Aurora, le istanze Amazon Neptune e le istanze Amazon DocumentDB si applicano a questa quota.
> Le seguenti limitazioni si applicano alle istanze database di Amazon RDS:
>
> - 10 per ogni versione di SQL Server (Enterprise, Standard, Web ed Express) nel modello "license-included" (licenza inclusa)
> - 10 per Oracle nel modello "license-included" (licenza inclusa)
> - 40 per Db2 secondo il modello di licenza "bring-your-own-license" (BYOL)
> - 40 per MySQL, MariaDB o PostgreSQL
> - 40 per Oracle secondo il modello di licenza "bring-your-own-license" (BYOL)
>
> Se l'applicazione richiede più istanze database, puoi richiedere altre istanze database aprendo la console Service Quotas. Nel pannello di navigazione, scegliere servizi AWS . Scegli Amazon Relational Database Service (Amazon RDS), scegli una quota e segui le istruzioni per richiedere un aumento della quota. Per ulteriori informazioni, consulta Richiesta di un aumento di quota nella Guida per l'utente per Service Quotas.
> In RDS per Oracle e RDS per SQL Server, il limite di replica di lettura è di 5 per database di origine per ogni regione.
> Per informazioni in merito AWS Backup, consulta la Guida per gli sviluppatori AWS Backup .

Se si utilizza una delle operazioni dell'API RDS e si supera la quota predefinita del numero di chiamate al secondo, l'API Amazon RDS emette un errore simile al seguente.

ClientError: Si è verificato un errore (ThrottlingException) durante la chiamata dell'operazione *API_Name*: Rate exceeded.

Qui, riduci il numero di chiamate al secondo. La quota è destinata a coprire la maggior parte dei casi d'uso. Se sono necessarie quote più elevate, puoi richiedere un aumento della quota utilizzando una delle seguenti opzioni:

- Dalla console, apri la console Service Quotas.
- Da AWS CLI, usa il request-service-quota-increase AWS CLI comando.

Per maggiori informazioni, consulta Guida per l'utente di Service Quotas.

Vincoli per la denominazione in Amazon RDS

Nella seguente tabella sono descritti i vincoli per la denominazione in Amazon RDS.

Risorsa o elemento	Vincoli
Identificatore istanze database	Gli identificatori hanno questi vincoli per la denominazione: • Devono contenere da 1 a 63 caratteri alfanumerici o trattini. • Il primo carattere deve essere una lettera. • Non può terminare con un trattino o contenere due trattini consecutivi. • Deve essere unico per tutte le istanze DB per AWS account, per AWS regione.
Nome database del	I vincoli del nome del database differiscono per ogni motore del database . Per ulteriori informazioni, consultare le impostazioni disponibili durante la creazione di ogni di istanza database. **Note** Questo approccio non si applica a SQL Server. In SQL Server, i database vengono creati dopo l'istanza database.
Nome utente master	I vincoli del nome utente master variano in base al motore del database. Per ulteriori informazioni, consultare le impostazioni disponibili durante la creazione di ogni di istanza database.
Master password (Password master)	La password dell'utente master del database può includere qualsiasi carattere ASCII stampabile, tranne /,

Risorsa o elemento	Vincoli
	', ", @ o uno spazio. Per Oracle, & è un'ulteriore limitazione dei caratteri. La password contiene il seguente numero di caratteri ASCII stampabili, a seconda del motore del database: • Db2:8—255 • MariaDB and MySQL: da 8 a 41 • Oracle: da 8 a 30 • SQL Server e PostgreSQL: da 8 a 128
Nome gruppo di parametri database	Questi nomi hanno i seguenti vincoli: • Devono contenere da 1 a 255 caratteri alfanumerici. • Il primo carattere deve essere una lettera. • I trattini sono consentiti, ma il nome non può terminare con un trattino o contenere due trattini consecutivi.
Nome gruppo di sottoreti del database	Questi nomi hanno i seguenti vincoli: • Devono contenere da 1 a 255 caratteri. • Sono consentiti caratteri alfanumerici, spazi, trattini, trattini bassi e punti.

Numero massimo di connessioni di database

Il numero massimo di connessioni di database simultanee varia in base al tipo di motore del database e all'allocazione di memoria per la classe di istanza database. Il numero massimo di connessioni è in genere impostato nel gruppo di parametri associato all'istanza database. L'eccezione è Microsoft SQL Server, dove è impostato nelle proprietà del server per l'istanza database in SQL Server Management Studio (SSMS).

Le connessioni al database consumano memoria. L'impostazione di uno di questi parametri su un valore troppo alto può causare una bassa condizione di memoria che potrebbe causare il passaggio di un'istanza database allo stato parametri incompatibili. Per ulteriori informazioni, consulta Diagnosi e risoluzione dello stato dei parametri incompatibili per un limite di memoria.

Se le applicazioni aprono e chiudono frequentemente connessioni o mantengono un numero elevato di connessioni di lunga durata aperte, ti consigliamo di utilizzare Server proxy per Amazon RDS. Proxy RDS è un proxy di database completamente gestito e ad alta disponibilità che utilizza il pooling di connessioni per condividere connessioni di database in modo sicuro ed efficiente. Per ulteriori informazioni sul Proxy RDS, consulta Utilizzo di Server proxy per Amazon RDS.

> **Note**
>
> Per Oracle, impostare il numero massimo di processi utente e sessioni utente e di sistema. Per Db2, non è possibile impostare il numero massimo di connessioni. Il limite è 64000.

Numero massimo di connessioni di database

Motore database	Parametro	Valori consentiti	Valore predefinito	Descrizione
MariaDB e MySQL	`max_connections`	Da 1 a 100000	Impostazione predefinita per tutte le versioni di MariaDB e MySQL tranne MariaDB versione 10.5 e 10.6: {DB InstanceClassMemory /12582880} Impostazione predefinita per MariaDB versione 10.5 e 10.6: MINIMO ({DB /25165760} ,12000InstanceClassMemory) **Note** In entrambi i casi, se il calcolo del valore predefini	Numero di connessioni client simultanee consentite

Motore database	Parametro	Valori consentiti	Valore predefinito	Descrizione
			to determina un valore maggiore di 16.000, Amazon RDS imposta il limite su 16.000 per le istanze database MariaDB e MySQL.	
Oracle	`processes`	Da 80 a 20000	MINIMO ({DB InstanceClassMemory /9868951}, 2000)	Processi utente
	`sessions`	Da 100 a 65535	–	Sessioni utente e sistema
PostgreSQL	`max_connections`	Da 6 a 8388607	MINIMO ({DB InstanceClassMemory /9531392}, 5000)	Numero massimo di connessioni simultanee
SQL Server	Numero massimo di connessioni simultanee	Da 0 a 32767	0 (illimitate)	Numero massimo di connessioni simultanee

`DBInstanceClassMemory` è in byte. Per informazioni dettagliate su come viene calcolato questo valore, consulta Specifica dei parametri del database. A causa della memoria riservata al sistema operativo e ai processi di gestione RDS, questa dimensione della memoria è inferiore al valore in gibibyte (GiB) mostrato in Specifiche hardware per le classi di istanza database .

Ad esempio, alcune classi di istanze database hanno 8 GiB di memoria, ovvero 8.589.934.592 byte. Per un'istanza database MySQL in esecuzione su una classe di istanza database con 8 GiB di memoria, ad esempio db.m7g.large, l'equazione che utilizza la memoria totale sarebbe `8589934592/12582880=683`. Tuttavia, la variabile `DBInstanceClassMemory` sottrae

automaticamente gli spazi riservati al sistema operativo e ai processi RDS che gestiscono l'istanza. Il risultato della sottrazione viene quindi diviso per 12.582.880. Questo calcolo dà come risultato circa 630 per il valore di `max_connections` anziché 683. Questo valore dipende dalla classe di istanza database e dal motore del database.

Quando un'istanza database MariaDB o MySQL è in esecuzione su una classe di istanza database di piccole dimensioni, come db.t3.micro o db.t3.small, la memoria totale disponibile è scarsa. Per queste classi di istanze database, RDS riserva una parte significativa della memoria disponibile, il che influisce sul valore `max_connections`. Ad esempio, il numero massimo predefinito di connessioni per un'istanza database MySQL in esecuzione su una classe di istanza database db.t3.micro è circa 60. Puoi determinare il valore `max_connections` per l'istanza database MariaDB o MySQL connettendoti ad essa ed eseguendo il seguente comando SQL:

```
SHOW GLOBAL VARIABLES LIKE 'max_connections';
```

Limiti delle dimensioni dei file in Amazon RDS

I limiti di dimensione dei file si applicano a determinate istanze database Amazon RDS. Per ulteriori informazioni, consultare i seguenti limiti specifici del motore:

- Limiti delle dimensioni dei file MariaDB in Amazon RDS
- Limiti delle dimensioni dei file MySQL in Amazon RDS
- Limiti delle dimensioni dei file Oracle in Amazon RDS

Risoluzione dei problemi per Amazon RDS

Utilizza le sezioni seguenti per risolvere i problemi che riscontri con le istanze database in Amazon RDS e Amazon Aurora.

Argomenti

Per informazioni sui problemi di debug con l'API Amazon RDS, consulta Risoluzione dei problemi delle applicazioni in Amazon RDS.

Impossibile connettersi all'istanza database di Amazon RDS

Quando non è possibile connettersi a un'istanza database, le cause comuni sono le seguenti:

- Regole in entrata: le regole di accesso applicate dal firewall locale e gli indirizzi IP autorizzati per accedere all'istanza database potrebbero non corrispondere. Il problema è probabilmente correlato alle regole in entrata del gruppo di sicurezza.

  Per impostazione predefinita, le istanze database non consentono l'accesso. L'accesso viene concesso tramite un gruppo di sicurezza associato al VPC che consente il traffico in entrata e in uscita dall'istanza database. Se necessario, aggiungi regole in entrata e in uscita per la situazione particolare al gruppo di sicurezza. Puoi specificare un indirizzo IP, un intervallo di indirizzi IP o un altro gruppo di sicurezza VPC.

> **Note**
>
> Quando si aggiunge una nuova regola in entrata, puoi scegliere My IP (Il mio IP) per Source (Origine) per consentire l'accesso all'istanza database dall'indirizzo IP rilevato nel browser.

Per ulteriori informazioni sulla configurazione di un gruppo di sicurezza, consulta Fornisci accesso alla istanza database nel VPC creando un gruppo di sicurezza.

> **Note**
>
> Le connessioni client da indirizzi IP all'interno dell'intervallo 169.254.0.0/16 non sono consentite. Si tratta di un intervallo APIPA (Automatic Private IP Addressing), utilizzato per gli indirizzi con collegamenti locali.

- Public accessibility (Accesso pubblico): per connettersi all'istanza database dall'esterno del VPC, ad esempio utilizzando un'applicazione client, occorre assegnare all'istanza un indirizzo IP pubblico.

  Per rendere l'istanza accessibile pubblicamente, modificarla e scegliere Yes (Sì) in Public accessibility (Accesso pubblico). Per ulteriori informazioni, consulta Nascondere istanze database in un VPC da Internet.
- Porta: la porta specificata al momento della creazione dell'istanza database non può essere utilizzata per inviare o ricevere comunicazioni a causa delle restrizioni del tuo firewall locale. Per determinare se la rete consente l'utilizzo della porta specificata per le comunicazioni in entrata e in uscita, consulta il tuo amministratore di rete.
- Disponibilità: per una nuova istanza database creata, lo stato dell'istanza database è `creating` finché non è pronta per essere utilizzata. Quando lo stato cambia in `available`, puoi connetterti all'istanza database. A seconda delle dimensioni dell'istanza di database, potresti dover attendere circa 20 minuti prima che questa diventi disponibile.
- Gateway Internet – Per un'istanza database che deve essere accessibile pubblicamente, le sottoreti nel gruppo di sottoreti del database devono disporre di un gateway Internet.

Per configurare un gateway Internet per una sottorete

1. Accedi AWS Management Console e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nel riquadro di navigazione, scegliere Databases (Database) e selezionare il nome dell'istanza database.
3. Nella scheda Connectivity & security (Connettività e sicurezza) annotare i valori dell'ID VPC in VPC e l'ID della sottorete in Subnets (Sottoreti).
4. Accedere alla console Amazon VPC all'indirizzo https://console.aws.amazon.com/vpc/.
5. Nel riquadro di navigazione, scegliere Internet Gateways. Verificare che al VPC sia associato un Internet gateway. In caso contrario, scegliere Create Internet Gateway (Crea Internet gateway) per crearne uno. Selezionare l'Internet gateway, quindi Attach to VPC (Associa a VPC) e seguire le istruzioni di associazione del gateway al VPC.
6. Nel riquadro di navigazione scegliere Subnets (Sottoreti) e selezionare la sottorete desiderata.
7. Nella scheda Route Table (Tabella di routing) verificare che sia presente un instradamento con `0.0.0.0/0` come destinazione e l'Internet gateway del VPC come target.

   Se si sta effettuando la connessione all'istanza utilizzando il relativo indirizzo IPv6, verificare che sia disponibile un instradamento per tutto il traffico IPv6 (`::/0`) che punti all'Internet gateway. In caso contrario, eseguire le seguenti operazioni:

   a. Selezionare l'ID per la tabella di routing (rtb-xxxxxxxx) per navigare alla tabella di routing.
   b. Nella scheda Routes (Route), scegliere Edit routes (Modifica route). Selezionare Add route (Aggiungi route), utilizzare `0.0.0.0/0` come destinazione e il gateway internet come target.

      Per IPv6, selezionare Add route (Aggiungi route), utilizzare `::/0` come destinazione e il gateway internet come target.
   c. Selezionare Save routes (Salva route).

   Inoltre, se stai tentando di connetterti all'endpoint IPv6, assicurati che l'intervallo di indirizzi IPv6 del client sia autorizzato a connettersi all'istanza database.

Per ulteriori informazioni, consulta Uso di un'istanza database in un VPC.

Per problemi di connessione specifici del motore, consulta i seguenti argomenti:

- Risoluzione dei problemi relativi alle connessioni all'istanza database di SQL Server
- Risoluzione dei problemi relativi alle connessioni all'istanza database Oracle
- Risoluzione dei problemi relativi alle connessioni all'istanza RDS per PostgreSQL
- Numero massimo di connessioni MySQL e MariaDB

Test della connessione a un'istanza database

Puoi verificare la connessione a un'istanza database utilizzando strumenti comuni di Linux o Windows.

Da un terminale Linux o Unix, puoi eseguire il test della connessione immettendo quanto segue. Sostituisci *DB-instance-endpoint* con l'endpoint e *port* con la porta dell'istanza database.

```
nc -zv DB-instance-endpoint port
```

Di seguito è riportato un comando di esempio e il valore restituito.

```
nc -zv postgresql1.c6c8mn7fake0.us-west-2.rds.amazonaws.com 8299

  Connection to postgresql1.c6c8mn7fake0.us-west-2.rds.amazonaws.com 8299 port [tcp/
vvr-data] succeeded!
```

Gli utenti Windows possono utilizzare Telnet per eseguire il test della connessione a un'istanza di database. Le operazioni di Telnet non sono supportate, ad eccezione del test della connessione. Se una connessione ha esito positivo, l'operazione non restituisce alcun messaggio. Se una connessione non va a buon fine, riceverai un messaggio di errore del tipo seguente.

```
C:\>telnet sg-postgresql1.c6c8mntfake0.us-west-2.rds.amazonaws.com 819

  Connecting To sg-postgresql1.c6c8mntfake0.us-west-2.rds.amazonaws.com...Could not
 open
  connection to the host, on port 819: Connect failed
```

Se le operazioni di Telnet hanno esito positivo, il tuo gruppo di sicurezza è configurato correttamente.

> **Note**
>
> Amazon RDS non accetta il traffico del protocollo ICMP (Internet Control Message Protocol), incluso il ping.

Risoluzione di problemi di autenticazione della connessione

In alcuni casi, è possibile connettersi all'istanza database, ma si ricevono errori di autenticazione. In questi casi, potrebbe essere necessario ripristinare la password utente master per l'istanza database. A tale scopo, è necessario modificare l'istanza RDS.

Per ulteriori informazioni sulla modifica di un'istanza database , consulta Modifica di un'istanza database Amazon RDS.

Problemi relativi alla sicurezza di Amazon RDS

Per evitare problemi di sicurezza, non utilizzare mai il nome AWS utente principale e la password per un account utente. È consigliabile utilizzare il master Account AWS per creare utenti e assegnarli agli account utente DB. Puoi anche utilizzare il tuo account principale per creare altri account utente, se necessario.

Per informazioni sulla creazione di utenti, consulta Creazione di un utente IAM nell' Account AWS. Per informazioni sulla creazione di utenti in AWS IAM Identity Center, consulta Gestire le identità in IAM Identity Center.

Messaggio di errore "Impossibile recuperare gli attributi dell'account, alcune funzioni della console potrebbero non essere attive."

Puoi ricevere questo errore per diversi motivi. È possibile che l'account non disponga delle autorizzazioni o che l'account non sia stato configurato correttamente. Se si tratta di un nuovo account, potrebbe essere necessario attendere che l'account sia pronto. Se si tratta di un account esistente, nelle policy di accesso potrebbero non essere disponibili alcune autorizzazioni per eseguire determinate operazioni, come creare un'istanza database. Per risolvere il problema, l'amministratore deve fornire i ruoli necessari per il tuo account. Per ulteriori informazioni, consulta la documentazione di IAM.

Risoluzione dei problemi relativi allo stato di rete non compatibile

Lo stato di rete non compatibile significa che il database potrebbe essere ancora accessibile a livello di database ma non è possibile modificarlo o riavviarlo.

Cause

Lo stato di rete non compatibile dell'istanza database potrebbe essere il risultato di una delle seguenti azioni:

- Modifica della classe dell'istanza database.
- Modifica dell'istanza database per utilizzare l'implementazione multi-AZ del cluster database.
- Sostituzione di un host a causa di un evento di manutenzione.
- Avvio di un'istanza database di sostituzione.
- Ripristino da un backup snapshot.
- Avvio di un'istanza database arrestata.

Risoluzione

Usa il comando start-db-instance

Per correggere un database che si trova in uno stato di rete non compatibile, segui queste istruzioni:

1. Apri https://console.aws.amazon.com/rds/ e scegli Database nel pannello di navigazione.
2. Scegli l'istanza database che si trova nello stato di rete non compatibile e annota l'identificatore dell'istanza database, l'ID del VPC e gli ID di sottorete nella scheda Connettività e sicurezza.
3. Usa il AWS CLI per eseguire il `start-db-instance` comando. Specifica il valore `--db-instance-identifier`.

> **Note**
>
> L'esecuzione di questo comando quando il database è in modalità non compatibile potrebbe causare tempi di inattività.
> Il comando `start-db-instance` non risolve questo problema per le istanze database RDS per SQL Server.

Lo stato del database diventa Disponibile se il comando viene eseguito correttamente.

Se il database si riavvia, l'istanza database potrebbe eseguire l'ultima operazione eseguita sull'istanza prima che tale istanza assumesse lo stato di rete non compatibile. Ciò potrebbe riportare l'istanza allo stato di rete non compatibile.

Se il comando `start-db-instance` ha esito negativo o l'istanza torna allo stato di rete non compatibile, apri la pagina Database nella console RDS e seleziona il database. Passa alla sezione Log ed eventi. Nella sezione Eventi recenti sono visualizzati ulteriori passaggi di risoluzione da seguire. I messaggi sono classificati come segue:

- CONTROLLO DELLE RISORSE INTERNE: potrebbero essersi verificati problemi con le risorse interne.
- CONTROLLO DNS: verifica la risoluzione DNS e i nomi host per il VPC nella console VPC.
- CONTROLLO ENI: l'interfaccia di rete elastica (ENI) per il database potrebbe non esistere.
- CONTROLLO DEL GATEWAY: il gateway Internet per il database disponibile pubblicamente non è collegato al VPC.
- CONTROLLO IP: non ci sono indirizzi IP gratuiti nelle tue sottoreti.
- CONTROLLO DEL GRUPPO DI SICUREZZA: non ci sono gruppi di sicurezza associati al database oppure i gruppi di sicurezza non sono validi.
- CONTROLLO DELLE SOTTORETI: non ci sono sottoreti valide nel tuo gruppo di sottoreti database o ci sono problemi nella tua sottorete.
- CONTROLLO DEL VPC: il VPC associato al database non è valido.

Eseguire point-in-time il ripristino

È consigliabile disporre di un backup (snapshot o logico), nel caso in cui il database passi a uno stato di rete non compatibile. Per informazioni, consulta Introduzione ai backup. Se hai attivato i backup automatici, interrompi temporaneamente le scritture sul database ed esegui un point-in-time ripristino.

> **Note**
>
> Dopo che un'istanza passa allo stato di rete non compatibile, l'istanza database potrebbe non essere accessibile per eseguire un backup logico.

Se non hai attivato i backup automatici, crea una nuova istanza database. Esegui quindi la migrazione dei dati utilizzando AWS Database Migration Service (AWS DMS) o utilizzando uno strumento di backup e ripristino.

Se il problema persiste, contatta AWS Support per ulteriore assistenza.

Reimpostazione della password del ruolo di proprietario dell'istanza di database

Se si viene bloccati dal di istanze DB, è possibile accedere come utente master. È quindi possibile reimpostare le credenziali per altri utenti o ruoli amministrativi. Se non riesci ad accedere come utente principale, il proprietario dell' AWS account può reimpostare la password dell'utente principale. Per informazioni dettagliate sugli account o sui ruoli amministrativi che potrebbe essere necessario reimpostare, vedere Privilegi dell'account utente master.

Puoi modificare la password dell'istanza DB utilizzando la console Amazon RDS, il AWS CLI comando o utilizzando l'modify-db-instanceoperazione API ModifyDBInstance. Per ulteriori informazioni sulla modifica di un'istanza database, consulta Modifica di un'istanza database Amazon RDS.

Errore o riavvio di un'istanza di database Amazon RDS

Un errore dell'istanza database può verificarsi quando viene riavviata. Può anche verificarsi quando l'istanza database viene inserita in uno stato che impedisce l'accesso e quando il database viene riavviato. Un riavvio può verificarsi quando si riavvia manualmente l'istanza database. Un riavvio può anche verificarsi quando si modifica un'impostazione dell'istanza database che richiede un riavvio prima che la modifica diventi effettiva.

Il riavvio di un'istanza database può verificarsi quando si modifica un'impostazione che richiede un riavvio o quando il riavvio viene innescato manualmente. Un riavvio può verificarsi immediatamente se si modifica un'impostazione e si richiede che la modifica abbia effetto immediato. Oppure può verificarsi durante la finestra di manutenzione dell'istanza database.

Un riavvio dell'istanza del database si verifica immediatamente quando si verifica una delle seguenti condizioni:

- Il periodo di retention dei backup per un'istanza database viene modificato da 0 a un valore diverso da zero o da un valore diverso da zero a 0. Quindi si imposta Apply Immediately (Applica immediatamente) su `true`.
- Si modifica la classe di istanza database e si imposta Apply Immediately (Applica immediatamente) su `true`.
- Si modifica il tipo di storage da Magnetico (Standard) a Uso generale (SSD) o da IPOS con provisioning (SSD) oppure da IPOS con provisioning (SSD) o Uso generale (SSD) a Magnetico (Standard).

Un riavvio dell'istanza di database avviene durante la finestra di manutenzione quando si verifica una delle seguenti condizioni:

- Si modifica il periodo di retention dei backup per un'istanza database da 0 a un valore diverso da zero o da un valore diverso da zero a 0 e Apply Immediately (Applica immediatamente) è impostato su `false`.
- Si modifica la classe di istanza database e si imposta Apply Immediately (Applica immediatamente) su `false`.

Quando si modifica un parametro statico in un gruppo di parametri database, la modifica non diventa effettiva fino al riavvio dell'istanza database associata al gruppo di parametri. La modifica richiede un riavvio manuale. L'istanza database non viene riavviata automaticamente durante la finestra di manutenzione.

Per visualizzare una tabella che illustra le operazioni di un'istanza database e l'effetto dell'impostazione del valore Apply Immediately (Applica immediatamente), consulta Modifica di un'istanza database Amazon RDS.

Modifiche ai parametri di database Amazon RDS che non hanno effetto

In alcuni casi, si potrebbe modificare un parametro in un gruppo di parametri database senza che le modifiche diventino effettive. In tal caso, è probabile che sia necessario riavviare l'istanza database associata al gruppo di parametri database. Quando si modifica un parametro dinamico, la modifica diventa immediatamente effettiva. Quando si modifica un parametro statico, la modifica non diventa effettiva finché l'istanza database associata al gruppo di parametri non viene riavviata.

Puoi riavviare un'istanza database utilizzando la console RDS. In alternativa puoi chiamare esplicitamente l'operazione API `RebootDBInstance`. È possibile riavviare senza failover se l'istanza database è in un'implementazione Multi-AZ. La necessità di riavviare l'istanza database associata dopo la modifica di un parametro statico consente di ridurre il rischio di errore di configurazione di un parametro che influenza una chiamata API. Un esempio è la chiamata di `ModifyDBInstance` per modificare la classe di istanza database. Per ulteriori informazioni, consulta Modifica di parametri in un gruppo di parametri del database.

Mancanza di spazio di storage per l'istanza di database Amazon RDS

Se l'istanza di database non dispone di spazio di storage sufficiente, potrebbe non essere più disponibile. Ti consigliamo vivamente di monitorare costantemente la `FreeStorageSpace` metrica pubblicata in CloudWatch per assicurarti che la tua istanza DB abbia abbastanza spazio di archiviazione libero.

Se la capacità di storage dell'istanza database si esaurisce, il suo stato cambia in `storage-full`. Di seguito è riportato, ad esempio, l'output di una chiamata all'operazione API `DescribeDBInstances` per un'istanza database che ha esaurito lo spazio di storage.

```
aws rds describe-db-instances --db-instance-identifier mydbinstance

DBINSTANCE  mydbinstance  2009-12-22T23:06:11.915Z  db.m5.large  mysql8.0  50  sa
storage-full  mydbinstance.clla4j4jgyph.us-east-1.rds.amazonaws.com  3306
us-east-1b  3
 SECGROUP  default  active
 PARAMGRP  default.mysql8.0  in-sync
```

Per eseguire il ripristino da questo scenario, aggiungi altro spazio di archiviazione all'istanza utilizzando l'operazione `ModifyDBInstance` API o il AWS CLI comando seguente.

Per LinuxmacOS, oUnix:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --allocated-storage 60 \
    --apply-immediately
```

Per Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --allocated-storage 60 ^
    --apply-immediately
```

```
DBINSTANCE  mydbinstance  2009-12-22T23:06:11.915Z  db.m5.large  mysql8.0  50  sa
storage-full  mydbinstance.clla4j4jgyph.us-east-1.rds.amazonaws.com  3306
us-east-1b  3  60
 SECGROUP  default  active
 PARAMGRP  default.mysql8.0  in-sync
```

Ora, l'istanza database che descrivi si trova nello stato `modifying`, il che significa che è in corso il dimensionamento dello storage.

```
aws rds describe-db-instances --db-instance-identifier mydbinstance
```

```
DBINSTANCE  mydbinstance  2009-12-22T23:06:11.915Z  db.m5.large  mysql8.0  50  sa
modifying  mydbinstance.clla4j4jgyph.us-east-1.rds.amazonaws.com
3306  us-east-1b  3  60
 SECGROUP  default  active
 PARAMGRP  default.mysql8.0  in-sync
```

Al termine del dimensionamento dello storage, lo stato dell'istanza database cambia in `available`.

```
aws rds describe-db-instances --db-instance-identifier mydbinstance
```

```
DBINSTANCE  mydbinstance  2009-12-22T23:06:11.915Z  db.m5.large  mysql8.0  60  sa
available  mydbinstance.clla4j4jgyph.us-east-1.rds.amazonaws.com  3306
us-east-1b  3
 SECGROUP  default  active
 PARAMGRP  default.mysql8.0  in-sync
```

Puoi continuare a ricevere notifiche quando lo spazio di storage disponibile è esaurito utilizzando l'operazione `DescribeEvents`. Ad esempio, in questo scenario, se effettui una chiamata `DescribeEvents` dopo queste operazioni, viene visualizzato il seguente risultato:

```
aws rds describe-events --source-type db-instance --source-identifier mydbinstance
```

```
2009-12-22T23:44:14.374Z  mydbinstance  Allocated storage has been exhausted db-
instance
2009-12-23T00:14:02.737Z  mydbinstance  Applying modification to allocated storage db-
instance
2009-12-23T00:31:54.764Z  mydbinstance  Finished applying modification to allocated
 storage
```

Capacità insufficiente dell'istanza di database Amazon RDS

L'errore `InsufficientDBInstanceCapacity` può essere restituito quando si tenta di creare, avviare o modificare un'istanza database. Può anche essere restituito quando si tenta di ripristinare un'istanza database da uno snapshot DB. Quando viene restituito questo errore, una causa comune è che la classe dell'istanza database specifica non è disponibile nella zona di disponibilità richiesta. Puoi provare una delle seguenti soluzioni per risolvere il problema:

- Ritenta la richiesta con una classe di istanza database differente.
- Ritenta la richiesta con una zona di disponibilità differente.
- Ritenta la richiesta senza specificare una zona di disponibilità esplicita.

Per ulteriori informazioni sulla risoluzione dei problemi di capacità delle istanze per Amazon EC2, consulta Capacità dell'istanza insufficiente nella Guida per l'utente Amazon EC2.

Per ulteriori informazioni sulla modifica di un'istanza database, consulta Modifica di un'istanza database Amazon RDS.

Problemi di memoria liberabile in Amazon RDS

La memoria liberabile è la memoria RAM (Random Access Memory) su un'istanza database che può essere resa disponibile al motore di database. È la somma della memoria libera del sistema operativo (OS) e della memoria disponibile del buffer e della cache delle pagine. Il motore di database utilizza la maggior parte della memoria sull'host, ma anche i processi del sistema operativo utilizzano RAM. La memoria attualmente allocata al motore di database o utilizzata dai processi del sistema operativo non è inclusa nella memoria liberabile. Quando il motore di database sta per esaurire la memoria, l'istanza database può utilizzare lo spazio temporaneo normalmente utilizzato per il buffering e la memorizzazione nella cache. Come accennato in precedenza, questo spazio temporaneo è incluso nella memoria liberabile.

Utilizzi la `FreeableMemory` metrica di Amazon CloudWatch per monitorare la memoria disponibile. Per ulteriori informazioni, consulta Panoramica del monitoraggio dei parametri di Amazon RDS.

Se la memoria liberabile dell'istanza database diventa insufficiente o viene utilizzato uno spazio di scambio, valutare l'aumento a una classe di istanza database più grande. Per ulteriori informazioni, consulta Classi di istanze database.

Inoltre, puoi modificare le impostazioni della memoria. Ad esempio, in RDS per MySQL, puoi regolare le dimensioni del parametro `innodb_buffer_pool_size`. Questo parametro è impostato per default sul 75% della memoria fisica. Per ulteriori suggerimenti per la risoluzione di problemi di MySQL, consulta Come è possibile risolvere i problemi di memoria liberabile insufficiente in un database Amazon RDS per MySQL?

Problemi relativi a MySQL e MariaDB

Puoi diagnosticare e risolvere i problemi relativi alle istanze database MySQL e MariaDB.

Argomenti

- Numero massimo di connessioni MySQL e MariaDB
- Diagnosi e risoluzione dello stato dei parametri incompatibili per un limite di memoria
- Diagnosi e risoluzione del ritardo tra repliche di lettura
- Diagnosi e risoluzione di un errore relativo alla replica di lettura MySQL o MariaDB
- La creazione di trigger con log binario abilitato richiede i privilegi SUPER
- Diagnosi e risoluzione degli errori point-in-time di ripristino
- Errore di replica interrotta
- Creazione della replica di lettura non riuscita o interruzione della replica in seguito a errore irreversibile 1236

Numero massimo di connessioni MySQL e MariaDB

Il numero massimo di connessioni consentite a un'istanza database di RDS for MySQL o RDS for MariaDB si basa sulla quantità di memoria disponibile per la relativa classe dell'istanza database. Una classe di istanza database con più memoria disponibile permetterà un maggior numero di connessioni. Per ulteriori informazioni sulle classi di istanza database, consulta Classi di istanze database.

Il limite di connessioni per un'istanza database è impostata per default sul valore massimo per la classe dell'istanza database. Puoi limitare il numero di connessioni simultanee a un qualsiasi valore fino al numero massimo di connessioni consentite. Utilizza il parametro `max_connections` nel gruppo di parametri per l'istanza database. Per ulteriori informazioni, consulta Numero massimo di connessioni di database e Utilizzo di gruppi di parametri.

Puoi recuperare il numero massimo di connessioni consentite per un'istanza database MySQL o MariaDB eseguendo la query seguente.

```
SELECT @@max_connections;
```

Puoi recuperare il numero di connessioni attive a un'istanza database MySQL o MariaDB eseguendo la query seguente.

```
SHOW STATUS WHERE `variable_name` = 'Threads_connected';
```

Diagnosi e risoluzione dello stato dei parametri incompatibili per un limite di memoria

Un'istanza database MariaDB o MySQL può essere impostata sullo stato parametri incompatibili per un limite di memoria quando sono soddisfatte entrambe le seguenti condizioni:

- L'istanza database viene riavviata almeno tre volte in un'ora o almeno cinque volte in un giorno quando lo stato dell'istanza database Disponibile.
- Un tentativo di riavvio dell'istanza database non riesce perché un'operazione di manutenzione o un processo di monitoraggio non sono riusciti a riavviare l'istanza database.
- Il potenziale utilizzo della memoria dell'istanza database supera 1,2 volte la memoria allocata alla classe di istanza database.

Quando un'istanza database viene riavviata per la terza volta in un'ora o per la quinta volta in un giorno, viene eseguito un controllo dell'utilizzo della memoria. Il controllo calcola il potenziale utilizzo della memoria dell'istanza database. Il valore restituito dal calcolo è la somma dei seguenti valori:

- Valore 1 – La somma dei seguenti parametri:
  - `innodb_additional_mem_pool_size`
  - `innodb_buffer_pool_size`

Puoi modificare il valore per. `innodb_buffer_pool_size` Tuttavia, il valore non corrisponderà sempre a quello immesso. Questa mancata corrispondenza si verifica per diversi motivi. Innanzitutto, se l'istanza DB è un'istanza micro DB, sovrascriviamo il valore predefinito e lo impostiamo su 256 MB. Per ulteriori informazioni, consulta Sovrascrivere innodb_buffer_pool_size.

In secondo luogo, ci assicuriamo che 500 MB di memoria siano riservati sull'istanza DB per l'host manager, il motore, il sistema operativo e il kernel.

Infine, ottimizziamo `innodb_buffer_pool_size` dividendola in unità. L'host manager arrotonda per difetto al multiplo più vicino di tali unità. Le unità vengono calcolate `innodb_buffer_pool_chunk_size` moltiplicando per. `innodb_buffer_pool_instances` Per ulteriori informazioni, vedere Configurazione della dimensione del pool di buffer di InnoDB nella documentazione di MySQL.

L'impostazione predefinita `innodb_buffer_pool_instances` è 8, a meno che non `innodb_buffer_pool_size` sia inferiore a 1 GB. Se `innodb_buffer_pool_size` è inferiore a 1 GB, l'impostazione predefinita per `innodb_buffer_pool_instances` è 1. L'impostazione predefinita per `innodb_buffer_pool_chunk_size` è 128 MB.

- `innodb_log_buffer_size`
- `key_buffer_size`
- `query_cache_size` (solo MySQL versione 5.7)
- `tmp_table_size`

- Valore 2 – Il parametro `max_connections` moltiplicato per la somma dei seguenti parametri:
  - `binlog_cache_size`
  - `join_buffer_size`
  - `read_buffer_size`
  - `read_rnd_buffer_size`
  - `sort_buffer_size`
  - `thread_stack`
- Valore 3 – Se il parametro `performance_schema` è abilitato, moltiplicare il parametro `max_connections` per 429498.

  Se invece il parametro `performance_schema` è disabilitato, questo valore è zero.

Quindi, il valore restituito dal calcolo è il seguente:

```
Value 1 + Value 2 + Value 3
```

Quando questo valore supera 1,2 volte la memoria allocata alla classe di istanza database utilizzata dall'istanza database, l'istanza database viene posizionata nello stato dei parametri incompatibili . Per informazioni sulla memoria allocata alle classi di istanza database, consulta Specifiche hardware per le classi di istanza database .

Il calcolo moltiplica il valore del parametro `max_connections` per la somma di diversi parametri. Se il parametro `max_connections` è impostato su un valore elevato, è possibile che il controllo restituisca un valore eccessivamente elevato per il potenziale utilizzo della memoria dell'istanza database. In questo caso, considera la riduzione del valore del parametro `max_connections`.

Per risolvere il problema, completa la seguente procedura:

1. Regola i parametri di memoria nel gruppo di parametri database associato all'istanza database. Esegui questa operazione in modo che il potenziale di utilizzo della memoria sia inferiore a 1,2 volte la memoria allocata alla classe di istanza database

   Per informazioni sull'estensione dei parametri consulta Modifica di parametri in un gruppo di parametri del database.
2. Riavviare l'istanza database.

   Per informazioni sull'estensione dei parametri consulta Avvio di un'istanza database Amazon RDS arrestata in precedenza.

Diagnosi e risoluzione del ritardo tra repliche di lettura

Quando una replica di lettura MySQL or MariaDB viene creata ed è disponibile, Amazon RDS esegue per prima cosa la replica delle modifiche apportate all'istanza database di origine dal momento in cui l'operazione di creazione della replica di lettura è stata avviata. Durante questa fase, il ritardo di replica per la replica di lettura è maggiore di 0. Puoi monitorare questo ritardo in Amazon CloudWatch visualizzando la metrica Amazon RDS. `ReplicaLag`

Il parametro `ReplicaLag` indica il valore del campo `Seconds_Behind_Master` del comando `SHOW REPLICA STATUS` MySQL o MariaDB. Per ulteriori informazioni, consulta Istruzione SHOW REPLICA STATUS nella documentazione di MySQL.

Quando il parametro `ReplicaLag` è 0, la replica ha raggiunto l'istanza del database di origine. Se il parametro `ReplicaLag` restituisce -1, la replica potrebbe non essere attiva. Per la risoluzione di problemi relativi a un errore di replica, consulta Diagnosi e risoluzione di un errore relativo alla replica di lettura MySQL o MariaDB. Un valore di `ReplicaLag` pari a -1 può anche indicare che il valore di `Seconds_Behind_Master` non può essere determinato oppure è `NULL`.

> **Note**
>
> Le versioni precedenti di MariaDB e MySQL utilizzavano `SHOW SLAVE STATUS` anziché `SHOW REPLICA STATUS`. Se si utilizza una versione di MariaDB precedente alla 10.5 o una versione di MySQL precedente alla 8.0.23, utilizzare `SHOW SLAVE STATUS`.

Il parametro `ReplicaLag` restituisce -1 durante un'interruzione della rete o quando viene applicata una patch durante la finestra di manutenzione. In questo caso, attendi fino al ripristino della connettività di rete o al termine della finestra di manutenzione prima di controllare nuovamente il parametro `ReplicaLag` .

La tecnologia di replica di lettura di MySQL e MariaDB è asincrona. Pertanto, è possibile che si verifichino incrementi occasionali del parametro `BinLogDiskUsage` nell'istanza database di origine e del parametro `ReplicaLag` nella replica di lettura. Ad esempio, considera una situazione in cui si verifica un elevato volume di operazioni di scrittura nell'istanza database di origine in parallelo. Contemporaneamente, le operazioni di scrittura nella replica di lettura vengono serializzate utilizzando un singolo thread I/O. Tale situazione può portare a un ritardo tra l'istanza di origine e la replica di lettura.

Per ulteriori informazioni sulle repliche di lettura e su MySQL, consulta la pagina dei dettagli dell'implementazione di repliche nella documentazione di MySQL. Per ulteriori informazioni sulle repliche di lettura e su MariaDB, consulta la pagina della panoramica della replica nella documentazione di MariaDB.

Puoi ridurre il ritardo tra gli aggiornamenti di un'istanza database di origine e i successivi aggiornamenti della replica di lettura attenendoti alla seguente procedura:

- Imposta la classe dell'istanza database della replica di lettura in modo che le sue dimensioni di storage siano paragonabili a quelle dell'istanza del database di origine.
- Assicurati che le impostazioni dei parametri dei gruppi di parametri database utilizzati dall'istanza database di origine e la replica di lettura siano compatibili. Per ulteriori informazioni e un esempio, consulta la discussione sul parametro `max_allowed_packet` nella sezione seguente.

- Disabilita la cache delle query. Per le tabelle modificate di frequente, l'uso della cache delle query può aumentare il ritardo della replica perché la cache viene bloccata e aggiornata spesso. In questo caso, potresti visualizzare un ritardo della replica inferiore se disabiliti la cache delle query. Puoi disabilitare la cache delle query impostando il parametro `query_cache_type parameter` sul valore 0 nel gruppo di parametri di database dell'istanza di database. Per ulteriori informazioni sulla cache delle query, consulta la sezione relativa alla configurazione della cache delle query.
- Riscaldare il buffer pool sulla replica di lettura per InnoDB per MySQL o MariaDB. Ad esempio, supponi di disporre di un set ridotto di tabelle che vengono aggiornate di frequente e di utilizzare lo schema di tabella InnoDB o XtraDB. In questo caso, esegui il dump di tali tabelle nella replica di lettura In questo modo, il motore di database esegue la scansione delle righe delle tabelle del disco e le memorizza nella cache nel pool di buffer, il che può ridurre il ritardo della replica. Di seguito viene riportato un esempio.

  PerLinux, o: macOS Unix

```
PROMPT> mysqldump \
    -h <endpoint> \
    --port=<port> \
    -u=<username> \
    -p <password> \
    database_name table1 table2 > /dev/null
```

  Per Windows:

```
PROMPT> mysqldump ^
    -h <endpoint> ^
    --port=<port> ^
    -u=<username> ^
    -p <password> ^
    database_name table1 table2 > /dev/null
```

Diagnosi e risoluzione di un errore relativo alla replica di lettura MySQL o MariaDB

Amazon RDS monitora lo stato delle repliche di lettura. RDS aggiorna il campo Replication State (Stato di replica) dell'istanza della replica di lettura con il valore `Error`, se la replica viene arrestata per qualsiasi motivo. Puoi rivedere i dettagli dell'errore associato generato dai motori MySQL o

MariaDB visualizzando il campo Replication Error (Errore di replica). Vengono generati anche eventi che indicano lo stato della replica di lettura, inclusi RDS-EVENT-0045, RDS-EVENT-0046 e RDS-EVENT-0057. Per ulteriori informazioni sugli eventi e sulla sottoscrizione a essi, consulta Utilizzo della notifica degli eventi di Amazon RDS. Se viene restituito un messaggio di errore MySQL, controlla l'errore nella documentazione dei messaggi di errore MySQL. Se viene restituito un messaggio di errore MariaDB, controlla l'errore nella documentazione dei messaggi di errore MariaDB.

Situazioni comuni che possono causare errori di replica sono:

- Il valore del parametro `max_allowed_packet` di una replica di lettura è inferiore al parametro `max_allowed_packet` dell'istanza database di origine.

  Il parametro `max_allowed_packet` è un parametro personalizzato che puoi impostare in un gruppo di parametri database. Il parametro `max_allowed_packet` viene utilizzato per specificare la dimensione massima delle query DML (Data Manipulation Language) che possono essere eseguite sul database. In alcuni casi, il valore `max_allowed_packet` dell'istanza database di origine potrebbe essere più grande del valore `max_allowed_packet` per la replica di lettura. In questo caso, il processo di replica può generare un errore e interrompere la replica. L'errore più comune è `packet bigger than 'max_allowed_packet' bytes`. Puoi correggere questo errore impostando l'origine e la replica di lettura in modo che utilizzino i gruppi di parametri database con gli stessi valori del parametro `max_allowed_packet`.
- Scrittura in tabelle su una replica di lettura. Se crei indici su una replica di lettura, il parametro `read_only` deve essere impostato su 0 affinché gli indici vengano creati. Se esegui la scrittura su tabelle sulla replica di lettura, ciò può interrompere la replica.
- Uso di un motore di storage non transazionale come MyISAM. Le repliche di lettura richiedono un motore di storage transazionale. La replica è supportata solo per i seguenti motori di archiviazione: InnoDB per MySQL o MariaDB.

  Puoi convertire una tabella MyISAM in InnoDB, utilizzando il comando seguente:

  ```
  alter table <schema>.<table_name> engine=innodb;
  ```
- Utilizzo di query non deterministiche non sicure come `SYSDATE()`. Per ulteriori informazioni, consulta la pagina relativa alla determinazione delle istruzioni sicure e non sicure nel logging binario nella documentazione MySQL.

La seguente procedura può essere di aiuto per la risoluzione dell'errore di replica:

- Se riscontri un errore logico e puoi ignorarlo in modo sicuro, attieniti alla procedura descritta in Ignorare l'errore di replica corrente. L'istanza di database MySQL o MariaDB deve eseguire una versione che includa la procedura `mysql_rds_skip_repl_error`. Per ulteriori informazioni, consulta mysql.rds_skip_repl_error.
- Se si verifica un problema di posizione del log binario (binlog), puoi modificare la posizione di riproduzione della replica con il comando `mysql_rds_next_master_log`. L'istanza database MySQL o MariaDB deve eseguire una versione che supporti il comando `mysql_rds_next_master_log` per modificare la posizione di riproduzione della replica. Per informazioni sulla versione, consulta mysql.rds_next_master_log.
- Si potrebbe verificare un problema temporaneo a livello di prestazioni dovuto a un elevato carico DML. In tal caso, puoi impostare il parametro `innodb_flush_log_at_trx_commit` su 2 nel gruppo di parametri database sulla replica di lettura. Ciò può agevolare il recupero delle prestazioni della replica di lettura, sebbene le proprietà ACID (atomicità, consistenza, isolamento e durata) subiscano una riduzione temporanea.
- Puoi eliminare la replica di lettura e creare un'istanza utilizzando lo stesso identificatore istanze DB. In questo modo, l'endpoint rimane identico a quello della vecchia replica di lettura.

Quando un problema relativo alla replica viene risolto, il campo Replication State (Stato di replica) cambia in replicating (replica in corso). Per ulteriori informazioni, consulta Risoluzione dei problemi relativi a una replica di lettura MySQL.

La creazione di trigger con log binario abilitato richiede i privilegi SUPER

Durante la creazione di trigger in un'istanza database RDS for MySQL o RDS for MariaDB, potresti ricevere il seguente errore.

```
"You do not have the SUPER privilege and binary logging is enabled"
```

L'utilizzo di trigger quando il log binario è abilitato richiede i privilegi SUPER, che sono soggetti a restrizioni per le istanze database RDS for MySQL e RDS for MariaDB. Puoi creare trigger quando il log binario è abilitato senza privilegi SUPER impostando il parametro `log_bin_trust_function_creators` su true. Per impostare il parametro `log_bin_trust_function_creators` su true, crea un gruppo di parametri di database o modificane uno esistente.

Puoi creare un nuovo gruppo di parametri database per poter creare trigger nell'istanza database RDS per MySQL o RDS per MariaDB con la registrazione binaria abilitata. A tale scopo, utilizza i seguenti comandi CLI. Per modificare un gruppo di parametri esistente, inizia dalla fase 2.

Per creare un nuovo gruppo di parametri per consentire trigger con log binario abilitato utilizzando CLI

1. Crea un nuovo set di parametri.

   Per LinuxmacOS, oUnix:

   ```
   aws rds create-db-parameter-group \
       --db-parameter-group-name allow-triggers \
       --db-parameter-group-family mysql8.0 \
       --description "parameter group allowing triggers"
   ```

   Per Windows:

   ```
   aws rds create-db-parameter-group ^
       --db-parameter-group-name allow-triggers ^
       --db-parameter-group-family mysql8.0 ^
       --description "parameter group allowing triggers"
   ```

2. Modifica il gruppo di parametri di database per consentire i trigger.

   Per LinuxmacOS, oUnix:

   ```
   aws rds modify-db-parameter-group \
       --db-parameter-group-name allow-triggers \
       --parameters "ParameterName=log_bin_trust_function_creators,
    ParameterValue=true, ApplyMethod=pending-reboot"
   ```

   Per Windows:

   ```
   aws rds modify-db-parameter-group ^
       --db-parameter-group-name allow-triggers ^
       --parameters "ParameterName=log_bin_trust_function_creators,
    ParameterValue=true, ApplyMethod=pending-reboot"
   ```

3. Modifica l'istanza di database per utilizzare il nuovo gruppo di parametri di database.

Per LinuxmacOS, oUnix:

```
aws rds modify-db-instance \
    --db-instance-identifier mydbinstance \
    --db-parameter-group-name allow-triggers \
    --apply-immediately
```

Per Windows:

```
aws rds modify-db-instance ^
    --db-instance-identifier mydbinstance ^
    --db-parameter-group-name allow-triggers ^
    --apply-immediately
```

4. Per rendere effettive le modifiche, riavvia manualmente l'istanza database.

```
aws rds reboot-db-instance --db-instance-identifier mydbinstance
```

Diagnosi e risoluzione degli errori point-in-time di ripristino

Ripristino di un'istanza di database che include tabelle temporanee

Quando tenti di point-in-time ripristinare (PITR) la tua istanza MySQL o MariaDB, potresti riscontrare il seguente errore.

```
Database instance could not be restored because there has been incompatible database
 activity for restore
functionality. Common examples of incompatible activity include using temporary tables,
 in-memory tables,
or using MyISAM tables. In this case, use of Temporary table was detected.
```

Il ripristino PITR si basa su snapshot di backup e log binari (binlog) di MySQL o MariaDB per eseguire il ripristino di un'istanza database a un momento specifico. Le informazioni nelle tabelle temporanee potrebbero non essere affidabili nei binlog e causare un errore di ripristino PITR. Se utilizzi tabelle temporanee nell'istanza database MySQL o MariaDB, puoi ridurre la possibilità di un errore PITR. A questo scopo, esegui backup più frequenti. È più probabile che tale errore si verifichi tra la creazione di una tabella temporanea e il successivo snapshot di backup.

Ripristino di un'istanza di database che include tabelle in memoria

È possibile che si verifichi un problema durante il ripristino di un database con tabelle in memoria. Il contenuto delle tabelle in memoria viene rimosso durante il riavvio. Pertanto, le tabelle in memoria potrebbero risultare vuote dopo un riavvio. Quando utilizzi tabelle in memoria, è consigliabile progettare l'architettura della soluzione in modo che gestisca le tabelle vuote in caso di riavvio. Se utilizzi tabelle in memoria con istanze database replicate, potrebbe essere necessario ricreare le repliche di lettura dopo un riavvio. Ciò potrebbe essere necessario se una replica di lettura viene riavviata e non è in grado di ripristinare i dati da una tabella in memoria vuota.

Per ulteriori informazioni sui backup e sul ripristino PITR, consulta Introduzione ai backup e Ripristino a un'ora specifica per un'istanza database.

Errore di replica interrotta

Quando si chiama il comando `mysql.rds_skip_repl_error`, è possibile che venga visualizzato un messaggio di errore che indica che la replica è inattiva o disattivata.

Questo messaggio di errore viene visualizzato perché la replica è stata arrestata e non può essere riavviata.

Se devi ignorare un numero elevato di errori, il ritardo della replica potrebbe superare il periodo di retention predefinito per i file di log binari. In questo caso, potresti riscontrare un errore irreversibile a causa di file di log binari che sono stati eliminati prima di essere riprodotti sulla replica. Questa eliminazione causa l'arresto della replica e non è più possibile chiamare il comando `mysql.rds_skip_repl_error` per ignorare errori di replica.

Puoi limitare questo problema aumentando il numero di ore di retention dei file di log binari nella sorgente di replica. Una volta aumentato il tempo di retention dei file binlog, puoi riavviare la replica e chiamare il comando `mysql.rds_skip_repl_error` secondo necessità.

Per impostare il periodo di retention dei file binlog, utilizza la procedura mysql.rds_set_configuration. Specifica un parametro di configurazione di "binlog retention hours" (ore di retention dei file binlog) insieme al numero di ore di retention dei file binlog nel cluster di database, fino a 720 (30 giorni). Nell'esempio seguente il periodo di retention dei file binlog è impostato su 48 ore.

```
CALL mysql.rds_set_configuration('binlog retention hours', 48);
```

Creazione della replica di lettura non riuscita o interruzione della replica in seguito a errore irreversibile 1236

Dopo aver modificato i valori dei parametri predefiniti per un'istanza di database MySQL o MariaDB, potresti riscontrare uno dei seguenti problemi:

- Non è possibile creare una replica di lettura per l'istanza database.
- La replica non riesce e viene visualizzato `fatal error 1236`.

Alcuni valori di parametri predefiniti per le istanze database MySQL o MariaDB garantiscono che il database sia conforme ad ACID e che le repliche di lettura siano protette da arresti anomali. A questo scopo viene fatto in modo che ogni commit sia completamente sincronizzato mediante la scrittura della transazione nel log binario prima dell'esecuzione del commit. La modifica di questi parametri rispetto ai valori predefiniti per migliorare le prestazioni può causare errori di replica quando una transazione non è stata scritta nel log binario.

Per risolvere questo problema, utilizza i seguenti valori di parametri:

- `sync_binlog = 1`
- `innodb_support_xa = 1`
- `innodb_flush_log_at_trx_commit = 1`

Impossibile impostare il periodo di retention dei backup su 0

Esistono diversi i motivi per cui potrebbe essere necessario impostare il periodo di retention dei backup su 0. Ad esempio, puoi disabilitare immediatamente i backup automatici impostando il periodo di retention su 0.

In alcuni casi, potrebbe essere necessario impostare il valore su 0 e ricevere un messaggio che indica che il periodo di retention deve essere compreso tra 1 e 35. In questi casi, verificare di non aver impostato una replica di lettura per l'istanza. Le repliche di lettura richiedono i backup per gestire i log delle repliche di lettura e pertanto non puoi impostare un periodo di retention pari a 0.

Documentazione di riferimento dell'API Amazon RDS

Oltre alla AWS Management Console e AWS Command Line Interface (AWS CLI), Amazon RDS fornisce anche un'API. È possibile usare l'API per automatizzare le attività per la gestione delle istanze database e altri oggetti in Amazon RDS.

- Per un elenco alfabetico delle operazioni API, consulta Operazioni.
- Per un elenco alfabetico dei tipi di dati, consulta la pagina Tipi di dati.
- Per un elenco di parametri di query comuni, consulta la pagina Parametri Comuni.
- Per le descrizioni dei codici di errore, consulta la pagina Errori comuni.

Per ulteriori informazioni sui AWS CLI, consulta Riferimento AWS Command Line Interface per Amazon RDS.

Argomenti

Uso dell'API query

Le sezioni seguenti illustrano brevemente le autenticazioni dei parametri e delle richieste usate con l'API query.

Per informazioni generali sul funzionamento dell'API Query, consulta Richieste di query in Amazon EC2 API Reference.

Parametri di query

Le richieste basate su query HTTP sono richieste HTTP che utilizzano i verbi HTTP GET oppure POST e un parametro di query denominato `Action`.

Ogni richiesta di query deve includere alcuni parametri comuni per gestire l'autenticazione e la selezione di un'azione.

Alcune operazioni accettano elenchi di parametri. Questi elenchi sono specificati usando l'annotazione `param.`*n*. I valori di *n* sono numero a partire da 1.

Per ulteriori informazioni su endpoint e regioni Amazon RDS, consulta la pagina relativa ad Amazon Relational Database Service (RDS) nella sezione relativa a regioni ed endpoint della Riferimenti generali di Amazon Web Services.

Autenticazione delle richieste di query

È possibile inviare solo richieste di query tramite HTTPS ed è necessario includere una firma in ogni richiesta di query. È necessario usare AWS Signature Version 4 o Signature Version 2. Per ulteriori informazioni, consulta Processo di firma Signature Version 4 e Processo di firma Signature Version 2.

Risoluzione dei problemi delle applicazioni in Amazon RDS

Amazon RDS fornisce errori specifici e descrittivi per aiutarti a risolvere i problemi mentre interagisci con l'API Amazon RDS.

Argomenti

- Errore durante il recupero
- Suggerimenti per la risoluzione dei problemi

Per informazioni sulla risoluzione dei problemi per le istanze database Amazon RDS, consulta Risoluzione dei problemi per Amazon RDS.

Errore durante il recupero

In genere, si desidera che l'applicazione verifichi se una richiesta ha generato un errore prima di trascorrere del tempo a elaborare i risultati. Il modo più semplice per determinare se si è verificato un errore consiste nel cercare un nodo `Error` nella risposta dall'API Amazon RDS.

La sintassi XPath fornisce un modo semplice per rilevare la presenza di un nodo `Error`. Fornisce inoltre un modo relativamente semplice per recuperare il messaggio e il codice di errore. Il seguente snippet di codice usa Perl e il modulo XML::XPath per determinare se si è verificato un errore durante una richiesta. Se si è verificato un errore, il codice stampa il primo codice di errore e il messaggio nella risposta.

```
use XML::XPath;
    my $xp = XML::XPath->new(xml =>$response);
    if ( $xp->find("//Error") )
```

```
    {print "There was an error processing your request:\n", " Error code: ",
    $xp->findvalue("//Error[1]/Code"), "\n", " ",
    $xp->findvalue("//Error[1]/Message"), "\n\n"; }
```

Suggerimenti per la risoluzione dei problemi

Ti consigliamo i seguenti processi per diagnosticare e risolvere i problemi con l'API Amazon RDS:

- Verifica che Amazon RDS funzioni normalmente nella Regione AWS di destinazione visitando l'indirizzo http://status.aws.amazon.com.
- Verificare la struttura della richiesta.

  Ogni operazione Amazon RDS ha una pagina di riferimento nella documentazione di riferimento dell'API Amazon RDS. Controllare nuovamente che si stia usando i parametri correttamente. Per le idee su cosa potrebbe essere sbagliato, guarda le richieste di esempio o gli scenari utente per vedere se quegli esempi contengono operazioni simili.
- Controlla AWS re:Post.

  Amazon RDS dispone di una community di sviluppo in cui puoi cercare soluzioni ai problemi che altri hanno riscontrato lungo il percorso. Per vedere gli argomenti, consulta AWS re:Post.

Cronologia dei documenti

Versione corrente dell'API: 2014-10-31

La tabella seguente descrive le modifiche importanti introdotte in ogni versione della Guida per l'utente di Amazon RDS dopo maggio 2018. Per ricevere notifiche sugli aggiornamenti di questa documentazione, puoi abbonarti a un feed RSS.

> **Note**
>
> Please change to "Puoi filtrare le nuove funzionalità Amazon RDS alla pagina Quali sono le novità del database?. Per Prodotti, scegli Amazon RDS. Quindi esegui la ricerca utilizzando parole chiave come **`RDS Proxy`** o **`Oracle 2023`**.

Modifica	Descrizione	Data
AWS Il driver Python è disponibile a livello generale	Il driver Python di Amazon Web Services (AWS) è progettato come wrapper Python avanzato. Questo wrapper è complementare ed estende le funzionalità del driver open source Psycopg. Per ulteriori informazioni, consulta Connessione alle istanze DB con i driver. AWS	23 maggio 2024
RDS Proxy è disponibile in altre regioni	Il proxy RDS è ora disponibi le nelle regioni Asia Pacifico (Hyderabad), Asia Pacifico (Melbourne), Medio Oriente (Emirati Arabi Uniti), Israele (Tel Aviv), Canada occidentale (Calgary) ed Europa (Zurigo). Per ulteriori informazioni sul proxy RDS, consulta Utilizzo	21 maggio 2024

	di Server proxy per Amazon RDS.	
Licenza Db2 tramite Marketplace AWS	Con la licenza Db2 valida Marketplace AWS, ora puoi pagare una tariffa oraria per abbonarti alle licenze Db2 per RDS for Db2. Per ulteriori informazioni, consulta le opzioni di licenza RDS for Db2.	21 maggio 2024
Amazon RDS supporta l'accesso granulare per Performance Insights	Ora puoi consentire o negare l'accesso a singole dimensioni in Performance Insights. Questo accesso granulare può essere utilizzato per `GetResourceMetrics` e azioni. `DescribeDimensionKeys` `GetDimensionKeyDetails` Per ulteriori informazioni, consulta Concessione dell'accesso granulare per Performance Insights.	21 maggio 2024
Versioni Amazon RDS Extended Support per RDS per MySQL	È possibile visualizzare tutte le versioni delle versioni di RDS Extended Support for RDS for MySQL. Per ulteriori informazioni, consulta le versioni di Amazon RDS Extended Support per RDS for MySQL.	16 maggio 2024

Amazon RDS supporta MySQL 8.3 nell'ambiente Database Preview	MySQL 8.3 è ora disponibile nell'ambiente Database Preview negli Stati Uniti orientali (Ohio). Regione AWS Per ulteriori informazioni, vedere MySQL versione 8.3 nell'ambiente Database Preview.	30 aprile 2024
Amazon RDS per Db2 supporta i fusi orari	RDS for Db2 ora supporta l'impostazione di fusi orari locali per le nuove istanze DB RDS per Db2. Per ulteriori informazioni, consulta Fusi orari locali per le istanze database di Amazon RDS per Db2.	25 aprile 2024
Aggiornamento alle autorizzazioni del ruolo collegato ai servizi di IAM	La `AmazonRDSCustomServiceRolePolicy` politica ora concede autorizzazioni aggiuntive per associare un ruolo di servizio come profilo di istanza a un'istanza personalizzata RDS. Per ulteriori informazioni, consulta Aggiornamenti di Amazon RDS sulle policy gestite da AWS.	19 aprile 2024
Amazon RDS per Oracle supporta lo switchover di Oracle Data Guard in tutto Regioni AWS	Ora puoi utilizzare lo switchover di Oracle Data Guard in tutte le regioni supportate. Per ulteriori informazioni, vedere Panoramica dello switchover di Oracle Data Guard.	16 aprile 2024

RDS Custom per Oracle supporta Oracle Standard Edition 2	È ora possibile creare istanze DB utilizzando Standard Edition 2 su Oracle Database 12c Release 1 (12.1), 12c Release 2 (12.2), 18c e 19c. È possibile creare sia CDB che non CDB. Per ulteriori informazioni, consulta Edition and licensing support for RDS Custom for Oracle.	11 aprile 2024
Amazon RDS per Oracle supporta la versione 23.2.v1 di Oracle APEX	Puoi utilizzare APEX 23.2.v1 con Oracle Database 19c e versioni successive. Per ulteriori informazioni, consulta Oracle Application Express.	11 aprile 2024
Aggiornamento delle autorizzazioni per i ruoli collegati ai servizi RDS Custom	`AmazonRDSCustomServiceRolePolicy` Ora concede autorizzazioni aggiuntive per consentire a RDS Custom for SQL Server di ottenere informazioni sul tipo di istanza EC2 e modificare il tipo di istanza host DB. Per ulteriori informazioni, consulta Aggiornamenti alle politiche gestite. AWS	8 aprile 2024
Amazon RDS Custom for Oracle supporta la classe di istanze DB db.x2iezn	Ora puoi utilizzare la classe di istanza db.x2iezn per le istanze DB RDS Custom for Oracle. Per ulteriori informazioni, consulta Supporto delle classi di istanza database per RDS Custom per Oracle.	26 marzo 2024

Amazon RDS supporta le classi di istanze db.c6gd per cluster DB Multi-AZ	Ora puoi utilizzare le classi di istanze db.c6gd per le implementazioni di cluster DB Multi-AZ. Per ulteriori informazioni, consulta Disponibilità delle classi di istanze per i cluster DB Multi-AZ.	21 marzo 2024
Estensione del supporto per Amazon RDS	La creazione o il ripristino di un database RDS per MySQL 5.7 o RDS per PostgreSQL 11 ora registra automaticamente quel database in Amazon RDS Extended Support in modo che le applicazioni esistenti continuino a funzionare così come sono. Puoi disattiva re RDS Extended Support per evitare addebiti dopo la fine della data di supporto standard RDS per il tuo motore di database. Per ulteriori informazioni, consulta Utilizzo dell'estensione del supporto per Amazon RDS.	21 marzo 2024

Integrazione RDS per Db2 con AWS License Manager	RDS per Db2 è ora integrato con. AWS License Manager Se si utilizza il modello Bring Your Own License, l' AWS License Manager integrazione aiuta a monitorare l'utilizzo della licenza Db2 all'interno dell'organizzazione. Per ulteriori informazioni, consulta Integrazione con. AWS License Manager	20 marzo 2024
Rotazione dei certificati CA per cluster DB Multi-AZ	Ora puoi ruotare i certificati CA per i tuoi cluster DB Multi-AZ. Prendi in considerazione l'utilizzo di uno dei nuovi certificati CA rds-ca-rsa 2048-g1, 4096-g1 o 384-g1. rds-ca-rsa rds-ca-ecc Per ulteriori informazioni, consulta Rotazione del certificato SSL/TLS.	6 marzo 2024
Amazon RDS supporta lo storage io2 Block Express	Ora puoi creare istanze DB RDS che utilizzano il tipo di storage io2 Block Express. Per ulteriori informazioni, consulta io2 Block Express storage.	6 marzo 2024
RDS Custom for SQL Server supporta le classi di istanze DB db.r5b e db.x2iedn	È ora possibile utilizzare le classi di istanze db.r5b e db.x2iedn per le istanze DB di RDS Custom per SQL Server. Per ulteriori informazioni, consulta Supporto delle classi di istanze DB per RDS Custom for SQL Server.	4 marzo 2024

RDS Custom for Oracle è disponibile nella regione Medio Oriente (Emirati Arabi Uniti)	È possibile creare istanze RDS Custom per Oracle DB nella regione Medio Oriente (Emirati Arabi Uniti). Per una tabella che mostra tutte le aree supportate Regioni AWS, consulta Regioni supportate e motori DB per RDS Custom for Oracle.	4 marzo 2024
Nuova politica AWS gestita	Amazon RDS ha aggiunto una nuova policy gestita denominata `AmazonRDSCustomInstanceProfileRolePolicy` per consentire a RDS Custom di eseguire azioni di automazione e attività di gestione del database tramite un profilo di istanza EC2. Per ulteriori informazioni, consulta Aggiornamenti di Amazon RDS sulle policy gestite da AWS.	27 febbraio 2024
Amazon RDS supporta MariadB 10.11.7, 10.6.17, 10.5.24 e 10.4.33	Ora puoi creare istanze Amazon RDS DB con MariaDB versione 10.11.7, 10.6.17, 10.5.24 e 10.4.33. Per ulteriori informazioni, consulta Versioni di MariaDB in Amazon RDS.	26 febbraio 2024
I cluster Amazon RDS Multi-AZ DB supportano il volume di storage Amazon EBS gp3	I cluster DB Multi-AZ ora supportano i volumi EBS basati su SSD gp3. Per ulteriori informazioni, consulta gp3 storage.	26 febbraio 2024

Supporto Amazon RDS per la AWS Secrets Manager regione di Israele (Tel Aviv)	Amazon RDS supporta Secrets Manager nella regione di Israele (Tel Aviv). Per ulteriori informazioni, consulta Password management with Amazon RDS and AWS Secrets Manager (Gestione delle password per Amazon RDS e AWS Secrets Manager).	21 febbraio 2024
Amazon RDS per Db2 supporta la registrazione di audit	RDS per Db2 ora supporta la registrazione di audit a livello di database. Quando abiliti la registrazione di controllo per un database RDS for Db2, Amazon RDS registra l'attività del database e archivia i log di controllo in Amazon S3. Per ulteriori informazioni, consulta Db2 audit logging.	15 febbraio 2024
Estensione del supporto per Amazon RDS	Amazon RDS ora abilita automaticamente Amazon RDS Extended Support quando le versioni principali del motore RDS for MySQL e RDS for PostgreSQL nelle istanze DB e nei cluster DB Multi-AZ raggiungono la data di fine del supporto standard RDS. Per ulteriori informazioni, consulta Utilizzo dell'estensione del supporto per Amazon RDS.	15 febbraio 2024

Amazon RDS supporta MySQL 8.0.36	Ora puoi creare istanze Amazon RDS DB con MySQL versione 8.0.36. Per ulteriori informazioni, consulta Versioni di MySQL in Amazon RDS.	12 febbraio 2024
Amazon RDS supporta la collazione EBCDIC per RDS per Db2	Ora puoi creare database Db2 che utilizzano sequenze di confronto EBCDIC per ordinare i contenuti nei database. Per ulteriori informazioni, consulta la collazione EBCDIC per i database Db2 su Amazon RDS.	29 gennaio 2024
Aggiornamento al certificato CA predefinito	Il certificato CA predefinito è impostato su `rds-ca-rsa2048-g1` . Per ulteriori informazioni, consulta la sezione Utilizzo di SSL/TLS per crittografare una connessione a un'istanza database.	26 gennaio 2024
Amazon RDS per PostgreSQL supporta due nuove casse per PL/Rust, croaring-rs e num-bigint	Puoi usare due nuove casse in Amazon RDS for PostgreSQL. Per ulteriori informazioni, consulta Using crates with PL/Rust.	24 gennaio 2024

Amazon RDS per PostgreSQL supporta la versione 1.3 di TLS	È possibile utilizzare Transport Layer Security (TLS) versione 1.3 in RDS per PostgreSQL. Per ulteriori informazioni, consulta Utilizzo del protocollo SSL con un'istanza database PostgreSQL.	24 gennaio 2024
RDS Custom per SQL Server supporta Microsoft SQL Server 2022	È ora possibile creare istanze DB RDS Custom per SQL Server che utilizzano SQL Server 2022. Per ulteriori informazioni, vedere Utilizzo di RDS Custom per SQL Server.	22 gennaio 2024
Aggiornamento delle autorizzazioni delle policy AWS gestite	Il `AmazonRDSServiceRolePolicy` ruolo `AWSServiceRoleForRDS` collegato al servizio ha nuovi ID di dichiarazione. Per ulteriori informazioni, consulta Aggiornamenti di Amazon RDS sulle policy gestite da AWS.	19 gennaio 2024
RDS Custom for Oracle supporta la regione Europa (Parigi)	È possibile creare istanze RDS Custom per Oracle DB nella regione Europa (Parigi). Per ulteriori informazioni, consulta Regioni e motori DB supportati per RDS Custom for Oracle.	18 gennaio 2024

Amazon RDS for MySQL supporta la replica da più fonti	Ora puoi utilizzare la replica da più fonti su istanze DB RDS per MySQL. Per ulteriori informazioni, consulta Configurazione della replica multi-source su RDS for MySQL.	16 gennaio 2024
Amazon RDS supporta MySQL 8.2 nell'ambiente Database Preview	MySQL 8.2 è ora disponibile nell'ambiente Database Preview negli Stati Uniti orientali (Ohio). Regione AWS Per ulteriori informazioni, vedere MySQL versione 8.2 nell'ambiente Database Preview.	11 gennaio 2024
RDS Proxy è disponibile nella regione Europa (Spagna)	Il proxy RDS è ora disponibile nella regione Europa (Spagna). Per ulteriori informazioni sul proxy RDS, consulta Utilizzo di Server proxy per Amazon RDS.	8 gennaio 2024
Amazon RDS è disponibile nella regione Canada occidentale (Calgary)	Amazon RDS è ora disponibile nella regione Canada occidentale (Calgary). Per ulteriori informazioni, consulta Regioni e zone di disponibilità.	20 dicembre 2023
Amazon RDS per Db2 supporta 5.000 utenti locali	Ora puoi aggiungere fino a 5.000 utenti locali a un elenco di autorizzazioni. Per ulteriori informazioni, vedere rdsadmin.add_user.	20 dicembre 2023

Amazon RDS supporta la visualizzazione e la risposta ai consigli	I consigli di Amazon RDS ora includono consigli proattivi basati su soglie e reattivi basati sull'apprendimento automatico per RDS for PostgreSQL. Per ulteriori informazioni, consulta Visualizzazione e risposta ai consigli di Amazon RDS.	19 dicembre 2023
Amazon RDS supporta MariadB 10.11.6, 10.6.16, 10.5.23 e 10.4.32	Ora puoi creare istanze Amazon RDS DB con MariaDB versione 10.11.6, 10.6.16, 10.5.23 e 10.4.32. Per ulteriori informazioni, consulta Versioni di MariaDB in Amazon RDS.	12 dicembre 2023
Amazon RDS introduce integrazioni zero-ETL con Amazon Redshift (anteprima)	Le integrazioni zero-ETL forniscono una soluzione completamente gestita per rendere disponibili i dati transazionali in Amazon Redshift entro pochi secondi dalla loro scrittura su un'istanz a DB RDS for MySQL. Per ulteriori informazioni, consulta Lavorare con le integrazioni Zero-ETL di Amazon RDS con Amazon Redshift (anteprima).	28 novembre 2023
Amazon RDS supporta i motori di IBM Db2 database	Ora puoi eseguire motori di IBM Db2 database in Amazon RDS. Per ulteriori informazi oni, consulta Amazon RDS for Db2.	27 novembre 2023

RDS per PostgreSQL supporta gli aggiornamenti delle versioni principali a PostgreSQL 16.1 e gli aggiornamenti delle versioni secondarie a 15.5, 14.10, 13.13, 12.17 e 11.22	Con RDS per PostgreSQL, ora puoi aggiornare il motore DB alla versione principal e 16.1 e gli aggiornamenti delle versioni secondarie a 15.5, 14.10, 13.13, 12.17 e 11.22. Per ulteriori informazi oni, consulta Aggiornamento del motore DB PostgreSQL per Amazon RDS.	17 novembre 2023
RDS Custom for Oracle supporta i gruppi di opzioni	È possibile creare o modificar e un gruppo di opzioni e associarlo a un'istanza DB RDS Custom for Oracle. L'`Timezone`opzione è ora supportata. Per ulteriori informazioni, vedere Utilizzo dei gruppi di opzioni in RDS Custom for Oracle.	17 novembre 2023
Amazon RDS for MySQL supporta il plug-in Group Replication	È ora possibile configurare un cluster active-active con RDS per istanze DB MySQL versione 8.0.35 o superiore utilizzando il plug-in Group Replication sviluppato e gestito dalla comunità MySQL. Per ulteriori informazioni, consulta Configurazione dei cluster active-active per RDS for MySQL.	17 novembre 2023

Amazon RDS Proxy supporta RDS per PostgreSQL 16.1	È ora possibile creare proxy utilizzando RDS Proxy per istanze DB RDS per PostgreSQL 16.1. Per ulteriori informazioni, consultare Utlizzo Proxy di Amazon RDS.	17 novembre 2023
RDS Custom per SQL Server supporta Microsoft SQL Server 2019 Developer edition	È possibile creare istanze DB RDS Custom per SQL Server che utilizzano l'edizione SQL Server 2019 Developer. Per ulteriori informazioni, vedi l'argomento Modello Porta i tuoi media (BYOM) con RDS Custom per SQL Server.	16 novembre 2023
Aggiornamenti di versione minori dei cluster DB Multi-AZ con tempi di inattività minimi	Quando esegui un aggiornamento di versione minore di un cluster DB Multi-AZ, Amazon RDS ora aggiorna le istanze DB di lettura prima dell'istanza di scrittura, riducendo così in modo significativo i tempi di inattività. Puoi ridurre ulteriormente i tempi di inattività a un secondo o meno utilizzando RDS Proxy. Per ulteriori informazioni, consulta Implementazioni cluster di database multi-AZ.	16 novembre 2023
RDS per SQL Server supporta Microsoft SQL Server 2022	Ora puoi creare istanze DB RDS che utilizzano SQL Server 2022. Per ulteriori informazioni, consulta Versioni di Microsoft SQL Server su Amazon RDS.	15 novembre 2023

RDS for MySQL supporta l'aggiornamento delle istantanee dalla versione 5.7 alla 8.0	È ora possibile aggiornare la versione del motore di uno snapshot RDS for MySQL dalla versione 5.7 alla versione 8.0. È possibile farlo utilizzando o utilizzando l'API AWS Management Console RDS o `ModifyDBSnapshot` . AWS CLI Per ulteriori informazioni, vedere Aggiornamento di una versione del motore di snapshot MySQL DB.	15 novembre 2023
RDS Custom for SQL Server supporta il ripristino point-in-time di 1.000 database	Ora puoi creare fino a 1.000 database idonei per il backup completo e il ripristino point-in-time sulla tua istanza DB RDS Custom for SQL Server. Per ulteriori informazioni, vedere Ripristino di un'istanza RDS Custom for SQL Server in un point-in-time.	15 novembre 2023
RDS Custom per SQL Server supporta l'utilizzo di una chiave Service Master	RDS Custom per SQL Server ora supporta l'utilizzo di una Service Master Key (SMK). Un SMK consente di crittografare oggetti come le credenziali e di utilizzare funzionalità di SQL Server come TDE e crittografia a colonne. Per ulteriori informazioni, vedere Utilizzo di una chiave master del servizio con RDS Custom per SQL Server.	13 novembre 2023

Amazon RDS supporta MySQL 8.1 nell'ambiente di anteprima del database	MySQL 8.1 è ora disponibile nell'ambiente Database Preview negli Stati Uniti orientali (Ohio). Regione AWS Per maggiori informazioni, consulta MySQL versione 8.1 nell'ambiente di anteprima del database.	10 novembre 2023
RDS supporta MySQL 8.0.35 e MySQL 5.7.44	È ora possibile creare istanze database Amazon RDS che eseguono MySQL versione 8.0.35 e 5.7.44. Per ulteriori informazioni, consulta Versioni di MySQL in Amazon RDS.	9 novembre 2023
Server proxy per RDS supporta i cluster di database Multi-AZ	Server proxy per RDS ora supporta la connessione a cluster database Multi-AZ. Per ulteriori informazioni, consulta Utilizzo degli endpoint Amazon RDS Proxy.	9 novembre 2023
RDS Custom for Oracle è disponibile in AWS GovCloud (US) Regions	Amazon RDS è ora disponibile nelle AWS GovCloud (US) Regions. Per ulteriori informazioni, consulta Regioni e motori DB supportati per RDS Custom for Oracle.	9 novembre 2023

Scritture ottimizzate per Amazon RDS supporta la classe di istanza database db.m5	Scritture ottimizzate per Amazon RDS ora supporta la classe di istanza database db.m5. Per ulteriori informazioni, consulta Prestazioni di scrittura migliorate con Scritture ottimizzate per Amazon RDS per MariaDB e Prestazioni di scrittura migliorate con Scritture ottimizzate per Amazon RDS per MySQL.	9 novembre 2023
Amazon RDS per Oracle supporta la configurazione multi-tenant dell'architettura CDB	Con la funzionalità multi-tenant di RDS per Oracle, RDS offre un'architettura e un'esperienza Oracle multitenant completamente gestite per i database Oracle. È possibile utilizzare le API RDS per creare più PDB, chiamati database del tenant, in un CDB. RDS offre la configurazione multi-tenant dell'architettura CDB come alternativa alla configurazione a tenant singolo legacy. Per ulteriori informazioni, consulta Configurazione multi-tenant dell'architettura CDB.	8 novembre 2023

Amazon RDS esporta i parametri di Performance Insights in Amazon CloudWatch	Performance Insights ti consente di esportare i dashboard delle metriche preconfigurati o personalizzati su Amazon. CloudWatch I dashboard delle metriche esportate possono essere visualizzati nella console. CloudWatch Puoi anche esportare un widget metrico di Performance Insights selezionato e visualizzare i dati delle metriche nella CloudWatch console. Per ulteriori informazioni, consulta Esportazione delle metriche di Performance Insights in. CloudWatch	8 novembre 2023
Amazon RDS Custom per Oracle consente di aggiornare il sistema operativo su un'istanza database	È ora possibile aggiornare il database o il sistema operativo per un'istanza database RDS Custom per Oracle utilizzando il comando CLI `modify-db-instance` . Per ulteriori informazioni, consulta Aggiornamento di un'istanza database per Amazon RDS Custom for Oracle.	7 novembre 2023

Server proxy per RDS supporta il protocollo esteso per RDS per PostgreSQL	È ora possibile eseguire protocolli di query estesi su un'istanza database RDS per PostgreSQL. Per ulteriori informazioni, consultare Utlizzo Proxy di Amazon RDS.	6 novembre 2023
RDS per PostgreSQL supporta le implementazioni blu/verde RDS	Ora è possibile creare un'implementazione blu/verde da un'istanza database RDS per PostgreSQL. Per ulteriori informazioni, consulta Utilizzo delle implementazioni blu/verde Amazon RDS per gli aggiornamenti del database.	26 ottobre 2023
Aggiornamento delle politiche AWS gestite	Le policy gestite `AmazonRDSPerformanceInsightsReadOnly` e `AmazonRDSPerformanceInsightsFullAccess` ora includono `Sid` (ID istruzione) come identificativo nell'istruzione della policy. Per ulteriori informazioni, consulta Aggiornamenti di Amazon RDS sulle policy gestite da AWS.	23 ottobre 2023
RDS Custom per Oracle supporta la Regione Europa (Milano)	Per ulteriori informazioni, consulta Regioni e motori DB supportati per RDS Custom for Oracle.	23 ottobre 2023

Abilitazione di Scritture ottimizzate per RDS sui database esistenti	Ora puoi abilitare Scritture ottimizzate per RDS su un'istanza database esistente anche se è stata creata con una versione del motore, una classe di istanza database o una configurazione del file system che non supporta la funzionalità. Per ulteriori informazioni, consulta Abilitazione delle scritture ottimizzate per RDS in un database esistente per RDS per MySQL e Abilitazione delle scritture ottimizzate per RDS in un database esistente per RDS per MariaDB.	19 ottobre 2023
Amazon RDS supporta l'utilizzo di un volume di log dedicato (DLV).	Ora è possibile utilizzare un volume di log dedicato (DLV) con RDS per MariaDB, RDS per MySQL e RDS per PostgreSQL. I DLV sono ideali per database con archiviazione allocata di grandi dimensioni, requisiti di I/O al secondo (IOPS) elevati o carichi di lavoro sensibili alla latenza. Per ulteriori informazioni, consulta Utilizzo di un volume di log dedicato (DLV).	17 ottobre 2023

Amazon RDS per PostgreSQL, MySQL e MariaDB supportano nuove classi di istanza database	Puoi creare istanze database di Amazon RDS che eseguono PostgreSQL, MySQL e MariaDB che utilizzano le classi di istanza database db.m6.in, db.m6idn, db.r6.in e db.r6.idn. Per ulteriori informazioni, consulta Motori di database supportati per tutte le classi di istanza database disponibili.	12 ottobre 2023
Amazon RDS per PostgreSQL supporta pgactive	L'estensione pgactive è disponibile in Amazon RDS per PostgreSQL. Per ulteriori informazioni, consulta Utilizzo delle estensioni PostgreSQL con Amazon RDS per PostgreSQL.	9 ottobre 2023
RDS Custom per Oracle è disponibile nella regione Asia Pacific (Giacarta)	È possibile creare istanze RDS Custom per Oracle DB nella regione Asia Pacifico (Giacarta). Per ulteriori informazioni, consulta Regioni supportate e motori DB per RDS Custom for Oracle.	5 ottobre 2023

RDS Custom per SQL Server supporta nove regole di confronto a livello di server	RDS Custom per SQL Server ora supporta un'ampia gamma di regole di confronto dei server, con codifica tradizionale e UTF-8, per le versioni locali SQL_Latin, giapponese, tedesco e arabo. Per ulteriori informazioni, consulta l'argomento relativo alle regole di confronto e supporto caratteri per istanze database RDS Custom per SQL Server.	26 settembre 2023
Aggiornamento delle autorizzazioni delle policy AWS gestite	Il ruolo `AWSServiceRoleForRDSCustom` collegato al servizio dispone `AmazonRDSCustomServiceRolePolicy` di nuove autorizzazioni che consentono a RDS Custom di creare, modificare ed eliminare le regole gestite. EventBridge Per ulteriori informazioni, consulta Aggiornamenti di Amazon RDS sulle policy gestite da AWS.	20 settembre 2023

Amazon RDS pubblica i contatori di Performance Insights su Amazon CloudWatch	La funzione matematica dei parametri DB_PERF_INSIGHTS nella console CloudWatch consente di interrogare i contatori di Amazon RDS for Performance Insights. Per ulteriori informazioni, consulta Creazione di CloudWatch allarmi per monitorare Amazon RDS.	20 settembre 2023
Performance Insights supporta statistiche a livello digest per Server	Quando utilizzi Performance Insights, puoi visualizzare le statistiche SQL sia a livello di istruzione che di digest per Amazon RDS per SQL Server. Per ulteriori informazioni, consulta Analisi delle query in esecuzione in SQL Server.	18 settembre 2023
Amazon RDS per PostgreSQL, MySQL e MariaDB supportano i tipi di classi di istanza database db.m6.id e db.r6.id	Ora puoi creare istanze database Amazon RDS che eseguono PostgreSQL, MySQL e MariaDB che utilizzano le classi di istanza database di tipo db.m6g.x e db.r6g.x. Questi tipi offrono archiviazione SSD locale basata su NVMe. Per ulteriori informazioni, consulta Motori di database supportati per tutte le classi di istanza database disponibili.	11 settembre 2023

Supporto per aggiornamenti della versione principale per cluster database muti-AZ Amazon RDS per PostgreSQL	Ora è possibile eseguire aggiornamenti della versione principale dei cluster database multi-AZ RDS per PostgreSQL. Per ulteriori informazioni, consulta Implementazioni cluster di database multi-AZ.	7 settembre 2023
Amazon RDS supporta MariaDB 10.11.5, 10.6.15, 10.5.22 e 10.4.31	Ora puoi creare istanze database Amazon RDS che eseguono MariaDB versione 10.11.5, 10.6.15, 10.5.22 e 10.4.31. Per ulteriori informazioni, consulta Versioni di MariaDB in Amazon RDS.	7 settembre 2023
Estensione del supporto di Amazon RDS	Amazon RDS annuncia la possibilità a breve di continuare a eseguire le versioni principali del motore RDS per MySQL e RDS per PostgreSQL nelle istanze database oltre la data di fine del supporto RDS standard. Per ulteriori informazioni, consulta Utilizzo dell'estensione del supporto per Amazon RDS.	1 settembre 2023
RDS Custom supporta l'avvio e l'arresto di un'istanza database RDS Custom per SQL Server	RDS Custom ora supporta l'avvio e l'arresto di un'istanza database RDS Custom per SQL Server. Per ulteriori informazioni, consulta Avvio e arresto di un'istanza database RDS Custom per SQL Server.	31 agosto 2023

Scritture ottimizzate per Amazon RDS supporta la classe di istanza database db.r5	Scritture ottimizzate per Amazon RDS ora supporta la classe di istanza database db.r5. Per ulteriori informazioni, consulta Prestazioni di scrittura migliorate con Scritture ottimizzate per Amazon RDS per MariaDB e Prestazioni di scrittura migliorate con Scritture ottimizzate per Amazon RDS per MySQL.	31 agosto 2023
Amazon RDS per Oracle supporta l'aggiornamento automatico dei file con il fuso orario per i CDB	Con l'opzione `TIMEZONE_FILE_AUTOUPGRADE`, è possibile aggiornare il file di fuso orario corrente alla versione più recente del database container (CDB) RDS per Oracle. Per ulteriori informazioni, consulta Aggiornamento automatico del file di fuso orario Oracle.	29 agosto 2023
Scritture ottimizzate per Amazon RDS supporta le classi di istanza database db.m6g e db.m6i	Scritture ottimizzate per Amazon RDS ora supporta le classi di istanza database db.m6g e db.m6i. Per ulteriori informazioni, consulta Prestazioni di scrittura migliorate con Scritture ottimizzate per Amazon RDS per MariaDB e Prestazioni di scrittura migliorate con Scritture ottimizzate per Amazon RDS per MySQL.	28 agosto 2023

Amazon RDS supporta MariaDB versione 10.11	Ora puoi creare istanze database Amazon RDS che eseguono MariaDB versione 10.11. Per ulteriori informazioni, consulta Versioni di MariaDB in Amazon RDS.	21 agosto 2023
Aggiornamento delle autorizzazioni delle policy AWS gestite	La policy `AmazonRDSCustomServiceRolePolicy` del ruolo collegato ai servizi `AWSServiceRoleForRDSCustom` dispone di nuove autorizzazioni che consentono a RDS Custom di creare interfacce di rete. Per ulteriori informazioni, consulta Aggiornamenti di Amazon RDS sulle policy gestite da AWS.	18 agosto 2023
Aggiornamento delle autorizzazioni delle policy AWS gestite	La policy gestita `AmazonRDSFullAccess` dispone di nuove autorizzazioni che consentono di generare, visualizzare ed eliminare il report di analisi delle prestazioni per un periodo di tempo. Per ulteriori informazioni, consulta Aggiornamenti di Amazon RDS sulle policy gestite da AWS.	17 agosto 2023

Aggiornamento delle autorizzazioni delle policy AWS gestite	L'aggiunta di nuove autorizzazioni alla policy gestita `AmazonRDSPerformanceInsightsReadOnly` e l'aggiunta di una nuova policy gestita `AmazonRDSPerformanceInsightsFullAccess` consente di generare un report di analisi del carico del database per un periodo di tempo. Per ulteriori informazioni, consulta Aggiornamenti di Amazon RDS sulle policy gestite da AWS.	16 agosto 2023
Amazon RDS supporta l'analisi delle prestazioni per un periodo di tempo	Performance Insights consente di creare e visualizzare report di analisi delle prestazioni per un periodo di tempo specifico. Il report fornisce gli approfondimenti identificati e i suggerimenti per risolvere problemi relativi alle prestazioni. Per ulteriori informazioni, consulta Analisi del carico del database per un periodo di tempo.	16 agosto 2023

Amazon RDS Custom per Oracle supporta le classi di istanza database db.r5b e db.x2iedn	È ora possibile utilizzare le classi di istanza db.r5b e db.x2iedn per le istanze database RDS Custom for Oracle. Per ulteriori informazioni, consulta Supporto delle classi di istanza database per RDS Custom per Oracle.	16 agosto 2023
Amazon RDS Custom per Oracle supporta le classi di istanza database db.m6i, db.r6i e db.t3	È ora possibile utilizzare le classi di istanza db.m6i, db.r6i e db.t3 per le istanze database RDS Custom per Oracle. Per ulteriori informazioni, consulta Supporto delle classi di istanza database per RDS Custom per Oracle.	15 agosto 2023
Amazon RDS per PostgreSQL ora supporta PostgreSQL versione 16 Beta 3 nell'ambiente di anteprima del database	La versione 16 Beta 3 di PostgreSQL è ora disponibile nell'ambiente di anteprima del database negli Stati Uniti orientali (Ohio). Regione AWS Per ulteriori informazioni, consultare Lavorare sull'ambiente di anteprima del database.	11 agosto 2023
Amazon RDS supporta MySQL 8.0.34 e 5.7.43	È ora possibile creare istanze database Amazon RDS che eseguono MySQL versione 8.0.34 e 5.7.43. Per ulteriori informazioni, consulta Versioni di MySQL in Amazon RDS.	9 agosto 2023

RDS per SQL Server supporta la visualizzazione dei parametri del sistema operativo per la replica di standby	È ora possibile visualizzare i parametri del sistema operativo per la replica di standby per RDS per SQL Server. Per ulteriori informazioni, consulta Visualizzazione dei parametri nella console RDS.	3 agosto 2023
RDS per Oracle supporta Oracle Data Guard per CDB	RDS per Oracle supporta le repliche di lettura Data Guard per database dei container (CDB) Oracle Database 19c e 21c. È possibile creare, gestire e promuovere repliche di lettura in un CDB, proprio come in un non CDB, usando API RDS esistenti. Per ulteriori informazioni, consulta Repliche di lettura multi-tenant.	1° agosto 2023
Amazon RDS è disponibile nella regione Israele (Tel Aviv)	Amazon RDS è ora disponibile nella regione Israele (Tel Aviv). Per ulteriori informazioni, consulta Regioni e zone di disponibilità.	1° agosto 2023
Amazon RDS supporta Oracle APEX versione 23.1.v1	È possibile usare APEX 23.1.v1 con Oracle Database 19c e versioni successive. Per ulteriori informazioni, consulta Oracle Application Express.	26 luglio 2023

Amazon RDS Custom per Oracle supporta un SID Oracle non predefinito	Quando crei un'istanza database RDS Custom per Oracle utilizzando Oracle Database 19c, puoi specifica re un identificatore di sistema Oracle non predefinito (Oracle SID). Questo valore è anche il nome del CDB. Per ulteriori informazioni, consulta Considerazioni sull'architettura multilocazione.	21 luglio 2023
RDS per SQL Server supporta Active Directory gestito dal cliente	È ora possibile utilizzare Active Directory gestito dal cliente per unire direttame nte le istanze database RDS per SQL Server ai domini Microsoft Active Directory (AD). I domini AD gestiti dal cliente possono essere on-premise o nel cloud. Per ulteriori informazioni, consulta Utilizzo di Active Directory gestito dal cliente.	7 luglio 2023
Supporto della funzionalità PostgreSQL di replica logica per cluster database multi-AZ	Puoi utilizzare la funzionalità di replica logica di PostgreSQ L con il cluster database multi-AZ per replicare e sincroniz zare singole tabelle anziché l'intera istanza database. Per ulteriori informazioni, consulta Utilizzo della replica logica di PostgreSQL con cluster database multi-AZ.	6 luglio 2023

Amazon RDS per PostgreSQL ora supporta PostgreSQL versione 16 Beta 2 nell'ambiente di anteprima del database	La versione 16 Beta 2 di PostgreSQL è ora disponibile nell'ambiente di anteprima del database negli Stati Uniti orientali (Ohio). Regione AWS Per ulteriori informazioni, consultare Lavorare sull'ambiente di anteprima del database.	6 luglio 2023
Aggiornamento delle autorizzazioni delle policy gestite AWS	La policy `AmazonRDSCustomServiceRolePolicy` del ruolo collegato ai servizi `AWSServiceRoleForRDSCustom` dispone di nuove autorizzazioni che consentono a RDS Custom per Oracle di utilizzare gli snapshot. Per ulteriori informazioni, consulta Aggiornamenti di Amazon RDS sulle policy gestite da AWS.	23 giugno 2023
RDS supporta MariaDB versione 10.6.14, 10.5.21 e 10.4.30	Puoi ora creare istanze database Amazon RDS che eseguono MariaDB versione 10.6.14, 10.5.21 e 10.4.30. Per ulteriori informazioni, consulta Versioni di MariaDB in Amazon RDS.	22 giugno 2023

RDS supporta MySQL 8.0.33 e 5.7.42	Puoi ora creare creare istanze database Amazon RDS che eseguono MySQL versione 8.0.33 e 5.7.42. Per ulteriori informazioni, consulta Versioni di MySQL in Amazon RDS.	15 giugno 2023
RDS supporta MariaDB 10.6.13, 10.5.20, 10.4.29 e 10.3.39	Ora puoi creare istanze database Amazon RDS che eseguono MariaDB versione 10.6.13, 10.5.20, 10.4.29 e 10.3.39. Per ulteriori informazioni, consulta Versioni di MariaDB in Amazon RDS.	15 giugno 2023
RDS per Oracle supporta le tablespace trasportabili	È possibile migrare i dati da un database Oracle on-premise a un'istanza database RDS per Oracle utilizzando le tablespace trasportabili. Questa tecnica non richiede licenze aggiuntive ed è la tecnica di migrazione con il minor tempo di inattività. Per ulteriori informazioni, consulta Migrazione utilizzando le tablespace trasportabili Oracle.	15 giugno 2023
Amazon RDS supporta Server proxy per RDS con RDS per MariaDB versione 10.6	Puoi ora creare creare un Server proxy per RDS con un database RDS per MariaDB versione 10.6. Per ulteriori informazioni sul proxy RDS, consulta Utilizzo di Server proxy per Amazon RDS.	15 giugno 2023

RDS Custom per SQL Server supporta il modello di servizio Porta i tuoi media (BYOM)	Puoi ora creare una versione del motore personalizzato (CEV) utilizzando i tuoi media SQL Server. Per ulteriori informazioni, vedi l'argomento Modello Porta i tuoi media (BYOM) con RDS Custom per SQL Server.	8 giugno 2023
RDS per Oracle può convertire un database Oracle 19c non CDB in un database CDB	Se l'istanza database esegue Oracle Database 19c con RU di aprile 2021 o versione successiva, è possibile convertire un database non CDB in un database CDB (database container). Dopo aver convertito l'architettura, è possibile aggiornare il database CDB versione 19c a un database CDB versione 21c. Questo passaggio è necessario perché non è possibile aggiornare il database e convertire l'architettura utilizzando un unico comando. Per ulteriori informazioni, consulta l'argomento relativo alla Conversione di un database non CDB RDS per Oracle in un database CDB.	31 maggio 2023

Cluster database multi-AZ disponibili nelle regioni Cina	I cluster DB Multi-AZ sono ora disponibili in Regioni AWS Cina (Pechino) e Cina (Ningxia). Per ulteriori informazioni, consulta Regioni supportate e motori DB per cluster DB Multi-AZ in Amazon RDS.	30 maggio 2023
Letture ottimizzate per Amazon RDS supporta i cluster database multi-AZ	Letture ottimizzate per Amazon RDS ora supporta i cluster database multi-AZ. Per ulteriori informazioni, consulta Prestazioni delle query migliorate per RDS per MySQL con Letture ottimizzate per Amazon RDS e Prestazioni delle query migliorate per RDS per PostgreSQL con Letture ottimizzate per Amazon RDS.	30 maggio 2023
RDS Custom for Oracle supporta la regione Asia Pacifico (Giacarta)	Per ulteriori informazioni, consulta Regioni supportate e motori DB per RDS Custom for Oracle.	29 maggio 2023

Creare una replica di lettura dell'istanza database con un cluster database RDS per PostgreSQL Multi-AZ di origine	Ora è possibile creare una replica di lettura dell'istanza database con un cluster database RDS per PostgreSQL Multi-AZ come origine. In precedenza, era supportato solo RDS per MySQL. Per ulteriori informazioni, consulta l'argomento relativo alla creazione di una replica di lettura di un'istanza database da un cluster di database Multi-AZ.	24 maggio 2023
Amazon RDS offre una combinazione di Performance Insights e CloudWatch metriche nel pannello di controllo Performance Insights.	Amazon RDS ora offre una visualizzazione consolidata di Performance Insights e delle CloudWatch metriche nella dashboard di Performance Insights. Per ulteriori informazioni, consultare Visualizzazione delle metriche combinate nella console Amazon RDS.	24 maggio 2023

Letture ottimizzate per Amazon RDS disponibili nelle regioni della Cina	Letture ottimizzate per Amazon RDS è ora disponibile nelle Regioni AWS Cina (Pechino) e Cina (Ningxia). Per ulteriori informazioni, vedi Prestazioni delle query migliorate per RDS per MariaDB con Letture ottimizzate per Amazon RDS e Prestazioni delle query migliorate per RDS per MySQL con Letture ottimizzate per Amazon RDS.	24 aprile 2023
Supporto Amazon RDS per AWS Secrets Manager le regioni cinesi	Amazon RDS supporta Secrets Manager nelle regioni Cina (Pechino) e Cina (Ningxia). Per ulteriori informazioni, consulta Password management with Amazon RDS and AWS Secrets Manager (Gestione delle password per Amazon RDS e AWS Secrets Manager).	20 aprile 2023

RDS Custom per Oracle supporta il riutilizzo degli ID AMI per le nuove CEV	Quando crei una versione del motore personalizzato (CEV), RDS Custom per Oracle usa per impostazione predefinita la Amazon Machine Image (AMI) più recente disponibi le. Ora puoi specificare un ID AMI utilizzato in una CEV precedente. Per ulteriori informazioni, consulta la sezione relativa alla creazione di una CEV.	19 aprile 2023
Amazon RDS supporta la pubblicazione di eventi contenenti tag agli abbonati degli argomenti	Le notifiche di eventi di Amazon RDS inviate ad Amazon Simple Notificat ion Service (Amazon SNS) o EventBridge Amazon ora contengono tag di evento nel corpo del messaggio. Questi tag forniscono i dati sulle risorse interessati dall'even to del servizio. Per ulteriori informazioni, consulta Tag e attributi delle notifiche eventi di Amazon RDS.	17 aprile 2023
Acquisto di istanze riservate per un cluster di database Multi-AZ	Ora puoi acquistare istanze database riservate per un cluster di database Multi-AZ. Per ulteriori informazioni, consulta l'argomento relativo alle istanze database riservate per un cluster di database Muti-AZ.	12 aprile 2023

Amazon RDS supporta le classi di istanza db.m7g e db.r7g	Ora è possibile utilizzare le classi di istanza db.m7g e db.r7g per le istanze database RDS per MySQL, RDS per MariaDB e RDS per PostgreSQL. Per ulteriori informazioni, consulta Motori di database supportati per tutte le classi di istanza database disponibili.	12 aprile 2023
Aggiornamento delle autorizzazioni del ruolo collegato ai servizi per Amazon RDS Custom	Ora `AmazonRDSCustomServiceRolePolicy` concede autorizzazioni aggiuntive per consentire a RDS Custom per SQL Server di utilizzare Amazon SQS e creare snapshot. Per ulteriori informazioni, consulta l'argomento relativo agli aggiornamenti sulle policy gestite da AWS.	6 aprile 2023

Migrazione a un cluster di database Multi-AZ RDS per MySQL utilizzando una replica di lettura	Ora puoi utilizzare una replica di lettura per eseguire la migrazione di un'implem entazione Single-AZ o di un'implementazione di istanza database Multi-AZ RDS per MySQL a un'implementazione di cluster di database Multi-AZ RDS per MySQL con tempi di inattività ridotti. Per ulteriori informazioni, consulta Migrating to a Multi-AZ DB cluster using a read replica (Migrazione a un cluster di database multi-AZ tramite una replica di lettura).	6 aprile 2023
Creazione di una replica di lettura di un'istanza database da un cluster di database Multi-AZ	Ora puoi creare una replica di lettura dell'istanza DB da un cluster DB Multi-AZ per superare la capacità di calcolo del cluster di origine. Per ulteriori informazioni, consulta l'argomento relativo alla creazione di una replica di lettura di un'istanza database da un cluster di database Multi-AZ.	6 aprile 2023

Amazon RDS Custom per SQL Server supporta le implementazioni Multi-AZ	Puoi creare un'implementazione Multi-AZ con RDS Custom per SQL Server. Per ulteriori informazioni, consulta l'argomento relativo alla gestione di un'implementazione Multi-AZ per RDS Custom per SQL Server.	6 aprile 2023
Aggiornamento delle autorizzazioni delle policy AWS gestite	Le `AmazonRDSReadOnlyAccess` policy `AmazonRDSFullAccess` and ora concedono autorizzazioni aggiuntive per consentire la visualizzazione dei risultati di Amazon DevOps Guru nella console RDS. Per ulteriori informazioni, consulta Aggiornamenti di Amazon RDS sulle policy gestite da AWS.	30 marzo 2023
Amazon RDS supporta Oracle APEX versione 22.2.v1	È possibile utilizzare APEX 22.2.v1 con tutte le versioni supportate di Oracle Database. Per ulteriori informazioni, consulta Oracle Application Express.	30 marzo 2023

Amazon DevOps Guru disponibile per RDS per PostgreSQL	RDS per PostgreSQL ti avvisa delle anomalie recenti rilevate da Amazon Guru. DevOps La pagina dei dettagli del database della console ti avvisa delle anomalie correnti e verificatesi nelle ultime 24 ore. DevOpsGuru pubblica approfondimenti proattivi con consigli per aiutare a risolvere i problemi nei database RDS per PostgreSQL prima che si verifichino. Per ulteriori informazioni, consulta Come funziona Guru for RDS. DevOps	30 marzo 2023
RDS Custom supporta il volume di archiviazione Amazon EBS gp3	RDS Custom per Oracle e RDS Custom per SQL Server supportano entrambi i volumi EBS basati su SSD io1, gp2 e gp3. Per ulteriori informazioni, vedere Requisiti generali per RDS Custom per Oracle e Requisiti generali per RDS Custom per Oracle.	29 marzo 2023

Aggiornamento delle autorizzazioni delle policy AWS gestite	Le `AmazonRDSReadOnlyAccess` politiche `AmazonRDSFullAccess` and ora concedono autorizzazioni aggiuntive ad Amazon. CloudWatch Per ulteriori informazioni, consulta Aggiornamenti di Amazon RDS sulle policy gestite da AWS.	16 marzo 2023
RDS Proxy è disponibile nelle regioni della Cina	RDS Proxy è ora disponibile nelle regioni Cina (Pechino) e Cina (Ningxia). Per ulteriori informazioni sul proxy RDS, consulta Utilizzo di Server proxy per Amazon RDS.	15 marzo 2023
RDS Proxy è disponibile nella regione Asia Pacific (Giacarta)	RDS Proxy è ora disponibile nella regione Asia Pacific (Giacarta). Per ulteriori informazioni sul proxy RDS, consulta Utilizzo di Server proxy per Amazon RDS.	8 marzo 2023
Scritture ottimizzate per Amazon RDS migliora le prestazioni delle transazioni di scrittura per RDS per MariaDB	Puoi migliorare le prestazioni delle transazioni di scrittura per le istanze database RDS per MariaDB con Scritture ottimizzate per Amazon RDS. Per ulteriori informazioni, consulta l'argomento relativo alle prestazioni di scrittura migliorate con Scritture ottimizzate per Amazon RDS per MariaDB.	7 marzo 2023

Amazon RDS per PostgreSQL versioni 15.2	Le nuove funzionalità di Amazon RDS per PostgreSQL 15.2 includono il comando SQL standard "MERGE" per le query SQL condizionali, miglioramenti delle prestazioni per l'ordinamento in memoria e basato su disco e il supporto per il commit in due fasi e il filtro riga/colonna per la replica logica.	27 febbraio 2023
RDS Custom per Oracle è disponibile nelle regioni Canada (Centrale) e Sud America (San Paolo)	Per una tabella che mostra tutte le aree supportate Regioni AWS, consulta Regioni supportate e motori DB per RDS Custom for Oracle.	22 febbraio 2023
Amazon RDS supporta backup automatici tra regioni per RDS per MariaDB e RDS per MySQL	Ora puoi replicare gli snapshot di database e i log delle transazioni tra le Regioni AWS per le istanze database RDS per MariaDB e RDS per MySQL. Per ulteriori informazioni, consulta Replica dei backup automatici in un'altra Regione AWS.	22 febbraio 2023

Amazon RDS per Oracle supporta il preavviso degli aggiornamenti automatici della versione secondaria	RDS ti notifica in anticipo la data in cui sarà disponibile una nuova versione secondaria del motore RDS per Oracle. RDS inizia a pianificare gli aggiornamenti automatici della versione secondaria delle istanze database RDS per Oracle a partire dalla data di disponibilità. Per ulteriori informazioni, consulta Prima di pianificare un aggiornamento automatico della versione secondaria.	21 febbraio 2023

Amazon RDS per SQL Server supporta i flussi di attività del database	È ora possibile monitorare un'istanza database SQL Server utilizzando i flussi di attività del database. Un'istanza database SQL Server dispone dell'audit del server gestito da Amazon RDS. È possibile definire le policy per registrare gli eventi del server nelle specifiche di audit del server. È possibile creare una specifica di audit del database e definire le policy per registrare gli eventi del database. Il flusso di attività viene raccolto e trasmesso a Amazon Kinesis. Da Kinesis, puoi monitorare il flusso di attività per eseguire ulteriori analisi. Per ulteriori informazioni, consulta Monitoraggio di Amazon RDS tramite i flussi di attività del database.	15 febbraio 2023
RDS supporta MySQL 8.0.32 e 5.7.41	Ora puoi creare istanze database Amazon RDS che eseguono MySQL versione 8.0.32 e 5.7.41. Per ulteriori informazioni, consulta Versioni di MySQL in Amazon RDS.	7 febbraio 2023

Amazon RDS per Oracle supporta nuove suite di cifratura per SSL	Se esegui Oracle Database 19c o 21c, puoi specificare sei nuove suite di cifratura nell'opzione SSL per RDS per Oracle. Queste suite supportano FIPS e sono conformi agli standard FedRAMP. Per ulteriori informazioni, consulta Oracle Secure Sockets Layer.	3 febbraio 2023
Amazon RDS per Oracle supporta nuove suite di cifratura per Oracle Enterprise Manager	È possibile utilizzare quattro nuove suite di crittografia conformi agli standard FedRAMP per l'opzione OEM. Per ulteriori informazioni, consulta Oracle Management Agent per Enterprise Manager Cloud Control.	3 febbraio 2023
RDS per Oracle supporta i flussi di attività del database nelle regioni Asia Pacifico (Hyderabad), Europa (Spagna) e Medio Oriente (Emirati Arabi Uniti)	Per ulteriori informazioni, consulta Regioni supportate e motori DB per i flussi di attività del database in Amazon RDS.	27 gennaio 2023

Migrazione a un cluster di database multi-AZ PostgreSQL utilizzando una replica di lettura	Utilizzando una replica di lettura, puoi eseguire la migrazione di un'implementazione single-AZ o di un'implementazione di istanza database multi-AZ RDS per PostgreSQL a un'implementazione di cluster di database multi-AZ RDS per PostgreSQL con tempi di inattività ridotti. Per ulteriori informazioni, consulta Migrating to a Multi-AZ DB cluster using a read replica (Migrazione a un cluster di database multi-AZ tramite una replica di lettura).	23 gennaio 2023
Amazon RDS è disponibile nella Regione Asia Pacifico (Melbourne)	Amazon RDS è ora disponibile nella Regione Asia Pacifico (Melbourne). Per ulteriori informazioni, consulta Regioni e zone di disponibilità.	23 gennaio 2023

RDS per MariaDB supporta l'applicazione di connessioni SSL/TLS	RDS per MariaDB ora supporta l'applicazione delle connessioni SSL/TLS mediante l'impostazione del parametro `require_secure_transport` su `ON`. Per ulteriori informazioni, consulta Requiring SSL/TLS for all connections to a MariaDB DB instance (Richiesta dell'uso di SSL/TLS per tutte le connessioni a un'istanza database MariaDB).	19 gennaio 2023
Amazon RDS Optimized Reads migliora le prestazioni delle query per RDS per MariaDB	Puoi ottenere un'elaborazione delle query più rapida per le istanze database RDS per MariaDB con Amazon RDS Optimized Reads. Per ulteriori informazioni, consulta Improving query performance for RDS for MariaDB with Amazon RDS Optimized Reads (Prestazioni delle query migliorate per RDS per MariaDB con Amazon RDS Optimized Reads).	11 gennaio 2023

Ripristino di uno snapshot di cluster di database multi-AZ in un'istanza database	Ora puoi ripristinare uno snapshot di cluster di database multi-AZ in un'implementazione single-AZ o in un'implementazione di istanza database multi-AZ. Per ulteriori informazioni, consulta Restoring from a Multi-AZ DB cluster snapshot to a DB instance (Ripristino di uno snapshot di cluster di database multi-AZ a un'istanza database).	10 gennaio 2023
Specifica dell'autorità di certificazione (CA) durante la creazione dell'istanza database	Ora è possibile specifica re quale CA utilizzare per il certificato del server di un'istanza database durante la creazione dell'istanza database. Per ulteriori informazioni, consulta Certifica te authorities (Autorità di certificazione).	5 gennaio 2023

RDS Custom per SQL Server supporta le versioni del motore personalizzate	Una versione del motore personalizzata (CEV) per RDS Custom per SQL Server è una Amazon Machine Image (AMI) con Microsoft SQL Server preinstallato. Scegli un'AMI Windows di Amazon EC2 da utilizzare come immagine di base e puoi installare altro software sul sistema operativo. È possibile personalizzare la configura zione del sistema operativo e di SQL Server per soddisfar e le esigenze aziendali. Per ulteriori informazioni, consulta Working with custom engine versions for RDS Custom for SQL Server (Utilizzo di versioni del motore personali zzate per RDS Custom per SQL Server).	28 dicembre 2022
Uso delle implementazioni blu/verde Amazon RDS disponibili in altre Regioni AWS	La funzionalità implement azione blu/verde è ora disponibile nelle regioni Cina (Pechino) e Cina (Ningxia). Per ulteriori informazioni, consulta Utilizzo delle implementazioni blu/verde Amazon RDS per gli aggiornamenti del database.	22 dicembre 2022

Aggiornamento alle autorizzazioni del ruolo collegato ai servizi di IAM	La ServiceRolePolicy politica di AmazonRDS ora concede autorizzazioni aggiuntive per. AWS Secrets Manager Per ulteriori informazioni, consulta Aggiornamenti di Amazon RDS sulle policy gestite da AWS.	22 dicembre 2022
Amazon RDS supporta la ridenominazione di un cluster di database multi-AZ	Ora è possibile rinominare un cluster di database multi-AZ. Per ulteriori informazioni, consulta Renaming a Multi-AZ DB cluster (Ridenominazione di un cluster di database multi-AZ).	22 dicembre 2022
Amazon RDS si integra con la gestione AWS Secrets Manager delle password	Amazon RDS può gestire la password dell'utente master per un'istanza database o un cluster di database multi-AZ in Secrets Manager. Per ulteriori informazioni, consulta Password management with Amazon RDS and AWS Secrets Manager (Gestione delle password per Amazon RDS e AWS Secrets Manager).	22 dicembre 2022

Amazon RDS Optimized Writes supporta le classi di istanza database db.r6g e db.r6gd	Amazon RDS Optimized Writes ora supporta le classi di istanza database db.r6g e db.r6gd. Per ulteriori informazioni, consulta Improving write performance with Amazon RDS Optimized Writes (Prestazioni di scrittura migliorate con Amazon RDS Optimized Writes).	22 dicembre 2022
Amazon RDS Custom per Oracle supporta nuovi Regioni AWS	È possibile creare istanze database RDS Custom per Oracle nelle regioni Asia Pacifico (Seoul) e Asia Pacifico (Osaka-Locale). Per ulteriori informazioni, consulta Regioni supportate e motori DB per RDS Custom for Oracle.	21 dicembre 2022
Amazon RDS on AWS Outposts supporta le repliche di lettura	Ora è possibile creare una replica di lettura da un'istanza database MySQL o PostgreSQL di RDS su Outposts. Per ulteriori informazioni, consulta Creating read replicas for Amazon RDS on AWS Outposts (Creazione di repliche di lettura per Amazon RDS su AWS Outposts).	19 dicembre 2022

RDS Custom per Oracle supporta la modifica della classe di istanza database	Ora è possibile modificare la classe dell'istanza database RDS Custom per Oracle. Per ulteriori informazioni, consulta Modifying your RDS Custom for Oracle DB instance (Modifica dell'istanza database RDS Custom per Oracle).	16 dicembre 2022
RDS per MySQL e RDS per PostgreSQL supportano le classi di istanza database db.x2iedn	Ora è possibile utilizzare le classi di istanza database db.x2iedn per le istanze database RDS per MySQL e RDS per PostgreSQL. Per ulteriori informazioni, consulta Motori di database supportati per tutte le classi di istanza database disponibili.	14 dicembre 2022
Amazon RDS Optimized Writes supporta le classi di istanza database db.x2iedn	Amazon RDS Optimized Writes ora supporta le classi di istanza database db.x2iedn . Per ulteriori informazioni, consulta Improving write performance with Amazon RDS Optimized Writes (Prestazioni di scrittura migliorate con Amazon RDS Optimized Writes).	14 dicembre 2022

Amazon RDS supporta la copia di gruppi di opzioni database quando si copiano gli snapshot di database	Ora puoi copiare un gruppo di opzioni Account AWS come parte di una richiesta di copia istantanea sui database RDS for Oracle. Per ulteriori informazioni, consulta Considerazioni su gruppi di opzioni.	13 dicembre 2022
Amazon RDS supporta Server proxy per RDS con RDS per PostgreSQL versione 14	Ora è possibile creare un Server proxy per RDS con un database RDS per PostgreSQL versione 14. Per ulteriori informazioni sul proxy RDS, consulta Utilizzo di Server proxy per Amazon RDS.	13 dicembre 2022
Amazon RDS per Oracle supporta le classi di istanza db.x2idn, db.x2iedn e db.x2iezn	Ora è possibile utilizzare le classi di istanza db.x2idn, db.x2iedn e db.x2iezn per le istanze database Amazon RDS per Oracle. Per ulteriori informazioni, consulta Motori DB supportati per classi di istanza database e Classi di istanza RDS per Oracle supportate.	12 dicembre 2022

Le istanze database RDS per PostgreSQL supportano Trusted Language Extensions per PostgreSQL	Trusted Language Extensions per PostgreSQL è un kit di sviluppo open source che ti consente di creare estensioni di PostgreSQL ad alte prestazioni ed eseguirle in modo sicuro sulla tua istanza database RDS per PostgreSQL. Per ulteriori informazioni, consulta Working with Trusted Language Extensions for PostgreSQL (Utilizzo di Trusted Language Extensions per PostgreSQL).	30 novembre 2022
Uso delle implementazioni blu/verde Amazon RDS per gli aggiornamenti del database	È possibile apportare modifiche a un'istanza database in un ambiente di gestione temporanea e testarle senza influire sull'istanza database di produzione. Quando sei pronto, puoi promuovere l'ambiente di gestione temporanea nel nuovo ambiente di produzione, con tempi di inattività minimi. Per ulteriori informazioni, consulta Utilizzo delle implementazioni blu/verde Amazon RDS per gli aggiornamenti del database.	27 novembre 2022

Amazon RDS Optimized Writes migliora le prestazioni delle transazioni di scrittura per RDS per MySQL	Puoi migliorare le prestazioni delle transazioni di scrittura per le istanze database RDS per MySQL con Amazon RDS Optimized Writes. Per ulteriori informazioni, consulta l'argomento relativo alle prestazioni di scrittura migliorate con Scritture ottimizzate per Amazon RDS per MySQL.	27 novembre 2022
Amazon RDS Optimized Reads migliora le prestazioni delle query per RDS per MySQL	Puoi ottenere un'elaborazione delle query più rapida per le istanze database RDS per MySQL con Amazon RDS Optimized Reads. Per ulteriori informazioni, consulta Prestazioni delle query migliorate con Amazon RDS Optimized Reads.	27 novembre 2022
Amazon RDS è disponibile nella regione Asia Pacific (Hyderabad)	Amazon RDS ora è disponibile nella regione Asia Pacifico (Hyderabad). Per ulteriori informazioni, consulta Regioni e zone di disponibilità.	22 novembre 2022
RDS supporta MariaDB 10.6.11, 10.5.18, 10.4.27 e 10.3.37	Ora puoi creare istanze database Amazon RDS che eseguono MariaDB versione 10.6.11, 10.5.18, 10.4.27 e 10.3.37. Per ulteriori informazioni, consulta Versioni di MariaDB in Amazon RDS.	18 novembre 2022

RDS Custom per Oracle supporta l'impostazione di parametri di installazione non predefiniti in una versione del motore personalizzata (CEV)	Quando si crea una CEV, è possibile impostare valori non predefiniti per la base Oracle, la home Oracle, il nome e l'ID dell'utente UNIX e il nome e l'ID del gruppo UNIX. In questo modo, è possibile ottenere un maggiore controllo sull'installazione del database nell'istanza database RDS Custom per Oracle. Per ulteriori informazioni, consulta Preparazione del manifesto CEV.	18 novembre 2022
Amazon RDS supporta Oracle APEX versione 22.1.v1	È possibile utilizzare APEX 22.1.v1 con tutte le versioni supportate di Oracle Database. Per ulteriori informazioni, consulta Oracle Application Express.	18 novembre 2022
RDS per SQL Server supporta le repliche di lettura tra regioni	Ora è possibile creare una replica di lettura tra regioni per migliorare la capacità di ripristino di emergenza, ridurre la latenza di lettura delle applicazioni ed eseguire l'offload dei carichi di lavoro di lettura dall'istanza database primaria. Per ulteriori informazioni, vedere Creazione di una replica di lettura in un altro Regione AWS	16 novembre 2022

Amazon RDS è disponibile nella regione Europa (Spagna)	Amazon RDS ora è disponibile nella regione Europa (Spagna). Per ulteriori informazioni, consulta Regioni e zone di disponibilità.	16 novembre 2022
RDS per SQL Server supporta server collegati per database Oracle	Ora è possibile creare un server collegato per accedere a database Oracle esterni per leggere dati ed eseguire comandi SQL. Per ulteriori informazioni, consulta Linked Servers with Oracle OLEDB with RDS for SQL Server (Server collegati con Oracle OLEDB con RDS per SQL Server).	15 novembre 2022
RDS Custom per Oracle supporta Oracle Multitenant	È possibile creare un'istanza database RDS Custom per Oracle come database container (CDB). Dopo la creazione, il CDB contiene la root CDB, il seed PDB e un PDB. È possibile aggiungere altri PDB manualmente utilizzando Oracle SQL. Per ulteriori informazioni, consulta Panoramica dell'architettura Amazon RDS Custom per Oracle.	15 novembre 2022

Amazon RDS per Oracle supporta l'integrazione Amazon EFS	Se aggiungi l'opzione `EFS_INTEGRATION` al gruppo di opzioni, puoi trasferire i file tra l'istanza database RDS per Oracle e un file system Amazon EFS. Per ulteriori dettagli, consulta Amazon EFS.	15 novembre 2022
RDS supporta MySQL 8.0.31 e 5.7.40	Ora puoi creare istanze database Amazon RDS che eseguono MySQL versione 8.0.31 e 5.7.40. Per ulteriori informazioni, consulta Versioni di MySQL in Amazon RDS.	10 novembre 2022
Amazon RDS è disponibile nella regione Europa (Zurigo)	Amazon RDS è ora disponibile nella regione Europa (Zurigo). Per ulteriori informazioni, consulta Regioni e zone di disponibilità.	9 novembre 2022
L'accesso ai backup dei log delle transazioni è ora disponibile con RDS per SQL Server	Ora è possibile visualizzare e copiare i backup dei log delle transazioni del database in un bucket Amazon S3. Per ulteriori informazioni, consulta Accesso ai backup dei log delle transazioni.	7 novembre 2022
Cluster DB Multi-AZ supportati in aggiunta Regioni AWS	I cluster DB Multi-AZ sono ora disponibili in aggiunta. Regioni AWS Per ulteriori informazioni, consulta Regioni supportate e motori DB per cluster DB Multi-AZ in Amazon RDS.	4 novembre 2022

Amazon RDS supporta l'archiviazione gp3	Ora puoi creare istanze database Amazon RDS che utilizzano volumi di archiviazione Amazon EBS SSD per uso generico (gp3), che consentono di personalizzare le prestazioni di archiviazione indipendentemente dalla capacità di archiviazione. Per ulteriori informazioni, consulta Storage SSD per scopi generici.	4 novembre 2022
Amazon RDS supporta un nuovo evento per gli aggiornamenti del sistema operativo	Amazon RDS ora supporta un nuovo evento di istanza database, RDS-EVENT-0230, nella categoria degli eventi di patch di sicurezza. Questo nuovo evento avvisa quando è disponibile un aggiornamento del sistema operativo per l'istanza database. Per ulteriori informazioni, consulta Monitoraggio di eventi Amazon RDS e Utilizzo degli aggiornamenti del sistema operativo.	28 ottobre 2022

Amazon RDS per Oracle supporta le classi di istanza ottimizzata per la memoria r5b preconfigurate	Le classi di istanza database Oracle db.r5b sono ottimizzate per carichi di lavoro che richiedono memoria, archiviazioni e I/O aggiuntivi per vCPU. Ad esempio, db.r5b.4xlarge.tpc2.mem2x ha il multithreading attivato e fornisce il doppio della memoria rispetto a db.r5b.4xlarge. Per ulteriori informazioni, consulta Classi di istanza di RDS for Oracle.	27 ottobre 2022
Amazon RDS supporta 15 repliche di lettura per le istanze database MariaDB, MySQL e PostgreSQL	Ora è possibile creare fino a 15 repliche di lettura per le istanze database MariaDB, MySQL e PostgreSQL. Per ulteriori informazioni sulle repliche di lettura, consulta Uso di repliche di lettura.	20 ottobre 2022
Amazon RDS per PostgreSQL ora supporta PostgreSQL versione 15 RC 3 nell'ambiente di anteprima del database	La versione 15 Beta 3 di PostgreSQL è ora disponibile nell'ambiente di anteprima del database negli Stati Uniti orientali (Ohio). Regione AWS Per ulteriori informazioni, consultare Lavorare sull'ambiente di anteprima del database.	18 ottobre 2022

Amazon RDS supporta la configurazione automatica della connettività tra un database RDS e un'istanza EC2	Puoi utilizzare il AWS Management Console per configurare la connettività tra un'istanza DB RDS esistente o un cluster DB Multi-AZ e un'istanza EC2. Per ulteriori informazioni, consulta Connecting an EC2 instance and an RDS database automatically (Connessione automatica di un'istanza EC2 e un database RDS).	14 ottobre 2022
AWS Driver JDBC per PostgreSQL generalmente disponibile	Il driver AWS JDBC per PostgreSQL è un driver client progettato per RDS per PostgreSQL. Il driver JDBC di AWS per PostgreSQL è ora disponibile a livello generale. Per ulteriori informazioni, consulta Connessione con il driver AWS JDBC per PostgreSQL.	6 ottobre 2022
Amazon RDS per Oracle supporta Oracle APEX versione 21.2.v1	APEX 21.2 include la patch 33420059. Per ulteriori informazioni, consulta Requisiti di versione APEX.	3 ottobre 2022
RDS supporta MySQL 5.7.39	Ora puoi creare istanze database Amazon RDS che eseguono MySQL versione 5.7.39. Per ulteriori informazioni, consulta Versioni di MySQL in Amazon RDS.	29 settembre 2022

RDS supporta MariaDB versione 10.6.10	Ora puoi creare istanze database Amazon RDS che eseguono MariaDB versione 10.6.10. Per ulteriori informazioni, consulta Versioni di MariaDB in Amazon RDS.	29 settembre 2022
Server proxy per RDS supporta RDS per SQL Server	Ora puoi creare un Server proxy per RDS per un'istanza database RDS che esegue Microsoft SQL Server versione 2014 o successiva. Per ulteriori informazioni sul proxy RDS, consulta Utilizzo di Server proxy per Amazon RDS.	19 settembre 2022
Amazon RDS supporta MariaDB versione 10.5.17, 10.4.26 e 10.3.36	Ora puoi creare istanze database Amazon RDS che eseguono MariaDB versioni 10.5.17, 10.4.26 e 10.3.36 Per ulteriori informazioni, consulta Versioni di MariaDB in Amazon RDS.	15 settembre 2022

Amazon RDS per Oracle supporta l'archiviazione delle istanze locali per i dati temporanei	Ora puoi avviare Amazon RDS per Oracle sui tipi di istanza Amazon EC2 db.r5d e db.m5d con lo spazio di tabella temporaneo e Database Smart Flash Cache (la cache flash) configurati per utilizzare un archivio istanze. Archiviando i dati temporanei localmente, puoi ottenere latenze di lettura e scrittura inferiori rispetto all'archiviazione standard basata su Amazon EBS. Per ulteriori informazioni, consulta Storing temporary Oracle data in the instance store (Archiviazione dei dati Oracle temporanei nell'archivio istanze).	14 settembre 2022
Approfondimenti sulle prestazioni mostra le prime 25 query SQL	Nel pannello di controllo di Approfondimenti sulle prestazioni, la scheda Top SQL (Prime istruzioni SQL) mostra le 25 query SQL che contribuiscono di più al carico del database. Per ulteriori informazioni, consulta Panoramica della scheda Prime istruzioni SQL.	13 settembre 2022

RDS supporta MySQL 8.0.30	Ora puoi creare istanze database Amazon RDS che eseguono MySQL versione 8.0.30. Per ulteriori informazioni, consulta Versioni di MySQL in Amazon RDS.	9 settembre 2022
Amazon RDS è disponibile nella regione Medio Oriente (EAU)	Amazon RDS è ora disponibile nella regione Medio Oriente (EAU). Per ulteriori informazioni, consulta Regioni e zone di disponibilità.	30 agosto 2022
Amazon RDS per SQL Server supporta le sottoscrizioni e-mail SSRS	È ora possibile utilizzare l'estensione SQL Server Reporting Services (SSRS) Email per inviare report agli utenti e sottoscrivere i report sul server di report. Per ulteriori informazioni, consulta la sezione relativa al supporto per SQL Server Analysis Services in RDS per SQL Server.	26 agosto 2022
RDS per Oracle supporta i backup delle repliche di lettura	È possibile abilitare i backup automatici e creare snapshot manuali delle repliche di RDS per Oracle. Per ulteriori informazioni, consulta la pagina Utilizzo dei backup delle repliche RDS per Oracle.	23 agosto 2022

RDS per Oracle supporta lo switchover su Oracle Data Guard	Uno switchover è un'inversione di ruolo tra un database primario e una replica Oracle montata o aperta. Durante uno switchover, il database primario originale passa a un ruolo di standby, mentre il database in standby originale passa al ruolo primario. Per ulteriori informazioni, consulta la pagina Esecuzione di uno switchover su Oracle Data Guard.	23 agosto 2022
Amazon RDS supporta la configurazione automatica della connettività con un'istanza EC2	Quando crei un'istanza DB o un cluster DB Multi-AZ, puoi utilizzarli AWS Management Console per configurare la connettività tra un'istanza Amazon Elastic Compute Cloud e la nuova istanza DB o cluster DB. Per ulteriori informazioni, consulta Configurazione della connettività di rete automatica con un'istanza EC2 per una nuova istanza database e Configurazione della connettività di rete automatica con un'istanza EC2 per un nuovo cluster di database.	22 agosto 2022

RDS Custom per Oracle supporta la promozione delle repliche Oracle	Se si utilizza RDS Custom per Oracle, è possibile promuovere le repliche Oracle gestite utilizzando il comando CLI `promote-read-replica` . Inoltre, è possibile eliminare l'istanza database primario, in modo che RDS Custom per Oracle promuova le repliche Oracle gestite in istanze autonome. Per ulteriori informazioni, consulta Utilizzo delle repliche di RDS Custom per Oracle.	5 agosto 2022
RDS per MySQL supporta l'applicazione di connessioni SSL/TLS	RDS per MySQL ora supporta l'applicazione delle connessioni SSL/TLS mediante l'impostazione del parametro `require_secure_transport` su `ON`. Per ulteriori informazioni, consulta Richiesta di una connessione SSL/TLS a un'istanza database MySQL.	1 agosto 2022

Amazon RDS termina il supporto di Oracle Database 12c Release 1 (12.1.0.2)	Non viene più fornito il supporto della versione 12.1.0.2 per i modelli di licenza BYL e LI. A partire dal 1° agosto 2022, RDS per Oracle inizia a fornire aggiornamenti automatici per istanze database 12c Release 1 (12.1.0.2) e snapshot 12.1.0.2 ripristinati a Oracle Database 19c. Per ulteriori informazioni, consulta Oracle Database 12c con Amazon RDS e la pianificazione della fine del supporto in AWS re:Post.	1 agosto 2022
Il proxy RDS supporta RDS per MariaDB	È ora possibile creare un proxy RDS per un'istanza database RDS che esegue MariaDB versione 10.2, 10.3, 10.4 o 10.5. Il supporto per MariaDB è incluso nella famiglia di motori MySQL. Per ulteriori informazioni sul proxy RDS, consulta Utilizzo di Server proxy per Amazon RDS.	26 luglio 2022
Amazon RDS per MariaDB supporta le classi di istanza database db.r5b	Puoi ora creare istanze database RDS per MariaDB che eseguono classi di istanza database db.r5b. Per ulteriori informazioni, consulta Motori DB supportati per classi di istanza DB.	25 luglio 2022

RDS per Oracle supporta la modifica dei flussi di attività di database	Se si utilizza RDS per Oracle, è possibile modificare lo stato della policy di controllo di un flusso di attività di database impostandolo su "locked" (bloccato) (impostazione predefinita) o "unlocked" (sbloccato). Anziché interrompere un flusso di attività, è possibile sbloccare lo stato della relativa policy, personalizzare la policy di controllo e quindi bloccare di nuovo lo stato della policy. Per informazioni sui flussi di attività di database, consulta Modifica di un flusso di attività di database.	22 luglio 2022
Approfondimenti sulle prestazioni supporta la regione Asia Pacific (Giacarta)	In precedenza, non era possibile utilizzare Approfondimenti sulle prestazioni nella regione Asia Pacific (Giacarta). Questa limitazione è stata eliminata. Per ulteriori informazioni, consulta Regioni supportate e motori DB per Performance Insights in Amazon RDS.	21 luglio 2022

Microsoft SQL Server 2012 ha raggiunto la fine del supporto su Amazon RDS	Microsoft SQL Server 2012 ha raggiunto la fine del supporto e ciò coincide con il piano Microsoft per terminare il supporto esteso per questa versione il 12 luglio 2022. Qualsiasi istanza Microsoft SQL Server 2012 esistente deve essere aggiornata automaticamente all'ultima versione minore di Microsoft SQL Server 2014 a partire dal 1° giugno 2022. Per ulteriori informazioni, consulta Supporto Microsoft SQL Server 2012 su Amazon RDS.	12 luglio 2022
RDS supporta MariaDB 10.6.8, 10.5.16, 10.4.25, 10.3.35 e 10.2.44	Ora puoi creare istanze database Amazon RDS che eseguono MariaDB versioni 10.6.8, 10.5.16, 10.4.25, 10.3.35 e 10.2.44. Per ulteriori informazioni, consulta Versioni di MariaDB supportate in Amazon RDS.	8 luglio 2022

RDS Approfondimenti sulle prestazioni supporta periodi di conservazione aggiuntivi	In precedenza, Approfondimenti sulle prestazioni offriva solo due periodi di conservazione: 7 giorni (impostazione predefinita) o 2 anni (731 giorni). Ora, se devi mantenere i dati sulle prestazioni per più di 7 giorni, puoi specificare da 1 a 24 mesi. Per ulteriori informazioni, consulta Prezzi e conservazione dei dati per Approfondimenti sulle prestazioni.	1 luglio 2022
RDS Custom supporta le regioni Asia Pacifico (Mumbai) ed Europa (Londra)	Puoi creare istanze DB RDS Custom per Oracle e RDS Custom per SQL Server in due nuove istanze Regioni AWS: Asia Pacifico (Mumbai) ed Europa (Londra). Per ulteriori informazioni, consulta Supporto Regione AWS per RDS Custom per Oracle e Supporto Regione AWS per RDS Custom per SQL Server.	21 giugno 2022

RDS Custom per Oracle supporta Oracle Database 18c e 12c Release 2 (12.2)	Ora puoi creare un CEV per RDS Custom per Oracle utilizzando i file di installazione per Oracle Database 18c e 12c Release 2 (12.2). Puoi utilizzare questi CEV per creare un'istanza database di RDS Custom per Oracle. Per ulteriori informazioni, consulta Utilizzo di versioni del motore personalizzate per Amazon RDS Custom per Oracle.	21 giugno 2022
I cluster DB Multi-AZ supportano le classi di istanza database db.m5d e db.r5d	Ora puoi creare cluster DB Multi-AZ che utilizzano le classi di istanza database db.m5d e db.r5d. Per ulteriori informazioni, consulta Implementazioni cluster di database multi-AZ e Tipi di classi di istanza database.	21 giugno 2022
Cluster DB Multi-AZ disponibili in aggiunta Regioni AWS	Ora puoi creare cluster DB Multi-AZ nelle seguenti regioni: Europa (Francoforte) ed Europa (Stoccolma). Per ulteriori informazioni, consulta Implementazioni cluster di database multi-AZ.	21 giugno 2022

RDS per Microsoft SQL Server supporta la migrazione di database che utilizzano Transparent Data Encryption (TDE)	RDS per SQL Server supporta ora la migrazione di database Microsoft SQL Server con TDE attivato, utilizzando backup e ripristino nativi. Per ulteriori informazioni consulta Supporto per Transparent Data Encryption in SQL Server.	14 giugno 2022
Amazon RDS supporta la pubblicazione di eventi su argomenti Amazon SNS crittografati	Amazon RDS può ora pubblicare eventi su Servizio di notifica semplice Amazon (Amazon SNS) con crittografia lato server (SSE) abilitata, per una protezione aggiuntiva degli eventi che contengono dati sensibili. Per ulteriori informazioni, consulta Sottoscrizione alle notifiche eventi di Amazon Redshift.	1 giugno 2022
RDS supporta MySQL 5.7.38	Ora puoi creare istanze database di Amazon RDS che eseguono MySQL versione 5.7.38. Per ulteriori informazioni, consulta Versioni di MySQL in Amazon RDS.	31 maggio 2022

RDS per PostgreSQL supporta le repliche di lettura a cascata	Ora è possibile utilizzare repliche di lettura a cascata con RDS per PostgreSQL versione 14.1 e versioni successive. Per ulteriori informazioni, consulta la pagina Utilizzo delle repliche di lettura PostgreSQL in Amazon RDS.	4 maggio 2022
Amazon RDS on AWS Outposts supporta operazioni di storage su larga scala e scalabilità automatica	Ora è possibile modificare le dimensioni di archiviazione delle istanze database su Outpost e utilizzare la scalabilità automatica dell'archiviazione. Per maggiori informazioni, consultare Amazon RDS su AWS Outposts supporto per le funzionalità di Amazon RDS.	2 maggio 2022
Cluster DB Multi-AZ disponibili in aggiunta Regioni AWS	Ora puoi creare cluster di database Multi-AZ nelle seguenti regioni: Asia Pacifico (Singapore) e Asia Pacifico (Sydney). Per ulteriori informazioni, consulta Implementazioni cluster di database multi-AZ.	29 aprile 2022

Amazon RDS supporta la modalità dual-stack	Ora è possibile eseguire le istanze database in modalità dual-stack. In modalità dual-stack, le risorse possono comunicare con l'istanza database tramite IPv4, IPv6 o entrambi. Per ulteriori informazioni, consulta Assegnazione di indirizzi IP con Amazon RDS.	29 aprile 2022
Amazon RDS pubblica i parametri di utilizzo su Amazon CloudWatch	Il `AWS/Usage` namespace in Amazon CloudWatch include parametri di utilizzo a livello di account per le quote dei servizi Amazon RDS. Per ulteriori informazioni, consulta i parametri di CloudWatch utilizzo di Amazon per Amazon RDS.	28 aprile 2022
Amazon RDS for MySQL supporta le classi di istanze database db.m6i e db.r6i	Ora puoi utilizzare le classi di istanze database db.m6i e db.r6i per le istanze database Amazon RDS che eseguono MySQL. Per ulteriori informazioni, consulta Motori di database supportati per tutte le classi di istanza database disponibili.	28 aprile 2022

Amazon RDS for PostgreSQL supporta le classi di istanze database db.m6i e db.r6i	Ora puoi utilizzare le classi di istanze database db.m6i e db.r6i per le istanze database Amazon RDS che eseguono PostgreSQL.. Per ulteriori informazioni, consulta Motori di database supportati per tutte le classi di istanza database disponibili.	27 aprile 2022
Amazon RDS for MariaDB supporta le classi di istanze database db.m6i e db.r6i	Ora puoi utilizzare le classi di istanze database db.m6i e db.r6i per le istanze database Amazon RDS che eseguono MariaDB. Per ulteriori informazioni, consulta Motori di database supportati per tutte le classi di istanza database disponibili.	26 aprile 2022
Amazon RDS on AWS Outposts supporta implementazioni Multi-AZ	Ora puoi creare un'istanza database in standby su un Outpost diverso. Per ulteriori informazioni, consulta Amazon RDS sul AWS Outposts supporto per le funzionalità di Amazon RDS.	19 aprile 2022
Amazon RDS for Oracle supporta le classi di istanze database db.m6i e db.r6i	Se esegui Oracle Database 19c, puoi utilizzare classi di istanze db.m6i e db.r6i. Le classi di istanze db.m6i sono per uso generico e sono ideali per un'ampia gamma di carichi di lavoro. Per ulteriori informazioni, consulta Classi di istanza di RDS for Oracle.	8 aprile 2022

Amazon RDS for SQL Server supporta la replica dei processi SQL Server Agent	Quando attivi questa funzionalità, i processi di SQL Server Agent creati, modificati o eliminati sull'host principale, vengono sincronizzati automaticamente con l'host secondario in una configurazione Multi-AZ. Per ulteriori informazioni, consulta Utilizzo di SQL Server Audit.	7 aprile 2022
Amazon RDS supporta RDS Proxy con RDS per PostgreSQL versione 13	Ora è possibile creare un proxy RDS con un database RDS for PostgreSQL versione 13. Per ulteriori informazioni sul proxy RDS, consulta Amazon RDS Proxy.	4 aprile 2022
Amazon RDS prevede di rendere obsoleto Oracle Database 12c	Oracle Database 12c sta per essere reso obsoleto. Oracle Corporation non fornirà più patch per le versioni di Oracle Database 12c dopo tali date. end-of-support Amazon RDS prevede di iniziare l'aggiornamento automatico delle istanze database di Oracle Database 12c a Oracle Database 19c. Per ulteriori informazioni, consultare Oracle Database 12c con Amazon RDS e Preparazione per l'aggiornamento automatico di Oracle Database 12c.	22 marzo 2022

Note di rilascio di Amazon RDS for PostgreSQL	Ora è disponibile una guida separata per le note di rilascio di Amazon RDS for PostgreSQL. Per ulteriori informazioni, consultare le Note di rilascio di Amazon RDS for PostgreSQL.	22 marzo 2022
Note di rilascio di Amazon RDS for Oracle	Ora è disponibile una guida separata per le note di rilascio di Amazon RDS for Oracle. Per ulteriori informazioni, consultare le Note di rilascio di Amazon RDS for Oracle.	22 marzo 2022
Cluster DB Multi-AZ disponibili in aggiunta Regioni AWS	Ora puoi creare cluster di database Multi-AZ nelle seguenti regioni: Stati Uniti orientali (Ohio) e Asia Pacifico (Tokyo). Per ulteriori informazioni, consulta Implementazioni cluster di database multi-AZ.	15 marzo 2022

Amazon RDS for PostgreSQL versioni 14.2, 13.6, 12.10, 11.15 e 10.20	RDS for PostgreSQL ora supporta le versioni 14.2, 13.6, 12.10, 11.15 e 10.20. Le versioni 14.2 e 13.6 aggiungono il supporto per due nuovi wrapper di dati esterni. L'estensione `mysql_fdw` consente a PostgreSQL di lavorare con i dati memorizzati nei database MySQL, MariaDB e Aurora MySQL. L'estensione `tds_fdw` consente a PostgreSQL di lavorare con i dati memorizzati nei database SQ Server. Per ulteriori informazioni, consulta Versioni database di PostgreSQL supportate.	12 marzo 2022
RDS supporta MySQL 5.7.37	Ora puoi creare istanze database di Amazon RDS che eseguono MySQL versione 5.7.37. Per ulteriori informazioni, consulta Versioni di MySQL in Amazon RDS.	11 marzo 2022
Amazon RDS for SQL Server supporta nuove classi di istanze database	Ora puoi creare istanze database Amazon RDS che eseguono Microsoft SQL Server che utilizzano classi di istanze database db.m6i e db.r6i. Per ulteriori informazioni, consulta Supporto di classe istanza database per Microsoft SQL Server.	9 marzo 2022

Amazon RDS for Oracle supporta Oracle Database 21c	Ora puoi creare istanze database di Amazon RDS che eseguono Oracle Database 21c (21.0.0.0). Questa è la prima release di Oracle Database che supporta solo l'architettura multitenant (CDB). Per maggiori informazioni, consulta Oracle Database 21c con Amazon RDS.	7 marzo 2022
RDS supporta MariaDB 10.6.7, 10.5.15, 10.4.24, 10.3.34 e 10.2.43	Puoi ora creare istanze database Amazon RDS che eseguono MariaDB versioni 10.6.7, 10.5.15, 10.4.24, 10.3.34 e 10.2.43. Per ulteriori informazioni, consulta Versioni di MariaDB in Amazon RDS.	3 marzo 2022
AWS Driver JDBC per MySQL generalmente disponibile	Il driver AWS JDBC per MySQL è un driver client progettato per RDS for MySQL. Il driver AWS JDBC per MySQL è ora disponibile a livello generale. Per maggiori informazioni, consulta Connessione con il driver JDBC per MySQL di Amazon Web Services.	2 marzo 2022

Cluster di database Multi-AZ generalmente disponibili	Un'implementazione cluster di database Multi-AZ è una modalità d'implementazione ad alta disponibilità di Amazon RDS con due istanze database in standby leggibili. Cluster di database Multi-AZ sono ora generalmente disponibili. Per ulteriori informazioni, consulta Implementazioni cluster di database multi-AZ.	1 marzo 2022
RDS supporta MySQL 8.0.28	Ora puoi creare istanze database di Amazon RDS che eseguono MySQL versione 8.0.28. Per ulteriori informazioni, consulta Versioni di MySQL in Amazon RDS.	28 febbraio 2022
Amazon RDS for Oracle supporta nuove impostazioni per Native Network Encryption (NNE)	Per controllare se i client possono connettersi con metodi di crittografia e checksum non sicuri, imposta `SQLNET.ALLOW_WEAK_CRYPTO_CLIENTS` e `SQLNET.ALLOW_WEAK_CRYPTO` nell'opzione NNE. Esempi di metodi non sicuri includono DES, 3DES, RC4 e MD5. Per ulteriori informazioni, consulta Impostazioni dell'opzione NNE.	25 febbraio 2022

Amazon RDS for SQL Server supporta i gruppi di disponibilità Always On per Microsoft SQL Server 2017 Standard Edition	Quando si crea un'istanza database utilizzando la configurazione multi-AZ in SQL Server 2017 Standard Edition 14.00.3401.7 e versioni successive, RDS utilizza automaticamente i gruppi di disponibilità. Per ulteriori informazioni, consulta Implementazioni Multi-AZ per Microsoft SQL Server.	18 febbraio 2022
RDS for Oracle supporta i flussi di attività di database nella regione Asia Pacifico (Jakarta)	Per ulteriori informazioni, consulta Supporto Regioni AWS per i flussi di attività del database.	16 febbraio 2022
Supporto Amazon RDS Custom per Oracle Database 12.1	Ora è possibile creare versioni personalizzate del motore per RDS Custom per Oracle che utilizzano Oracle Database 12.1 Enterprise Edition. Per ulteriori informazioni, consulta Utilizzo di versioni del motore personalizzate per Amazon RDS Custom per Oracle.	4 febbraio 2022
Amazon RDS for MariaDB supporta una nuova versione principale	Ora puoi creare istanze database Amazon RDS che eseguono MariaDB versione 10.6. Per ulteriori informazioni, consulta Supporto per MariaDB 10.6 in Amazon RDS.	3 febbraio 2022

Approfondimenti sulle prestazioni supporta l'acquisizione del piano per le query Oracle	La console Approfondimenti sulle prestazioni supporta una nuova dimensione del piano per le prime istruzioni SQL. Quando esegui la divisione per piano, puoi vedere quali piani stanno utilizzando le query Oracle principali. Se una query utilizza più piani, puoi confrontare i piani affiancati nella console e determinare quale piano è più efficiente. Puoi anche eseguire il drill-down per vedere quali passaggi di un piano hanno il costo più alto. Per ulteriori informazioni, consulta Analisi dei piani di esecuzione di Oracle utilizzando il pannello di controllo di Approfondimenti sulle prestazioni.	27 gennaio 2022

Approfondimenti sulle prestazioni supporta nuove API	Approfondimenti sulle prestazioni supporta le seguenti API: `GetResourceMetadata` , `ListAvailableResourceDimensions` e `ListAvailableResourceMetrics` . Per ulteriori informazioni, consultare Recupero dei parametri con l'API Approfondimenti sulle prestazioni in questo manuale e la Documentazione di riferimento dell'API di Amazon RDS Approfondimenti sulle prestazioni.	12 gennaio 2022
RDS Proxy supporta gli eventi	RDS Proxy ora genera eventi a cui puoi abbonarti e visualizzarli in CloudWatch Eventi o configurare per l'invio ad Amazon EventBridge. Per maggiori informazioni, consulta Utilizzo degli eventi RDS Proxy.	11 gennaio 2022
Amazon RDS for SQL Server supporta SSAS in modalità multidimensionale	RDS for SQL Server supporta l'esecuzione di SQL Server Analysis Services (SSAS) in modalità tabulare o multidimensionale. Per ulteriori informazioni, consulta Supporto per SQL Server Analysis Services in RDS for SQL Server.	7 gennaio 2022

Proxy RDS disponibile in aggiunta Regioni AWS	RDS Proxy è ora disponibile nelle seguenti regioni: Africa (Città del Capo), Asia Pacifico (Hong Kong), Asia Pacifico (Osaka), Europa (Milano), Europa (Parigi), Europa (Stoccolma), Medio Oriente (Bahrein) e Sud America (San Paolo). Per ulteriori informazioni sul proxy RDS, consulta Amazon RDS Proxy.	5 gennaio 2022
RDS supporta MySQL 8.0.27	Ora puoi creare istanze database di Amazon RDS che eseguono MySQL versione 8.0.27. Per ulteriori informazioni, consulta Versioni di MySQL in Amazon RDS.	21 dicembre 2021
Amazon RDS è disponibile nella regione Asia Pacific (Giacarta)	Amazon RDS è ora disponibile nella regione Asia Pacifico (Jakarta). Per ulteriori informazioni, consultare Regioni e zone di disponibilità.	13 dicembre 2021
Amazon RDS supporta MariaDB nella versione 10.5.13, 10.4.22, 10.3.32 e 10.2.41	Puoi ora creare istanze database Amazon RDS che eseguono MariaDB versioni 10.5.13, 10.4.22, 10.3.32 e 10.2.41. Per ulteriori informazioni, consulta Versioni di MariaDB in Amazon RDS.	8 dicembre 2021

Amazon RDS Custom per SQL Server	Amazon RDS Custom è un servizio di database gestito per applicazioni legacy, personalizzate e in pacchetti che richiedono l'accesso al sistema operativo e all'ambiente di database sottostanti. Con Amazon RDS Custom, si ottiene l'automazione di Amazon RDS e la flessibilità di Amazon EC2. Per ulteriori informazioni, consultare Utilizzo di Amazon RDS Custom.	1° dicembre 2021
Cluster database Multi-AZ (anteprima)	Ora è possibile creare cluster di database Multi-AZ per RDS for MySQL e RDS for PostgreSQL. Un'implementazione cluster di database Multi-AZ è una modalità d'implementazione ad alta disponibilità di Amazon RDS con due istanze database in standby leggibili. I cluster di database Multi-AZ sono nell'anteprima. Per ulteriori informazioni, consultare Implementazioni cluster di database Multi-AZ (anteprima).	23 novembre 2021

Amazon RDS supporta RDS Proxy con RDS per PostgreSQL versione 12	Ora è possibile creare un RDS Proxy con un database RDS for PostgreSQL versione 12. Per ulteriori informazioni sul RDS Proxy, consultare Utilizzo di Amazon RDS Proxy.	22 novembre 2021
Amazon RDS on AWS Outposts supporta i backup locali	Puoi archiviare backup automatici e istantanee manuali nel tuo Outpost Regione AWS o localmente. Per ulteriori informazioni, consulta Amazon RDS sul AWS Outposts supporto per le funzionalità di Amazon RDS.	22 novembre 2021
Supporto Amazon RDS per più account AWS KMS keys	Puoi utilizzare una chiave KMS da un altro AWS account per la crittografia durante l'esportazione di snapshot DB su Amazon S3. Per ulteriori informazioni, consulta Esportazione dei dati dello snapshot DB su Simple Storage Service (Amazon S3).	3 novembre 2021
Amazon RDS on AWS Outposts supporta la pubblicazione dei log del motore di database su Logs CloudWatch	RDS on Outposts ora supporta la pubblicazione dei log del motore di database su CloudWatch Logs. Per ulteriori informazioni, consulta il supporto di Amazon RDS on AWS Outposts per le funzionalità di Amazon RDS.	2 novembre 2021

Amazon RDS Custom per Oracle	Amazon RDS Custom è un servizio di database gestito per applicazioni legacy, personalizzate e in pacchetti che richiedono l'accesso al sistema operativo e all'ambiente di database sottostanti. Con Amazon RDS Custom, si ottiene l'automazione di Amazon RDS e la flessibilità di Amazon EC2. Per ulteriori informazioni, consultare Utilizzo di Amazon RDS Custom.	26 ottobre 2021
Supporto per la replica ritardata per RDS for MySQL versione 8.0	A partire da RDS for MySQL versione 8.0.26, è possibile configurare la replica ritardata per le istanze database RDS for MySQL versione 8.0. Per ulteriori informazioni, consulta la sezione Configurazione della funzione di replica ritardata con MySQL.	25 ottobre 2021
Supporto per MySQL 8.0.26	Ora puoi creare istanze database di Amazon RDS che eseguono MySQL versione 8.0.26. Per ulteriori informazioni, consulta Versioni di MySQL in Amazon RDS.	25 ottobre 2021

Supporto per la replica basata su GTID per RDS for MySQL versione 8.0	A partire da RDS for MySQL versione 8.0.26, è possibile configurare la replica basata su GTID per le istanze database RDS for MySQL versione 8.0. Per ulteriori informazioni, consulta Utilizzo della replica basata su GTID per RDS for MySQL.	25 ottobre 2021
Amazon RDS supporta RDS Proxy con RDS for MySQL 8.0	Ora è possibile creare un RDS Proxy per un'istanza database RDS for MySQL 8.0. Per ulteriori informazioni, consultar e Utilizzo Proxy di Amazon RDS.	21 ottobre 2021
Amazon RDS on AWS Outposts supporta versioni RDS aggiuntive per MySQL	RDS su Outposts ora supporta RDS per MySQL versioni 8.0.23 e 8.0.25. Per ulteriori informazioni, consulta il supporto di Amazon RDS on AWS Outposts per le funzionalità di Amazon RDS.	20 ottobre 2021
Amazon RDS for PostgreSQL ora supporta PostgreSQL versione 14 RC 1 nell'ambiente di anteprima del database	La versione 14 RC 1 di PostgreSQL è ora disponibile nell'ambiente di anteprima del database negli Stati Uniti orientali (Ohio). Regione AWS Per ulteriori informazioni, consultare Lavorare sull'ambiente di anteprima del database.	19 ottobre 2021

Amazon RDS supporta Performance Insights in aggiunta Regioni AWS	Approfondimenti sulle prestazioni è disponibile nelle regioni Medio Oriente (Bahrein), Africa (Città del Capo), Europa (Milano) e Asia Pacifico (Osaka). Per ulteriori informazioni, consulta Regioni supportate e motori DB per Performance Insights in Amazon RDS.	5 ottobre 2021
Approfondimenti sulle prestazioni supporta statistiche a livello digest per Oracle	Quando utilizzi Approfondimenti sulle prestazioni, puoi visualizzare le statistiche SQL sia a livello di istruzione che digest per Amazon RDS for Oracle. Per ulteriori informazioni, consulta Analisi delle query in esecuzione in Oracle.	4 ottobre 2021
Amazon RDS on AWS Outposts supporta versioni RDS aggiuntive per PostgreSQL	RDS su Outposts ora supporta RDS for PostgreSQL versioni 12.8 e 13.4. Per ulteriori informazioni, consulta il supporto di Amazon RDS on AWS Outposts per le funzionalità di Amazon RDS.	1° ottobre 2021
Amazon RDS supporta Oracle APEX versione 21.1.v1	È possibile utilizzare APEX 21.1.v1 con tutte le versioni supportate di Oracle Database. Per ulteriori informazioni, consulta Oracle Application Express.	24 settembre 2021

Amazon RDS for Oracle supporta la crittografia lato client per NNE	Quando si configura NNE, è possibile evitare di forzare la crittografia sul lato server. Ad esempio, è possibile che non si desideri forzare tutte le comunicazioni client a utilizzar e la crittografia perché il server lo richiede. In questo caso, è possibile forzare la crittogra fia sul lato client utilizzando le opzioni `SQLNET.*CLIENT`. Per ulteriori informazioni, consulta Native Network Encryption di Oracle.	24 settembre 2021
Amazon RDS for MySQL e RDS for PostgreSQL supporta le nuove classi di istanza database	Ora è possibile utilizzare le classi di istanza db.r5b, db.t4g, e db.x2g.per creare istanze database Amazon RDS che eseguono MySQL o PostgreSQL. Per ulteriori informazioni, consulta Motori di database supportati per tutte le classi di istanza database disponibili.	15 settembre 2021
Amazon RDS for Microsoft SQL Server supporta Java Database Connectivity (JDBC) con Microsoft Distributed Transaction Coordinator (MSDTC)	Transazioni XA JDBC sono ora supportate con MSDTC per SQL Server 2017 versione 14.00.3223.3 e versioni successive e SQL Server 2019. Per ulteriori informazi oni, consulta supporto per Microsoft Distributed Transacti on Coordinator in RDS for SQL Server.	7 settembre 2021

Amazon RDS supporta MariaDB versione 10.5.12, 10.4.21, 10.3.31, e 10.2.40	Puoi ora creare istanze database Amazon RDS che eseguono MariaDB versioni 10.5.12, 10.4.21 e 10.2.40. Per ulteriori informazioni, consulta Versioni di MariaDB in Amazon RDS.	2 settembre 2021
Amazon RDS ha terminato il supporto per Oracle Database 18c	È possibile creare istanze database solo per Oracle Database 12c e Oracle Database 19c. Se si dispone di snapshot di Oracle Database 18c, aggiornarli a una versione successiv a. Per ulteriori informazioni, consulta Aggiornamento di uno snapshot DB Oracle.	17 agosto 2021
Amazon RDS for SQL Server non supporta aggiornamenti a versioni secondarie automatic he	È ora possibile aggiornare automaticamente le istanze database RDS per SQL Server alla versione secondaria più recente. Per informazioni, consulta Aggiornamento del motore del database Microsoft SQL Server.	13 agosto 2021

Amazon RDS for PostgreSQL oggi supporta PostgreSQL versione 14 Beta 2 nell'ambiente di anteprima del database	Per ulteriori informazioni su PostgreSQL versione 14 beta 1, consulta le note di rilascio di PostgreSQL 14 beta 1. Per ulteriori informazioni su PostgreSQL versione 14 beta 2, consulta le note di rilascio di PostgreSQL 14 beta 2. Per informazioni sull'ambiente di anteprima del database, consulta Lavorare con l'ambiente di anteprima del database.	9 agosto 2021
Amazon RDS supporta RDS Proxy in un VPC condiviso	Ora è possibile creare un proxy RDS in un VPC condiviso. Per maggiori informazioni su RDS Proxy, consulta la sezione "Gestione delle connessioni con Amazon RDS Proxy" nella Guida per l'utente di Amazon RDS o nella Guida per l'utente di Aurora.	6 agosto 2021
Amazon RDS supporta MariaDB versione 10.2.39	Ora puoi creare istanze database Amazon RDS che eseguono MariaDB versione 10.2.39. Per ulteriori informazioni, consulta Versioni di MariaDB in Amazon RDS.	4 agosto 2021

Amazon RDS for Oracle aggiunge l'opzione `TIMEZONE_FILE_AUTOUPGRADE`	Con questa opzione, è possibile aggiornare il file del fuso orario corrente alla versione più recente dell'istanza database Oracle. Per ulteriori informazioni, consulta Aggiornamento automatico del file del fuso orario Oracle.	30 luglio 2021
Amazon RDS estende il supporto ai backup automatizzati tra regioni	Ora puoi replicare gli snapshot di database e i log delle transazioni tra più Regioni AWS. Per ulteriori informazioni, consulta Replicazione dei backup automatici in un'altra regione. AWS	19 luglio 2021
Supporto per MySQL 5.7.34	Ora puoi creare istanze database di Amazon RDS che eseguono MySQL versione 5.7.34. Per ulteriori informazioni, consulta Versioni di MySQL in Amazon RDS.	8 luglio 2021
Amazon RDS on AWS Outposts supporta versioni RDS aggiuntive per PostgreSQL	RDS su Outposts ora supporta RDS for PostgreSQL versioni 12.7 e 13.3. Per ulteriori informazioni, consulta il supporto di Amazon RDS on AWS Outposts per le funzionalità di Amazon RDS.	8 luglio 2021

Amazon RDS for PostgreSQL supporta oracle_fdw	È ora possibile utilizzare l'estensione oracle_fdw per fornire un wrapper di dati esterno per l'accesso ai database Oracle. Per ulteriori informazioni, consulta Accesso ai dati esterni con l'estensione oracle_fdw.	8 luglio 2021
Amazon RDS supporta Oracle Management Agent (OMA) versione 13.5	È possibile utilizzare Oracle Management Agent (OMA) versione 13.5 con Oracle Enterprise Manager (OEM) Cloud Control 13c Release 5 e versioni successive. Amazon RDS for Oracle installa OMA, che è in grado di comunicar e con Oracle Managemen t Service (OMS) per fornire informazioni di monitoraggio. Se si esegue OMS 13.5, è possibile gestire i database installando OMA 13.5. Per ulteriori informazioni, consulta Oracle Management Agent per Enterprise Manager Cloud Control.	7 luglio 2021

Amazon RDS for Oracle supporta il download dei log da Simple Storage Service (Amazon S3)	Se i log non sono presenti nell'istanza ma sono protetti dal tempo di conservazione del backup, utilizza `rdsadmin.rdsadmin_archive_log_download` per scaricarli di nuovo da Simple Storage Service (Amazon S3). RDS per Oracle salva i log nella directory `/rdsdbdata/log/arch` nell'istanza database. Per ulteriori informazioni, consulta Download dei log di ripristino archiviati da Simple Storage Service (Amazon S3).	2 luglio 2021
Amazon RDS supporta MariaDB versione 10.4.18 e 10.5.9	Puoi ora creare istanze database Amazon RDS che eseguono MariaDB versioni 10.4.18 e 10.5.9. Per ulteriori informazioni, consulta Versioni di MariaDB in Amazon RDS.	30 giugno 2021

Amazon RDS for Oracle supporta i flussi di attività database	È ora possibile monitorare un'istanza database Oracle utilizzando flussi di attività del database. Un database Oracle scrive record di verifica sul percorso di verifica unificata . Quando si avvia un flusso di attività del database su un'istanza database Oracle, Amazon Kinesis trasmette tutte le attività che corrispon dono alle policy di verifica di Oracle Database. Per ulteriori informazioni, consulta Monitoraggio di Amazon RDS tramite i flussi di attività del database.	23 giugno 2021
Amazon RDS for Oracle introduce classi di istanze ottimizzate per la memoria	Le nuove classi di istanza Oracle DB sono ottimizza te per carichi di lavoro che richiedono memoria, storage e I/O aggiuntivi per vCPU. Per ulteriori informazioni, consulta Classi di istanza di RDS for Oracle.	23 giugno 2021
Supporto per MySQL 8.0.25	Ora puoi creare istanze database di Amazon RDS che eseguono MySQL versione 8.0.25. Per ulteriori informazi oni, consulta Versioni di MySQL in Amazon RDS.	18 giugno 2021

Amazon RDS on AWS Outposts supporta versioni RDS aggiuntive per PostgreSQL	RDS su Outposts ora supporta RDS per PostgreSQL versioni 12.5, 12.6, 13.1 e 13.2. Per ulteriori informazioni, consulta il supporto di Amazon RDS on AWS Outposts per le funzionalità di Amazon RDS.	28 maggio 2021
Amazon RDS supporta MariaDB versione 10.2.37 e 10.3.28	Puoi ora creare istanze database Amazon RDS che eseguono MariaDB versioni 10.2.37 e 10.3.28. Per ulteriori informazioni, consulta Versioni di MariaDB in Amazon RDS.	27 maggio 2021
Amazon RDS for Oracle supporta il database di container multitenant (CDB)	Un'architettura multitenant consente a un database Oracle di essere un CDB. In Oracle Database 19c, il CDB può includere un singolo PDB. L'esperienza utente con un PDB è per lo più identica all'esperienza utente con un non CDB. Per ulteriori informazioni, consulta architettura RDS for Oracle.	25 maggio 2021
Amazon RDS su AWS Outposts supporta Amazon RDS per SQL Server	RDS su Outposts ora supporta Amazon RDS for SQL Server. Per ulteriori informazioni, consulta il supporto di Amazon RDS on AWS Outposts per le funzionalità di Amazon RDS.	11 maggio 2021

Amazon RDS estende il supporto ai backup automatizzati tra regioni	Ora puoi configurare istanze di database Amazon RDS che eseguono Microsoft SQL Server per replicare snapshot DB e log delle transazioni in una regione diversa. AWS Per ulteriori informazioni, consulta Replicazione dei backup automatici in un'altra regione. AWS	7 maggio 2021
Amazon RDS supporta backup automatizzati tra regioni per istanze di database crittografate	È ora possibile replicare gli snapshot DB e i log delle transazioni in una regione AWS diversa per istanze di database Amazon RDS crittografate che eseguono Oracle o PostgreSQL. Per ulteriori informazioni, vedere Replica dei backup automatici in un'altra regione. AWS	3 maggio 2021
Amazon RDS on AWS Outposts supporta il monitoraggio di Amazon CloudWatch	RDS on Outposts ora supporta il monitoraggio di Amazon CloudWatch . Per ulteriori informazioni, consulta il supporto di Amazon RDS on AWS Outposts per le funzionalità di Amazon RDS.	21 aprile 2021

RDS per PostgreSQL supporta le funzioni Lambda AWS	Ora puoi richiamare le funzioni AWS Lambda per le tue istanze DB RDS per PostgreSQL. Per ulteriori informazioni, consulta Richiamo di una funzione AWS Lambda da una istanza database RDS for PostgreSQL.	13 Aprile 2021
RDS per SQL Server supporta gli eventi estesi	È possibile utilizzare gli eventi estesi di SQL Server per acquisire informazioni di debug e risoluzione dei problemi. Per ulteriori informazioni, consulta Utilizzo di eventi estesi con Amazon RDS for Microsoft SQL Server.	8 aprile 2021
Supporto per MySQL 8.0.23, 5.7.33 e 5.6.51	Ora puoi creare istanze database di Amazon RDS che eseguono MySQL versione 8.0.23, 5.7.33 e 5.6.51. Per ulteriori informazioni, consulta Versioni di MySQL in Amazon RDS.	31 marzo 2021

Rollback automatico su aggiornamento non riuscito Amazon RDS for MySQL	Se un aggiornamento dell'istanza database da MySQL versione 5.7 a MySQL versione 8.0 non riesce, Amazon RDS esegue automaticamente il rollback delle modifiche eseguite per l'aggiornamento. Dopo il rollback, l'istanza database MySQL esegue MySQL versione 5.7. Per ulteriori informazioni, consulta Rollback dopo l'errore di aggiornamento da MySQL 5.7 a 8.0.	18 marzo 2021
Amazon RDS supporta repliche di lettura tra regioni nelle regioni opt-in	È ora possibile replicare istanze database in regioni opt-in. Per ulteriori informazioni, consulta Creazione di una replica di lettura in un'altra regione. AWS	18 marzo 2021

Amazon RDS prevede di rendere obsoleto Oracle Database 18c	Oracle Database 18c (18.0.0.0) sta per essere reso obsoleto. Oracle Corporation non fornirà più patch per Oracle Database 18c dopo tale data. end-of-support Il 1 luglio 2021, Amazon RDS prevede di iniziare l'aggiornamento automatico delle istanze di Oracle Database 18c a Oracle Database 19c. Prima dell'inizio degli aggiornamenti automatici, è consigliabile aggiornare manualmente le istanze di Oracle Database 18c esistenti a Oracle Database 19c. Per ulteriori informazioni, consulta Preparazione per l'aggiornamento automatico di Oracle Database 18c.	11 marzo 2021
Amazon RDS ha terminato il supporto per Oracle Database 11g	È possibile creare solo istanze database per Oracle Database 12c Release 1 (12.1.0.2) e versioni successive. Se si dispone di snapshot di Oracle Database 11g, aggiornarli a una versione successiva. Per ulteriori informazioni, consulta Aggiornamento di uno snapshot DB Oracle.	11 marzo 2021

Amazon RDS supporta backup continui di istanze DB in AWS Backup	Ora puoi creare backup automatici AWS Backup e ripristinare istanze DB da questi backup fino a un orario specificato. Per ulteriori informazioni, consulta Utilizzare per AWS Backup gestire i backup automatici.	10 marzo 2021
Amazon RDS supporta Oracle Management Agent (OMA) versione 13.4	È possibile utilizzare Oracle Management Agent (OMA) versione 13.4 con Oracle Enterprise Manager (OEM) Cloud Control 13c Release 4 Update 9. Amazon RDS for Oracle installa OMA, che è in grado di comunicare con Oracle Management Service (OMS) per fornire informazioni di monitoraggio. Se si esegue OMS 13.4, è possibile gestire i database installando OMA 13.4. Per ulteriori informazioni, consulta Oracle Management Agent per Enterprise Manager Cloud Control.	10 marzo 2021

Miglioramenti degli endpoint RDS Proxy	È possibile creare endpoint aggiuntivi associati a ciascun proxy RDS. La creazione di un endpoint in un VPC diverso consente l'accesso tra VPC per il proxy. I proxy per i cluster Aurora MySQL possono avere anche endpoint di sola lettura. Questi endpoint di lettura si connettono alle istanze database del lettore nei cluster e possono migliorare la scalabilità di lettura e la disponibilità per le applicazioni che richiedono un uso intensivo di query. Per maggiori informazioni su RDS Proxy, consulta la sezione "Gestione delle connessioni con Amazon RDS Proxy" nella Guida per l'utente di Amazon RDS o nella Guida per l'utente di Aurora.	8 marzo 2021
Amazon RDS estende il supporto ai backup automatizzati tra regioni	Ora puoi configurare istanze di database Amazon RDS che eseguono PostgreSQL per replicare snapshot DB e log delle transazioni in una regione diversa. AWS Per ulteriori informazioni, consulta Replicazione dei backup automatici in un'altra regione. AWS	8 marzo 2021

Filtri di replica per Amazon RDS for MariaDB e MySQL supportati nella regione Cina (Pechino) e Cina (Ningxia)	Il filtro di replica è ora supportato nelle regioni Cina (Pechino) e Cina (Ningxia) . Per ulteriori informazioni, consulta Configurazione dei filtri di replica con MariaDB e Configurazione dei filtri di replica con MySQL.	5 marzo 2021
Amazon RDS supporta la copia di snapshot DB tra regioni nelle regioni opt-in	Ora puoi copiare le istantane e del DB da e verso le regioni opzionali. AWS Per ulteriori informazioni, consulta Copiare istantanee tra regioni. AWS	4 marzo 2021
Amazon RDS for SQL Server supporta Gruppi di disponibilità Always On per Standard Edition	Quando si crea un'istanza database utilizzando la configurazione multi-AZ in SQL Server 2019 per il modulo di gestione di database Standard Edition, RDS utilizza automaticamente i gruppi di disponibilità. Per ulteriori informazioni, consulta Implementazioni Multi-AZ per Microsoft SQL Server.	23 febbraio 2021

Amazon RDS for Oracle introduce procedure relative agli advisor	Il pacchetto `rdsadmin_util` include le procedure `advisor_task_set_parameter` , `advisor_task_drop` e `dbms_stats_init` . Puoi utilizzare queste procedure per modificare, arrestare e riattivare attività di advisor come `AUTO_STATS_ADVISOR_TASK` . Per ulteriori informazioni, consulta Impostazione dei parametri per le attività advisor.	23 febbraio 2021
Amazon RDS fornisce i motivi di failover per istanze database Multi-AZ	Ora è possibile visualizzare spiegazioni più dettagliate quando un'istanza database Multi-AZ esegue il failover su una replica in standby. Per ulteriori informazioni, consulta Processo di failover per Amazon RDS.	18 febbraio 2021
Amazon RDS estende il supporto per l'esportazione di istantanee in Simple Storage Service (Amazon S3)	È ora possibile esportare i dati degli snapshot DB su Simple Storage Service (Amazon S3) in Cina. Per ulteriori informazioni, consulta Esportazione dei dati dello snapshot DB su Simple Storage Service (Amazon S3).	17 febbraio 2021

Filtri di replica per Amazon RDS for MariaDB e MySQL	È possibile configurare i filtri di replica per le istanze MySQL e MariaDB. I filtri di replica specificano quali database e tabelle vengono replicati in una replica di lettura. È possibile creare elenchi di database e tabelle da includere o escludere per ogni replica di lettura. Per ulteriori informazioni, consulta Configurazione dei filtri di replica con MariaDB e Configurazione dei filtri di replica con MySQL.	12 febbraio 2021
RDS per Oracle supporta APEX 20.2v1	È possibile utilizzare APEX 20.2.v1 con tutte le versioni supportate di Oracle Database. Per ulteriori informazioni, consulta Oracle Application Express.	2 febbraio 2021

Amazon RDS for SQL Server supporta lo storage delle istanze locali per il database tempdb	È ora possibile avviare Amazon RDS for SQL Server sui tipi di istanza Amazon EC2 db.r5d e db.m5d con il database tempdb configura to per l'utilizzo di un archivio istanze. Posizionando i file di dati tempdb e i file di registro localmente, è possibile ottenere latenze di lettura e scrittura inferiori rispetto allo storage standard basato su Amazon EBS. Per ulteriori informazioni, consulta Supporto dell'archivio istanze per il database tempdb in Amazon RDS for SQL Server.	27 gennaio 2021
Amazon RDS for PostgreSQ L supporta pg_partman e pg_cron	Amazon RDS for PostgreSQ L ora supporta le estension i pg_partman e pg_cron. Per ulteriori informazioni sull'este nsione pg_partman, consulta Gestione delle partizioni PostgreSQL con l'estensi one pg_partman. Per ulteriori informazioni sull'estensione pg_cron, consulta Pianifica zione della manutenzione con l'estensione PostgreSQL pg_cron.	12 gennaio 2021

Amazon RDS supporta la pubblicazione del registro di Oracle Management Agent su Amazon CloudWatch Logs	Il log di Oracle Management Agent è costituito da emctl.log , emdctlj.log, gcagent.log, gcagent_errors.log, emagent.nohup e secure.log. Amazon RDS pubblica ciascuno di questi log come flusso di log separato CloudWatch . Per ulteriori informazioni, consulta Pubblicazione dei log Oracle su Amazon CloudWatch Logs.	28 dicembre 2020
Amazon RDS on AWS Outposts supporta versioni di database aggiuntive	RDS su Outposts ora supporta versioni aggiuntive di MySQL e PostgreSQL. Per ulteriori informazioni, consulta il supporto di Amazon RDS on AWS Outposts per le funzionalità di Amazon RDS.	23 dicembre 2020
Amazon RDS su AWS Outposts supporta le COIP	RDS su Outposts ora supporta indirizzi IP di proprietà del cliente (CoIP). I CoIP forniscono connettività locale o esterna alle risorse nelle sottoreti di Outpost tramite la rete locale. Per ulteriori informazioni, consulta Indirizzi IP di proprietà del cliente per RDS su Outposts.	22 dicembre 2020

Amazon RDS for Oracle pianifica l'aggiornamento di istanze BYOL 11g a 19c	Il 4 gennaio 2021 è previsto l'aggiornamento automatico di tutte le edizioni delle istanze Oracle Database 11g sul modello Bring Your Own License (BYOL) a Oracle Database 19c. Tutte le istanze Oracle Database 11g, incluse le istanze riservate, verranno spostate all'ultimo aggiornamento disponibile di Release Update (RU). Per ulteriori informazioni, consulta Preparazione per l'aggiornamento automatico di Oracle Database 11g BYOL.	11 dicembre 2020
Amazon RDS supporta la replica di backup automatici in un'altra regione AWS	Ora puoi configurare le istanze del database Amazon RDS per replicare istantanee e log delle transazioni in una regione di destinazione AWS a tua scelta. Per ulteriori informazioni, consulta Replicazione dei backup automatici in un'altra regione. AWS	4 dicembre 2020

Amazon RDS for Oracle e Microsoft SQL Server supportano una nuova classe di istanza database	Ora puoi utilizzare la classe di istanza db.r5b per creare istanze database Amazon RDS che eseguono Oracle o SQL Server. Per ulteriori informazioni, consulta Motori di database supportati per tutte le classi di istanza database disponibili.	4 dicembre 2020
Supporto per MariaDB 10.2.32	Ora puoi creare istanze database Amazon RDS che eseguono MariaDB versione 10.2.32. Per ulteriori informazioni, consulta Versioni di MariaDB in Amazon RDS.	25 novembre 2020
Amazon RDS for SQL Server supporta Microsoft Business Intelligence Suite in SQL Server 2019	Ora puoi eseguire SQL Server Analysis Services, SQL Server Integration Services e SQL Server Reporting Services nelle istanze di DB utilizzando la versione principale più recente. Per ulteriori informazioni, consulta Opzioni per il modulo di gestione di database Microsoft SQL Server.	24 novembre 2020
Amazon RDS for PostgreSQL versione 13 nell'ambiente di anteprima del database	Amazon RDS for PostgreSQL ora supporta PostgreSQL versione 13 nell'ambiente di anteprima del database. Per ulteriori informazioni, consulta Versioni di PostgreSQL 13.	24 novembre 2020

Amazon RDS Approfondimenti sulle prestazioni introduce nuove dimensioni	È possibile raggruppare il carico del database in base ai gruppi di dimension i per database (PostgreS QL, MySQL e MariaDB), applicazione (PostgreSQL) e tipo di sessione (PostgreS QL). Amazon RDS supporta anche le dimensioni db.name (PostgreSQL, MySQL e MariaDB), db.application.nam e (PostgreSQL) e db.sessio n_type.name (PostgreSQL). Per ulteriori informazioni, consulta Tabelle dal carico superiore.	24 novembre 2020
Amazon RDS for MariaDB supporta una nuova versione principale	Ora puoi creare istanze database Amazon RDS che eseguono MariaDB versione 10.5. Per ulteriori informazioni, consulta Versioni di MariaDB in Amazon RDS.	23 novembre 2020
Supporto per MySQL 5.6.49	Ora puoi creare istanze database di Amazon RDS che eseguono MySQL versione 5.6.49. Per ulteriori informazi oni, consulta Versioni di MySQL in Amazon RDS.	20 novembre 2020
Supporto per MySQL 5.5.62	Ora puoi creare istanze database di Amazon RDS che eseguono MySQL versione 5.5.62. Per ulteriori informazi oni, consulta Versioni di MySQL in Amazon RDS.	20 novembre 2020

Approfondimenti sulle prestazioni supporta l'analisi delle statistiche di esecuzione di query PostgreSQL.	Ora puoi analizzare le statistiche delle query in esecuzione con Approfondimenti sulle prestazioni per le istanze database PostgreSQL. Per ulteriori informazioni, consulta Statistiche per PostgreSQL.	18 novembre 2020
Amazon RDS estende il supporto per la scalabilità automatica dello storage	Ora è possibile abilitare la scalabilità automatica dello storage durante la creazione di una replica di lettura, il ripristino di un'istanza database in un determinato momento o si ripristina un'istanza di MySQL DB da un backup Amazon S3. Per ulteriori informazioni, consulta Gestione automatica della capacità con scalabilità automatica Amazon RDS dello storage.	18 novembre 2020
Amazon RDS for SQL Server supporta Database Mail	Con Database Mail puoi inviare messaggi di posta elettronica dall'istanza database Amazon RDS for SQL Server. Dopo aver specificato i destinatari dell'e-mail, puoi aggiungere file o interrogare i risultati al messaggio inviato. Per ulteriori informazioni, consulta Utilizzo di Database Mail in Amazon RDS for SQL Server.	4 novembre 2020

Supporto per MySQL 8.0.21	Ora puoi creare istanze database di Amazon RDS che eseguono MySQL versione 8.0.21. Per ulteriori informazioni, consulta Versioni di MySQL in Amazon RDS.	22 ottobre 2020
Amazon RDS estende il supporto per l'esportazione di istantanee in Simple Storage Service (Amazon S3)	Ora puoi esportare i dati degli snapshot DB su Amazon S3 in tutte le AWS regioni commerciali. Per ulteriori informazioni, consulta Esportazione dei dati dello snapshot DB su Simple Storage Service (Amazon S3).	22 ottobre 2020
Amazon RDS for PostgreSQL supporta gli aggiornamenti di replica di lettura	Con Amazon RDS for PostgreSQL, quando esegui un aggiornamento della versione principale dell'istanza del database primario, anche le repliche di lettura vengono aggiornate automaticamente. Per ulteriori informazioni, consulta Aggiornamento del motore database PostgreSQL.	15 ottobre 2020
Amazon RDS for MariaDB, MySQL e PostgreSQL supportano le classi di istanza database Graviton	Ora è possibile utilizzare le classi di istanza database Graviton2 db.m6g.x e db.r6g.x per creare istanze database Amazon RDS con MariaDB, MySQL o PostgreSQL. Per ulteriori informazioni, consulta Motori di database supportati per tutte le classi di istanza database disponibili.	15 ottobre 2020

Amazon RDS for SQL Server supporta gli aggiornamenti a SQL Server 2019	È possibile aggiornare le istanze del DB di SQL Server a SQL Server 2019. Per informazioni, consulta Aggiornamento del motore del database Microsoft SQL Server.	6 ottobre 2020
Amazon RDS for Oracle supporta la specifica del set di caratteri nazionale	Il set di caratteri nazionale , denominato anche set di caratteri NCHAR, viene utilizzato nei tipi di dati `NCHAR`, `NVARCHAR2` e `NCLOB`. Quando si crea un database, è possibile specifica re AL16UTF16 (impostaz ione predefinita) o UTF8 come set di caratteri NCHAR. Per ulteriori informazioni, vedere Set di caratteri Oracle supportati in Amazon RDS.	2 ottobre 2020
Supporto per MySQL 5.7.31	Ora puoi creare istanze database di Amazon RDS che eseguono MySQL versione 5.7.31. Per ulteriori informazi oni, consulta Versioni di MySQL in Amazon RDS.	1 ottobre 2020

Amazon RDS for PostgreSQL supporta l'esportazione di dati in Simple Storage Service (Amazon S3)	Puoi eseguire query sui dati da un'istanza database PostgreSQL ed esportarli direttamente nei file archiviati in un bucket Simple Storage Service (Amazon S3). Per maggiori informazioni, consulta Esportazione di dati da un RDS per l'istanza database PostgreSQL a Simple Storage Service (Amazon S3).	24 settembre 2020
Amazon RDS per MySQL 8.0 supporta Percona XtraBackup	Ora puoi usare Percona XtraBackup per ripristinare un backup in un'istanza DB Amazon RDS for MySQL 8.0. Per ulteriori informazioni, consulta Ripristino di un backup in un'istanza database MySQL.	17 settembre 2020
Amazon RDS for SQL Server supporta il backup e il ripristino nativi su istanze DB con repliche di lettura	È possibile ripristinare un backup nativo di SQL Server in un'istanza database con repliche di lettura configurate. Per ulteriori informazioni, vedere Importazione ed esportazione di database SQL Server.	16 settembre 2020

Amazon RDS for SQL Server supporta fusi orari aggiuntivi	È possibile abbinare il fuso orario dell'istanza database con il fuso orario scelto. Per ulteriori informazioni, consulta Fuso orario locale per le istanze database Microsoft SQL Server.	11 settembre 2020
Amazon RDS for PostgreSQL versione 13 Beta 3 nell'ambiente di anteprima del database	Amazon RDS for PostgreSQL oggi supporta PostgreSQL versione 13 Beta 3 nell'ambiente di anteprima del database Per ulteriori informazioni, consulta Versioni di PostgreSQL 13.	9 settembre 2020
Amazon RDS for SQL Server supporta il flag di traccia 692	È ora possibile utilizzare il flag di traccia 692 come parametro di avvio utilizzando i gruppi di parametri DB. L'attivazione di questo flag di traccia disabilita gli inserti rapidi durante il caricamento di massa dei dati in heap o indici cluster. Per ulteriori informazioni, vedere Disattivazione degli inserti rapidi durante il caricamento di massa.	27 agosto 2020
Amazon RDS for SQL Server supporta Microsoft SQL Server 2019	È ora possibile creare istanze RDS DB che utilizzano SQL Server 2019. Per ulteriori informazioni, consulta Versioni di Microsoft SQL Server su Amazon RDS.	26 agosto 2020

RDS per Oracle supporta il database di replica montato	Quando si crea o si modifica una replica Oracle, è possibile posizionarla in modalità montata. Poiché il database di replica non accetta connessioni utente, non può gestire un carico di lavoro in sola lettura. La replica montata elimina i file di log redo archiviati dopo averli applicati. L'uso principale per le repliche montate è il ripristino di emergenza tra regioni. Per ulteriori informazioni, consulta Panoramica sulle repliche Oracle.	13 agosto 2020
RDS per Oracle prevede l'aggiornamento di istanze LI SE1 11g	Il 1° novembre 2020 prevediamo di iniziare l'aggiornamento automatico delle istanze Oracle Database 11g SE1 License Included (LI) a Oracle Database 19c Amazon RDS for Oracle. Tutte le istanze 11g, incluse le istanze riservate, verranno spostate all'ultimo aggiornamento disponibile di Oracle Release Update (RU). Per ulteriori informazioni, vedere Preparazione per l'aggiornamento automatico di Oracle Database 11g SE1.	31 luglio 2020

Amazon RDS supporta le nuove classi di istanza database Graviton2 nella versione di anteprima per PostgreSQL e MySQL	Ora è possibile creare istanze Amazon RDS DB che eseguono PostgreSQL o MySQL che utilizzano le classi di istanza database db.m6g.x e db.r6g.x. Per ulteriori informazioni, consulta Motori di database supportati per tutte le classi di istanza database disponibili.	30 luglio 2020
RDS per Oracle supporta APEX 20.1v1	È possibile utilizzare APEX 20.1v1 con tutte le versioni supportate di Oracle Database. Per ulteriori informazioni, consulta Oracle Application Express.	28 luglio 2020
Supporto per MySQL 8.0.20	Ora puoi creare istanze database di Amazon RDS che eseguono MySQL versione 8.0.20. Per ulteriori informazioni, consulta Versioni di MySQL in Amazon RDS.	23 luglio 2020
Amazon RDS for MariaDB e MySQL supportano nuove classi di istanza database	È ora possibile creare istanze database di Amazon RDS che eseguono MariaDB e MySQL che utilizzano le classi di istanza db.m5.16xlarge, db.m5.8xlarge, db.r5.16xlarge e db.r5.8xlarge. Per ulteriori informazioni, consulta Motori di database supportati per tutte le classi di istanza database disponibili.	23 luglio 2020

RDS per SQL Server supporta la disabilitazione delle versioni precedenti di TLS e cifrari	È possibile attivare e disattivare determinati protocolli di sicurezza e cifrari. Per ulteriori informazioni, vedere Configurazione dei protocolli di protezione e dei cifrari.	21 luglio 2020
RDS supporta Oracle Spatial su SE2	È possibile utilizzare Oracle Spatial in Standard Edition 2 (SE2) per tutte le versioni di 12.2, 18c e 19c. Per ulteriori informazioni, consulta Oracle Spatial.	9 luglio 2020
Amazon RDS supporta AWS PrivateLink	Amazon RDS ora supporta la creazione di endpoint Amazon VPC per le chiamate API Amazon RDS per mantenere il traffico tra le applicazioni e Amazon RDS nella rete. AWS Per ulteriori informazioni consulta Amazon RDS ed endpoint VPC di interfaccia (AWS PrivateLink).	9 luglio 2020
Amazon RDS per PostgreSQL versioni 9.4.x ha raggiunto la fine del supporto.	Amazon RDS for PostgreSQL non supporta più le versioni 9.4.x. Per le versioni supportate, consulta Versioni di database PostgreSQL supportate.	8 luglio 2020

Supporto per MariaDB 10.3.23 e 10.4.13	È ora possibile creare istanze database di Amazon RDS che eseguono MariaDB versione 10.3.23 e 10.4.13. Per ulteriori informazioni, consulta Versioni di MariaDB in Amazon RDS.	6 luglio 2020
Amazon RDS su AWS Outposts	È possibile creare istanze database Amazon RDS su AWS Outposts. Per ulteriori informazioni, consulta Utilizzo di Amazon RDS su AWS Outposts.	6 luglio 2020
Amazon RDS for Oracle crea automaticamente i file di inventario	Per aprire richieste di assistenza per i clienti BYOL, Oracle Support richiede i file di inventario generati da Opatch. Amazon RDS for Oracle crea automaticamente i file di inventario ogni ora nella directory BDUMP. Per ulteriori informazioni, consulta Accesso ai file Opatch.	6 luglio 2020
Supporto per MySQL 5.7.30 e 5.6.48	Ora puoi creare istanze database di Amazon RDS che eseguono MySQL versione 5.7.30 e 5.6.48. Per ulteriori informazioni, consulta Versioni di MySQL in Amazon RDS.	25 giugno 2020

Amazon RDS for Oracle supporta ADRCI	L'utility ADRCI (Automatic Diagnostic Repository Command Interpreter) è uno strumento a riga di comando Oracle utilizzato per gestire i dati di diagnostica. Utilizzando le funzioni del pacchetto Amazon RDS `rdsadmin_adrci_util` , puoi elencare e comprimere problemi e incidenti e mostrare anche i file di traccia. Per ulteriori informazioni, consulta Task comuni di diagnostica DBA per istanze di Oracle DB.	17 giugno 2020
Supporto per MySQL 8.0.19	Ora puoi creare istanze database di Amazon RDS che eseguono MySQL versione 8.0.19. Per ulteriori informazioni, consulta Versioni di MySQL in Amazon RDS.	2 giugno 2020
MySQL 8.0 supporta i nomi delle tabelle minuscole	È ora possibile impostare il parametro `lower_case_table_names` su `1` per le istanze Amazon RDS DB che eseguono MySQL versione 8.0.19 e versioni successive a 8.0. Per ulteriori informazioni, vedere Eccezioni dei parametri MySQL per le istanze DB Amazon RDS.	2 giugno 2020

Amazon RDS for Microsoft SQL Server supporta SQL Server Integration Services (SSIS)	SSIS è una piattaforma per l'integrazione dei dati e le applicazioni di workflow. È possibile abilitare SSIS su istanze database nuove o esistenti. È installato sulla stessa istanza database del motore del database. Per ulteriori informazioni, consulta la sezione relativa al supporto per SQL Server Analysis Services in SQL Server.	19 maggio 2020
Amazon RDS for Microsoft SQL Server supporta SQL Server Reporting Services (SSRS)	SSRS è un'applicazione basata su server utilizzata per la generazione e la distribuzione di report. È possibile abilitare SSRS su istanze database nuove o esistenti. È installato sulla stessa istanza database del motore del database. Per ulteriori informazioni, consulta la sezione relativa al supporto per SQL Server Analysis Services in SQL Server.	15 maggio 2020

Amazon RDS for Microsoft SQL Server supporta l'integrazione S3 su istanze Multi-AZ	È ora possibile utilizzare Simple Storage Service (Amazon S3) con le funzionalità di SQL Server, ad esempio l'inserimento di massa su istanze database Multi-AZ. Per maggiori informazioni, consulta Integrazione di un'istanza database Amazon RDS for SQL Server con Simple Storage Service (Amazon S3).	15 maggio 2020
Amazon RDS for Oracle supporta l'eliminazione del cestino riciclaggio	La procedura `rdsadmin.rdsadmin_util.purge_dba_recyclebin` elimina il cestino riciclaggio. Per ulteriori informazioni, consulta la sezione relativa all'eliminazione del cestino riciclaggio.	13 maggio 2020
Amazon RDS for Oracle migliora la gestibilità di AWR (Automatic Workload Repository)	Le procedure `rdsadmin.rdsadmin_diagnostic_util` generano report AWR ed estraggono i dati AWR in file di dump. Per ulteriori informazioni, consulta la sezione relativa alla generazione di report sulle prestazioni con AWR (Automatic Workload Repository).	13 maggio 2020

Amazon RDS for Microsoft SQL Server supporta Microsoft Distributed Transaction Coordinator (MSDTC)	Amazon RDS for SQL Server supporta transazioni distribuite tra host. Per ulteriori informazioni, consulta la sezione relativa al supporto per Microsoft Distributed Transaction Coordinator in SQL Server.	4 maggio 2020
Amazon RDS for Microsoft SQL Server supporta nuove versioni	È ora possibile creare istanze database Amazon RDS che eseguono SQL Server versioni 2017 CU19 14.00.3281.6, 2016 SP2 CU11 13.00.5598.27, 2014 SP3 CU4 12.00.6329.1 e 2012 SP4 GDR 11.0.7493.4 per tutte le edizioni. Per ulteriori informazioni, consulta Versioni di Microsoft SQL Server su Amazon RDS.	28 aprile 2020
Amazon RDS è disponibile in Regione Europa (Milano)	Amazon RDS è ora disponibile nella Regione Europa (Milano). Per ulteriori informazioni, consulta Regioni e zone di disponibilità.	28 aprile 2020
Supporto Amazon RDS per le Local Zones	È ora possibile avviare istanze database in una sottorete della zona locale. Per ulteriori informazioni, consulta Regioni, zone di disponibilità e zone locali	23 aprile 2020

Amazon RDS è disponibile in Regione Africa (Città del Capo)	Amazon RDS è ora disponibile nella Regione Africa (Città del Capo). Per ulteriori informazioni, consulta Regioni e zone di disponibilità.	22 aprile 2020
Amazon RDS for Microsoft SQL Server supporta SQL Server Analysis Services (SSAS)	SSAS è uno strumento di elaborazione analitica online (OLAP) e data mining installato all'interno di SQL Server. È possibile abilitare SSAS su istanze database esistenti o nuove. È installato sulla stessa istanza database del motore del database. Per ulteriori informazioni, consulta Supporto per SQL Server Analysis Services in SQL Server.	17 aprile 2020
Amazon RDS Proxy per PostgreSQL	Amazon RDS Proxy è ora disponibile per PostgreSQL. È possibile utilizzare RDS Proxy per ridurre il sovraccarico di gestione delle connessioni sull'istanza database e anche la possibilità di errori di "troppe connessioni". RDS Proxy è attualmente in anteprima pubblica per PostgreSQL. Per ulteriori informazioni, consulta Gestione delle connessioni con il proxy Amazon RDS (Anteprima).	8 aprile 2020

Amazon RDS for Oracle supporta Oracle APEX versione 19.2.v1	Amazon RDS for Oracle supporta ora Oracle Application Express (APEX) versione 19.2.v1. Per ulteriori informazioni, consulta Oracle Application Express.	8 aprile 2020
Amazon RDS for MariaDB supporta una nuova versione principale	Puoi ora creare istanze database Amazon RDS che eseguono MariaDB versione 10.4. Per ulteriori informazioni, consulta Versioni di MariaDB in Amazon RDS.	6 aprile 2020
Amazon RDS Approfondimenti sulle prestazioni è disponibile per Amazon RDS for MariaDB 10.4	Amazon RDS Approfondimenti sulle prestazioni è ora disponibile su Amazon RDS for MariaDB versione 10.4. Per ulteriori informazioni, consulta Utilizzo di Amazon RDS Approfondimenti sulle prestazioni.	6 aprile 2020
Amazon RDS per PostgreSQL versioni 9.3.x ha raggiunto la fine del supporto	Amazon RDS for PostgreSQL non supporta più le versioni 9.3.x. Per le versioni supportate, consulta Versioni di database PostgreSQL supportate.	3 aprile 2020
Amazon RDS for Microsoft SQL Server supporta le repliche di lettura	È ora possibile creare repliche di lettura per istanze database SQL Server. Per ulteriori informazioni, consulta Uso di repliche di lettura.	3 aprile 2020

Amazon RDS for Microsoft SQL Server supporta backup multifile	È ora possibile eseguire il backup dei database in più file utilizzando il backup e il ripristino nativo di SQL Server. Per ulteriori informazioni, consulta Backup di un database.	2 aprile 2020
Integrazione di Amazon RDS per Oracle con AWS License Manager	Amazon RDS for Oracle è ora integrato con AWS License Manager. Se utilizzi il modello Bring Your Own License, AWS License Manager l'integrazione semplifica il monitoraggio dell'utilizzo della licenza Oracle all'interno dell'organizzazione. Per ulteriori informazioni, consulta Integrazione con AWS License Manager.	23 marzo 2020
Supporto per 64 TiB su istanze db.r5 in Amazon RDS for MariaDB e MySQL	È ora possibile creare istanze database Amazon RDS for MariaDB e MySQL che utilizzano la classe di istanza database db.r5 con un massimo di 64 TiB di storage. Per ulteriori informazioni, consulta Fattori che influenzano le prestazioni di storage.	18 marzo 2020
Supporto per MySQL 8.0.17	Ora puoi creare istanze database di Amazon RDS che eseguono MySQL versione 8.017. Per ulteriori informazioni, consulta Versioni di MySQL in Amazon RDS.	10 marzo 2020

Amazon RDS Approfondimenti sulle prestazioni è disponibile per Amazon RDS for MySQL 8.0	Amazon RDS Approfondimenti sulle prestazioni è ora disponibile per Amazon RDS for MySQL versione 8.0.17 e versioni 8.0 successive. Per ulteriori informazioni, consulta Utilizzo di Amazon RDS Approfondimenti sulle prestazioni.	10 marzo 2020
Supporto per MySQL 5.6.46	Ora puoi creare istanze database di Amazon RDS che eseguono MySQL versione 5.6.46. Per ulteriori informazioni, consulta Versioni di MySQL in Amazon RDS.	28 febbraio 2020
Amazon RDS Approfondimenti sulle prestazioni è disponibile per Amazon RDS for MariaDB 10.3	Amazon RDS Approfondimenti sulle prestazioni è ora disponibile per Amazon RDS for MariaDB versione 10.3.13 e versioni successive 10.3. Per ulteriori informazioni, consulta Utilizzo di Amazon RDS Approfondimenti sulle prestazioni.	26 febbraio 2020
Supporto per MySQL 5.7.28	Ora puoi creare istanze database di Amazon RDS che eseguono MySQL versione 5.7.28. Per ulteriori informazioni, consulta Versioni di MySQL in Amazon RDS.	20 febbraio 2020

Supporto per MariaDB 10.3.20	È ora possibile creare istanze database di Amazon RDS che eseguono MariaDB versione 10.3.20. Per ulteriori informazioni, consulta Versioni di MariaDB in Amazon RDS.	20 febbraio 2020
Amazon RDS for Microsoft SQL Server supporta nuove classi di istanze database	Ora puoi creare istanze database di Amazon RDS che eseguono Oracle con le classi di istanze database db.z1d. Per ulteriori informazioni, consulta Supporto di classe istanza database per Microsoft SQL Server.	19 febbraio 2020
Supporto per domini Active Directory tra account, tra VPC in Amazon RDS for SQL Server	Amazon RDS for Microsoft SQL Server supporta ora l'associazione di istanze database a domini Active Directory di proprietà di account e VPC diversi. Per ulteriori informazioni, consulta Utilizzo dell'autenticazione di Windows con un'istanza database di Microsoft SQL Server .	13 febbraio 2020

Opzione Oracle OLAP	Amazon RDS for Oracle ora supporta l'opzione OLAP (On-line Analytical Processing) per le istanze database di Oracle. È possibile utilizzare Oracle OLAP per analizzare grandi quantità di dati creando oggetti e cubi dimensionali in conformità con lo standard OLAP. Per ulteriori informazioni, consulta Oracle OLAP.	13 febbraio 2020
Supporto FIPS 140-2 per Oracle	Amazon RDS for Oracle supporta la Federal Information Processing Standard Publication 140-2 (FIPS 140-2) per connessioni SSL/TLS. Per ulteriori informazioni, consulta Supporto FIPS.	11 febbraio 2020
Amazon RDS for PostgreSQL supporta le nuove classi di istanza database	È ora possibile creare istanze database di Amazon RDS che eseguono PostgreSQL che utilizzano le classi di istanza db.m5.16xlarge, db.m5.8xlarge, db.r5.16xlarge e db.r5.8xlarge. Per ulteriori informazioni, consulta Motori di database supportati per tutte le classi di istanza database disponibili.	11 febbraio 2020

Approfondimenti sulle prestazioni supporta l'analisi delle statistiche di esecuzione di query MariaDB e MySQL	È ora possibile analizzare le statistiche delle query in esecuzione con Approfondimenti sulle prestazioni per le istanze database MariaDB e MySQL. Per ulteriori informazioni, consulta Analisi delle statistiche delle query in esecuzione.	4 febbraio 2020
Supporto per l'esportazione di dati dello snapshot DB Simple Storage Service (Amazon S3) per MariaDB, MySQL e PostgreSQL	Amazon RDS supporta l'esportazione di dati dello snapshot DB Simple Storage Service (Amazon S3) per MariaDB, MySQL e PostgreSQL. Per ulteriori informazioni, consulta Esportazione dei dati dello snapshot DB su Simple Storage Service (Amazon S3).	23 gennaio 2020
Amazon RDS for MySQL supporta l'autenticazione Kerberos	Ora, puoi utilizzare l'autenticazione Kerberos per autenticare gli utenti quando si connettono alle istanze database Amazon RDS for MySQL. Per ulteriori informazioni, consulta Utilizzo dell'autenticazione Kerberos per MySQL.	21 gennaio 2020

Amazon RDS Approfondimenti sulle prestazioni supporta la visualizzazione di più testo SQL per Amazon RDS for Microsoft SQL Server	Amazon RDS Approfondimenti sulle prestazioni ora supporta la visualizzazione di una maggiore quantità di testo SQL nel pannello di Approfondimenti sulle prestazioni per le istanze database Amazon RDS for Microsoft SQL Server. Per ulteriori informazioni, consultare Visualizzazione di una maggiore quantità di testo SQL nel pannello di controllo di Approfondimenti sulle prestazioni.	17 dicembre 2019
Amazon RDS Proxy	Puoi ridurre il sovraccarico della gestione delle connessioni nel cluster e ridurre la possibilità di errori del tipo «troppe connessioni» utilizzando il proxy Amazon RDS. Puoi associare ogni proxy a un'istanza RDS di DB Servizi di dominio Active Directory o cluster di DB Aurora. Quindi utilizzi l'endpoint proxy nella stringa di connessione per l'applicazione. Il proxy Amazon RDS è attualmente in uno stato di anteprima pubblica. Supporta il motore database RDS for MySQL. Per ulteriori informazioni, consulta la sezione relativa alla gestione delle connessioni con il proxy Amazon RDS (Anteprima).	3 dicembre 2019

Amazon RDS su AWS Outposts (anteprima)	Con Amazon RDS attivo AWS Outposts, puoi creare database AWS relazionali gestiti nei tuoi data center locali. RDS su Outposts consente di eseguire database RDS su AWS Outposts. Per ulteriori informazioni, consulta Amazon RDS su AWS Outposts (anteprima).	3 dicembre 2019
Amazon RDS for Oracle supporta repliche di lettura tra regioni diverse	Amazon RDS for Oracle ora supporta repliche di lettura tra regioni diverse con Active Data Guard. Per ulteriori informazioni, consulta Gestione delle repliche di lettura e Gestione delle repliche di lettura Oracle.	26 novembre 2019
Approfondimenti sulle prestazioni supporta l'analisi delle statistiche di esecuzione di query Oracle	È ora possibile analizzare le statistiche delle query in esecuzione con Approfondimenti sulle prestazioni per le istanze Oracle DB. Per ulteriori informazioni, consulta Analisi delle statistiche delle query in esecuzione.	25 novembre 2019

Amazon RDS per Microsoft SQL Server supporta la pubblicazione di log su CloudWatch Logs	Puoi configurare la tua istanza DB Amazon RDS for SQL Server per pubblicare gli eventi di log direttamente su CloudWatch Amazon Logs. Per ulteriori informazioni, consulta Pubblicazione dei log di SQL Server su Amazon CloudWatch Logs.	25 novembre 2019
Amazon RDS for Microsoft SQL Server supporta nuove classi di istanze database	Ora puoi creare istanze database Amazon RDS che eseguono SQL Server che utilizzano classi di istanze database db.x1e e db.x1. Per ulteriori informazioni, consulta Supporto di classe istanza database per Microsoft SQL Server.	25 novembre 2019
Amazon RDS for Microsoft SQL Server supporta ripristini differenziali e di log	È possibile ripristinare backup e log differenziali tramite backup e ripristino nativo di SQL Server. Per maggiori informazioni, consulta Utilizzo di ripristino e backup nativo.	25 novembre 2019
Multi-AZ supportato su Amazon RDS for Microsoft SQL Server in nuove regioni	Multi-AZ su SQL Server è ora disponibile in Cina, Medio Oriente (Bahrein), e Europa (Stoccolma). Per ulteriori informazioni, consulta Implementazioni Multi-AZ per Microsoft SQL Server.	22 novembre 2019

Amazon RDS for Microsoft SQL Server ora supporta bulk insert e S3 integration	Puoi trasferire i file tra un'istanza database SQL Server e un bucket Simple Storage Service (Amazon S3). Puoi utilizzare Simple Storage Service (Amazon S3) con le caratteristiche di SQL Server, come bulk insert. Per maggiori informazioni, consulta Integrazione di un'istanza database Amazon RDS for SQL Server con Simple Storage Service (Amazon S3).	21 novembre 2019
Contatori Approfondimenti sulle prestazioni per Amazon RDS for Microsoft SQL Server	Ora puoi aggiungere i contatori delle performance ai grafici Approfondimenti sulle prestazioni per le istanze database Microsoft SQL Server. Per ulteriori informazioni, consulta Contatori Approfondimenti sulle prestazioni per Amazon RDS for Microsoft SQL Server.	12 novembre 2019

Amazon RDS for Microsoft SQL Server supporta nuove dimensioni di classi di istanze database	Ora puoi creare istanze database Amazon RDS che eseguono SQL Server che utilizzano dimensioni di istanze 8xlarge e 16xlarge per classi di istanze database db.m5 e db.r5. Dimensioni di istanze da piccole a 2xlarge sono ora disponibili nella classe di istanze db.t3. Per ulteriori informazioni, consulta Supporto di classe istanza database per Microsoft SQL Server.	11 novembre 2019
Supporto per aggiornamenti snapshot PostgreSQL	Se hai snapshot DB manuali esistenti delle tue istanze database Amazon RDS PostgreSQL, puoi aggiornarli alla versione successiva del motore del database PostgreSQL. Per maggiori informazioni, consulta Aggiornamento di uno snapshot DB PostgreSQL.	7 novembre 2019
Amazon RDS for Oracle supporta una nuova versione principale	Ora puoi creare istanze database di Amazon RDS che eseguono Oracle Database 19c (19.0). Per maggiori informazioni, consulta Oracle Database 19c con Amazon RDS.	7 novembre 2019

Amazon RDS for PostgreSQL versione 12.0 nell'ambiente di anteprima del database	Amazon RDS for PostgreSQL ora supporta PostgreSQL versione 12.0 nell'ambiente di anteprima del database. Per maggiori informazioni, consulta PostgreSQL versione 12.0 nell'ambiente di anteprima del database.	1 novembre 2019
Amazon RDS for PostgreSQL supporta l'autenticazione Kerberos	Ora puoi utilizzare l'autenticazione Kerberos per autenticare gli utenti quando si connettono all'istanza database Amazon RDS che esegue PostgreSQL. Per ulteriori informazioni, consulta Utilizzo di Autenticazione Kerberos con Amazon RDS for PostgreSQL.	28 ottobre 2019
Attività del database OEM Management Agent per le istanze database Oracle	Le istanze database Amazon RDS for Oracle ora supportano procedure per richiamare alcuni comandi EMCTL in Management Agent. Per ulteriori informazioni, consulta Attività del database OEM Agent.	24 ottobre 2019

Amazon RDS for PostgreSQL supporta Transportable Database di PostgreSQL	Transportable Database di PostgreSQL fornisce un metodo estremamente veloce per eseguire la migrazione di un database RDS di PostgreSQL tra due istanze database. Per ulteriori informazioni, consulta Trasporto di database PostgreSQL tra istanze database.	8 ottobre 2019
Amazon RDS for Oracle supporta l'autenticazione Kerberos	Ora, puoi utilizzare l'autenticazione Kerberos per autenticare gli utenti quando si connettono all'istanza database Amazon RDS che esegue Oracle. Per ulteriori informazioni, consulta la sezione relativa all'utilizzo dell'autenticazione Kerberos con Amazon RDS for Oracle.	30 settembre 2019
Amazon RDS for PostgreSQL versione 12 Beta 3 nell'ambiente di anteprima del database	Amazon RDS for PostgreSQL oggi supporta PostgreSQL Version 12 Beta 3 nell'ambiente di anteprima del database Per ulteriori informazioni, consulta PostgreSQL versione 12 Beta 3 su Amazon RDS nell'ambiente di anteprima del database.	28 agosto 2019

Supporto per MySQL 8.0.16	Ora puoi creare istanze database di Amazon RDS che eseguono MySQL versione 8.0.16. Per ulteriori informazioni, consulta Versioni di MySQL in Amazon RDS.	19 agosto 2019
Amazon RDS for Oracle supporta una nuova versione principale	Ora puoi creare istanze database di Amazon RDS che eseguono Oracle Database 18c (18.0). Per ulteriori informazioni, consulta Oracle Database 18c con Amazon RDS.	15 agosto 2019
Management Agent per OEM 13c Release 3	Le istanze database Amazon RDS for Oracle ora supportano il Management Agent per Oracle Enterprise Manager (OEM) Cloud Control 13c Release 3. Per ulteriori informazioni, consulta Oracle Management Agent per Enterprise Manager Cloud Control.	7 agosto 2019
Amazon RDS for PostgreSQL versione 12 Beta 2 nell'ambiente di anteprima del database	Amazon RDS for PostgreSQL oggi supporta PostgreSQL Version 12 Beta 2 nell'ambiente di anteprima del database Per ulteriori informazioni, consulta PostgreSQL versione 12 Beta 2 su Amazon RDS nell'ambiente di anteprima del database.	6 agosto 2019

Amazon RDS supporta collazioni server per SQL Server	Amazon RDS for SQL Server supporta una selezione di collazioni per nuove istanze database. Per ulteriori informazioni, consulta Collazioni e set di caratteri per Microsoft SQL Server.	29 luglio 2019
Supporto di Oracle APEX versione 19.1.v1 in Amazon RDS for Oracle	Amazon RDS for Oracle supporta ora Oracle Applicati on Express (APEX) versione 19.1.v1. Per ulteriori informazi oni, consulta Oracle Applicati on Express.	28 giugno 2019
Amazon RDS for PostgreSQ L versione 13 Beta 1 nell'ambi ente di anteprima del database	Amazon RDS for PostgreSQ L oggi supporta PostgreSQL Version 13 Beta 1 nell'ambi ente di anteprima del database Per ulteriori informazi oni, consulta Versioni di PostgreSQL 13.	22 giugno 2019
Scalabilità automatica dell'arch iviazione di Amazon RDS	La scalabilità automatica dello storage per le istanze DB di Amazon RDS consente ad Amazon RDS di espandere automaticamente lo storage associato a un'istanza DB per ridurre la possibilità che si verifichino condizioni. out-of-sp ace Per informazioni sull'Auto Scaling dello storage, consulta Uso dello storage per istanze database di Amazon RDS.	20 giugno 2019

Amazon RDS for Oracle supporta le classi di istanze database db.z1d	Ora puoi creare istanze database di Amazon RDS che eseguono Oracle con le classi di istanze database db.z1d. Per ulteriori informazioni, consulta la pagina relativa alla classe di istanza database.	13 giugno 2019
Amazon RDS Approfondimenti sulle prestazioni supporta la visualizzazione di una maggiore quantità di testo SQL per Amazon RDS for Oracle	Amazon RDS Approfondimenti sulle prestazioni ora supporta la visualizzazione di una maggiore quantità di testo SQL nel pannello di controllo Approfondimenti sulle prestazioni per le istanze database Amazon RDS for Oracle. Per ulteriori informazioni, consultare Visualizzazione di una maggiore quantità di testo SQL nel pannello di controllo di Approfondimenti sulle prestazioni.	10 giugno 2019
Amazon RDS aggiunge il supporto per ripristini nativi di database SQL Server fino a 16 TB.	Ora puoi eseguire ripristini nativi fino a 16 TB da SQL Server in Amazon RDS. Per ulteriori informazioni, consulta Amazon RDS for SQL Server: Limitazioni e consigli.	4 giugno 2019

Amazon RDS aggiunge supporto Microsoft SQL Server Audit	Utilizzando Amazon RDS for Microsoft SQL Server, puoi eseguire l'audit del server e degli eventi del livello database mediante SQL Server Audit e visualizzare i risultati sull'istanza database o inviare i file di log dell'audit direttamente a Simple Storage Service (Amazon S3). Per ulteriori informazioni, consulta SQL Server Audit.	23 maggio 2019
Miglioramenti alle raccomandazioni Amazon RDS	Amazon RDS ha migliorato i consigli automatici per le risorse di database. Ad esempio, Amazon RDS ora offre raccomandazioni per i parametri di database. Per ulteriori informazioni, consulta la pagina relativa all'Utilizzo dei consigli per Amazon RDS.	22 maggio 2019
Supporto per altri database per l'istanza database per Amazon RDS for SQL Server	È possibile creare fino a 30 database su ciascuna delle istanze database che eseguono Microsoft SQL Server. Per ulteriori informazioni, consulta Limiti delle istanze database Microsoft SQL Server.	21 maggio 2019

Supporto per 64 TiB e 80k IOPS di storage per Amazon RDS for MariaDB, MySQL e PostgreSQL	Ora, puoi creare istanze database Amazon RDS for MariaDB, MySQL e PostgreSQL con fino a 64 TiB di storage e fino a 80.000 IOPS sottoposti a provision ing. Per ulteriori informazioni, consulta la sezione relativa allo storage per le istanze database.	20 maggio 2019
Amazon RDS for MySQL supporta i precheck di aggiornamento	Quando si aggiorna un'istanz a database da MySQL 5.7 a MySQL 8.0, Amazon RDS esegue i precheck per eventuali incompatibilità. Per ulteriori informazioni, consulta Precheck per gli aggiornam enti da MySQL 5.7 a 8.0.	17 maggio 2019
Supporto per il plugin di convalida della password MySQL	Ora puoi utilizzare il plugin `validate_password` MySQL per la sicurezza migliorata delle istanze database Amazon RDS for MySQL. Per ulteriori informazi oni, consulta Utilizzo del plugin di convalida della password.	16 maggio 2019

Contatori Approfondimenti sulle prestazioni per Amazon RDS for Oracle	Ora puoi aggiungere i contatori nei grafici di Approfondimenti sulle prestazioni per le istanza database Oracle. Per ulteriori informazioni, consulta Contatori Approfondimenti sulle prestazioni per Amazon RDS for Oracle.	8 maggio 2019
Supporto per la fatturazione al secondo	Amazon RDS viene ora fatturato in incrementi di 1 secondo in tutte le AWS regioni tranne AWS GovCloud (Stati Uniti) per le istanze on-demand. Per ulteriori informazioni, consulta Fatturazione delle istanze del database per Amazon RDS.	25 aprile 2019
Supporto per l'importazione dei dati da Simple Storage Service (Amazon S3) per Amazon RDS for PostgreSQL	Puoi ora importare i dati dal file Simple Storage Service (Amazon S3) in una tabella in un'istanza database PostgreSQL RDS. Per ulteriori informazioni, consulta Importazione dei dati Amazon S3 in un'istanza database PostgreSQL RDS.	24 aprile 2019

Supporto per il ripristino dei backup 5.7 da Simple Storage Service (Amazon S3)	Puoi ora creare un backup del database MySQL versione 5.7, archiviarlo in Simple Storage Service (Amazon S3) e quindi ripristinare il file di backup in una nuova istanza database Amazon RDS che esegue MySQL. Per ulteriori informazioni, consulta Ripristino di un backup in un'istanza database MySQL.	17 aprile 2019
Supporto per aggiornamenti multipli principali della versione per Amazon RDS for PostgreSQL	Con Amazon RDS for PostgreSQL, puoi ora scegliere tra più versioni principali quando effettui un aggiornamento al motore database. Questa caratteristica ti consente di passare a una versione principale più recente quando aggiorni le versioni del motore PostgreSQL selezionate. Per ulteriori informazioni, consulta Aggiornamento del motore database PostgreSQL.	16 aprile 2019
Supporto per 64 TiB di archiviazione per Amazon RDS for Oracle	Ora puoi creare istanze database Amazon RDS for Oracle con fino a 64 TiB di storage e fino a 80.000 IOPS sottoposti a provisioning. Per ulteriori informazioni, consulta la sezione relativa allo storage per le istanze database.	4 Aprile 2019

Supporto per MySQL 8.0.15	Ora puoi creare istanze database di Amazon RDS che eseguono MySQL versione 8.0.15. Per ulteriori informazioni, consulta Versioni di MySQL in Amazon RDS.	3 aprile 2019
Supporto per MariaDB 10.3.13	Puoi ora creare istanze database Amazon RDS che eseguono MariaDB versione 10.3.13. Per ulteriori informazioni, consulta Versioni di MariaDB in Amazon RDS.	3 aprile 2019
Microsoft SQL Server 2008 R2 ha raggiunto la fine del supporto su Amazon RDS	Microsoft SQL Server 2008 R2 ha raggiunto la fine del supporto e ciò coincide con il piano Microsoft per terminare il supporto esteso per questa versione il 9 luglio 2019. Qualsiasi snapshot esistente Microsoft SQL Server 2008 R2 deve essere aggiornato automaticamente all'ultima versione minore di Microsoft SQL Server 2012 a partire dal 1° giugno 2019. Per ulteriori informazioni, consulta Supporto Microsoft SQL Server 2008 R2 su Amazon RDS .	2 Aprile 2019

Gruppi di disponibilità sempre attivi supportati in Microsoft SQL Server 2017	Ora puoi utilizzare i gruppi di disponibilità sempre attivi in SQL Server 2017 Enterprise Edition 14.00.3049.1 o versione successiva. Per ulteriori informazioni, consulta Implementazioni Multi-AZ per Microsoft SQL Server.	29 marzo 2019
Vista parametri volume	Puoi ora visualizzare i parametri per i volumi Amazon Elastic Block Store (Amazon EBS), che sono i dispositivi fisici utilizzati per i database e i log di storage. Per ulteriori informazioni, consulta Visualizzazione del monitoraggio avanzato.	20 marzo 2019
Supporto per MySQL 5.7.25.	Ora puoi creare istanze database di Amazon RDS che eseguono MySQL versione 5.7.25. Per ulteriori informazioni, consulta Versioni di MySQL in Amazon RDS.	19 marzo 2019
Amazon RDS for Oracle supporta operazioni DBA RMAN	Amazon RDS for Oracle ora supporta le operazioni DBA di Oracle RMAN (Recovery Manager), inclusi i backup RMAN. Per ulteriori informazioni, consulta Operazioni DBA Recovery Manager (RMAN) per istanza database Oracle.	14 marzo 2019

Amazon RDS for PostgreSQL supporta la versione 11.1	Ora puoi creare istanze database di Amazon RDS che eseguono PostgreSQL versione 11.1. Per ulteriori informazioni, consulta PostgreSQL versione 11.1 su Amazon RDS.	12 marzo 2019
Ripristino multi file disponibile in Amazon RDS for SQL Server	Ora puoi ripristinare da più file con Amazon RDS for SQL Server. Per ulteriori informazioni, consulta Ripristino di un database.	11 marzo 2019
MariaDB 10.2.21	Puoi ora creare istanze database Amazon RDS che eseguono MariaDB versione 10.2.21. Per ulteriori informazioni, consulta Versioni di MariaDB in Amazon RDS.	11 marzo 2019
Amazon RDS for Oracle supporta repliche di lettura	Amazon RDS for Oracle ora supporta repliche di lettura con Active Data Guard. Per ulteriori informazioni, consulta Gestione delle repliche di lettura e Gestione delle repliche di lettura Oracle.	11 marzo 2019
Amazon RDS Approfondimenti sulle prestazioni è disponibile per Amazon RDS for MariaDB	Amazon RDS Approfondimenti sulle prestazioni è ora disponibile per Amazon RDS for MariaDB. Per ulteriori informazioni, consulta la pagina Utilizzo di Amazon RDS Approfondimenti sulle prestazioni.	11 marzo 2019

MySQL 8.0.13 e 5.7.24	Puoi ora creare istanze database Amazon RDS che eseguono MySQL versioni 8.0.13 e 5.7.24. Per ulteriori informazioni, consulta Versioni di MySQL in Amazon RDS.	8 marzo 2019
Amazon RDS Approfondimenti sulle prestazioni disponibile per Amazon RDS for SQL Server	Amazon RDS Approfondimenti sulle prestazioni è ora disponibile per Amazon RDS for SQL Server. Per ulteriori informazioni, consulta la pagina Utilizzo di Amazon RDS Approfondimenti sulle prestazioni.	4 marzo 2019
Amazon RDS for Oracle supporta l'integrazione Amazon S3	Ora è possibile trasferire i file tra un'istanza database Amazon RDS for Oracle e un bucket Simple Storage Service (Amazon S3). Per ulteriori informazioni, consulta Integrazione di Amazon RDS for Oracle e Simple Storage Service (Amazon S3).	26 febbraio 2019
Amazon RDS for MySQL e Amazon RDS for MariaDB supportano classi di istanza database db.t	Puoi ora creare istanze database Amazon RDS che eseguono MySQL o MariaDB con classi di istanza database db.t3. Per ulteriori informazioni, consulta la pagina relativa alla classe di istanza database.	20 febbraio 2019

Amazon RDS for MySQL e Amazon RDS for MariaDB supportano classi di istanza database db.r	Puoi ora creare istanze database Amazon RDS che eseguono MySQL o MariaDB con classi di istanza database db.r5. Per ulteriori informazioni, consulta la pagina relativa alla classe di istanza database.	20 febbraio 2019
Contatori di Approfondimenti sulle prestazioni per RDS for MySQL e PostgresSQL	Ora puoi aggiungere i contatori delle prestazioni ai grafici di Approfondimenti sulle prestazioni per le istanze database MySQL e PostgreSQL. Per ulteriori informazioni, consulta la pagina relativa ai Componenti del pannello di controllo Approfondimenti sulle prestazioni.	19 febbraio 2019
Amazon RDS for PostgreSQL ora supporta l'ottimizzazione del parametro autovacuum adattiva	L'ottimizzazione del parametro autovacuum adattiva con Amazon RDS for PostgreSQL aiuta a prevenire il wraparound dell'ID della transazione regolando automaticamente i valori del parametro autovacuum. Per ulteriori informazioni, consulta Riduzione della probabilità di eseguire il wraparound dell'ID della transazione.	12 febbraio 2019

Amazon RDS for Oracle supporta le versioni di Oracle APEX 18.1.v1 e 18.2.v1	Amazon RDS for Oracle supporta ora Oracle Applicati on Express (APEX) versione 18.1.v1. e 18.2.v1. Per ulteriori informazioni, consulta Oracle Application Express.	11 febbraio 2019
Amazon RDS Approfondimenti sulle prestazioni supporta la visualizzazione di una maggiore quantità di testo SQL per RDS for MySQL.	Amazon RDS Approfond imenti sulle prestazioni ora supporta la visualizzazione di una maggiore quantità di testo SQL nel pannello di controllo di Approfond imenti sulle prestazioni per le istanze database MySQL. Per ulteriori informazioni, consultare Visualizzazione di una maggiore quantità di testo SQL nel pannello di controllo di Approfondimenti sulle prestazioni.	6 febbraio 2019
Amazon RDS for PostgreSQ L supporta le classi di istanza database db.t	Ora puoi creare istanze database Amazon RDS che eseguono PostgreSQ L con le classi di istanza database db.t3. Per ulteriori informazioni, consulta la pagina relativa alla classe di istanza database.	25 gennaio 2019

Amazon RDS for Oracle supporta le classi di istanze database db.t	Ora puoi creare istanze database di Amazon RDS che eseguono Oracle con le classi di istanze database db.t3. Per ulteriori informazioni, consulta la pagina relativa alla classe di istanza database.	25 gennaio 2019
Amazon RDS Approfondimenti sulle prestazioni supporta la visualizzazione di una maggiore quantità di testo per PostgreSQL Amazon RDS.	Amazon RDS Approfondimenti sulle prestazioni ora supporta la visualizzazione di una maggiore quantità di testo SQL nel pannello di controllo di Approfondimenti sulle prestazioni delle istanze database Amazon RDS PostgreSQL. Per ulteriori informazioni, consulta Visualizzazione di una maggiore quantità di testo SQL nel pannello di controllo di Approfondimenti sulle prestazioni.	24 gennaio 2019
Amazon RDS for Oracle supporta una nuova versione di SQLT	Amazon RDS for Oracle ora supporta SQLT versione 12.2.180725. Per ulteriori informazioni, consulta Oracle SQLT.	22 gennaio 2019

Amazon RDS for PostgreSQL supporta le classi di istanza database db.r	Ora puoi creare istanze database Amazon RDS che eseguono PostgreSQL con le classi di istanza database db.r5. Per ulteriori informazioni, consulta la pagina relativa alla classe di istanza database.	19 dicembre 2018
Amazon RDS for PostgreSQL ora supporta la gestione delle password con restrizioni	Amazon RDS for PostgreSQL ti consente di impostare restrizioni riguardo a chi può gestire le modifiche alle password degli utenti e alla scadenza delle password mediante il parametro `rds.restrict_password_commands` e il ruolo `rds_password`. Per ulteriori informazioni, consulta Restricting Password Management	19 dicembre 2018
Amazon RDS per PostgreSQL supporta il caricamento dei log del database su Amazon Logs CloudWatch	Amazon RDS per PostgreSQL supporta il caricamento dei log del database su Logs. CloudWatch Per ulteriori informazioni, consulta Pubblicazione dei log di PostgreSQL nei log. CloudWatch	10 dicembre 2018

Amazon RDS for Oracle supporta le classi di istanza database db.r	Ora puoi creare istanze database Amazon RDS che eseguono Oracle con le classi di istanza database db.r5. Per ulteriori informazioni, consulta la pagina relativa alla classe di istanza database.	20 Novembre 2018
Conservare i backup quando si elimina un'istanza database	Amazon RDS supporta la conservazione dei backup automatici quando elimini un'istanza database. Per ulteriori informazioni, consulta Utilizzo dei backup.	15 Novembre 2018
Amazon RDS for PostgreSQL supporta le classi di istanza database db.m	Ora puoi creare istanze database Amazon RDS che eseguono PostgreSQL con le classi di istanza database db.m5. Per ulteriori informazioni, consulta la pagina relativa alla classe di istanza database.	15 Novembre 2018
Amazon RDS for Oracle supporta una nuova versione principale	Ora puoi creare istanze database Amazon RDS che eseguono Oracle versione 12.2. Per ulteriori informazioni, consulta Oracle Database 12c Release 2 (12.2.0.1) con Amazon RDS.	13 Novembre 2018

Amazon RDS for SQL Server supporta Always On	Amazon RDS for SQL Server supporta i gruppi di disponibilità sempre attivi. Per ulteriori informazioni, consulta Implementazioni Multi-AZ per Microsoft SQL Server.	8 Novembre 2018
Amazon RDS for PostgreSQL supporta l'accesso alla rete in uscita con server DNS personalizzati	Amazon RDS for PostgreSQL supporta l'accesso alla rete in uscita con server DNS personalizzati. Per ulteriori informazioni sull'accesso di rete in uscita, inclusi i prerequisiti, consulta Using a Custom DNS Server for Outbound Network Access.	8 Novembre 2018
Amazon RDS for MariaDB, MySQL e PostgreSQL supporta 32 TiB di storage	Ora puoi creare istanze database Amazon RDS for MySQL, MariaDB e PostgreSQL con uno storage fino a 32 TiB. Per ulteriori informazioni, consulta la sezione relativa allo storage per le istanze database.	7 Novembre 2018
Amazon RDS for Oracle supporta i tipi di dati estesi	Ora puoi abilitare i tipi di dati estesi sulle istanze database di Amazon RDS che eseguono Oracle. Con i tipi di dati estesi, le dimensioni massime consentite per i tipi di dati VARCHAR2, NVARCHAR2 e RAW sono di 32.767 byte. Per ulteriori informazioni, consulta la pagina Utilizzo dei tipi di dati estesi.	6 Novembre 2018

Amazon RDS for Oracle supporta le classi di istanze database db.m	Ora puoi creare istanze database di Amazon RDS che eseguono Oracle con le classi di istanze database db.m5. Per ulteriori informazioni, consulta la pagina relativa alla classe di istanza database.	2 novembre 2018
Migrazione di Amazon RDS for Oracle da SE, SE1 o SE2 a EE	Ora puoi eseguire la migrazione da qualsiasi edizione Standard di Oracle Database (SE, SE1 o SE2) all'edizione Enterprise Edition (EE) di Oracle Database. Per ulteriori informazioni, consulta Migrazione tra edizioni Oracle.	31 ottobre 2018
Amazon RDS ora può arrestare le istanze Multi-AZ	Amazon RDS ora può arrestare un'istanza database inclusa in un'implementazione Multi-AZ. In passato la caratteristica di arresto dell'istanza aveva una limitazione per le istanze Multi-AZ. Per ulteriori informazioni, consulta l'argomento relativo all'arresto temporaneo di un'istanza database di Amazon RDS.	29 ottobre 2018
Amazon RDS Approfondimenti sulle prestazioni è disponibile per Amazon RDS for Oracle	Amazon RDS Approfondimenti sulle prestazioni è ora disponibile per Amazon RDS for Oracle. Per ulteriori informazioni, consulta la pagina Utilizzo di Amazon RDS Approfondimenti sulle prestazioni.	29 ottobre 2018

Amazon RDS for PostgreSQL supporta PostgreSQL versione 11 nell'ambiente di anteprima del database	Amazon RDS for PostgreSQL ora supporta PostgreSQL versione 11 nell'ambiente di anteprima del database. Per ulteriori informazioni, consulta PostgreSQL versione 11 su Amazon RDS nell'ambiente di anteprima del database.	25 Ottobre 2018
MySQL supporta una nuova versione principale	Ora puoi creare istanze database di Amazon RDS che eseguono MySQL versione 8.0. Per ulteriori informazioni, consulta Versioni di MySQL in Amazon RDS.	23 ottobre 2018
MariaDB supporta una nuova versione principale	Puoi ora creare istanze database Amazon RDS che eseguono MariaDB versione 10.3. Per ulteriori informazioni, consulta Versioni di MariaDB in Amazon RDS.	23 ottobre 2018
Amazon RDS for Oracle supporta Oracle JVM	Amazon RDS for Oracle ora supporta l'opzione Oracle Java Virtual Machine (JVM). Per ulteriori informazioni, consulta Oracle Java Virtual Machine.	16 ottobre 2018

Gruppo di parametri personalizzato per il recupero e il ripristino point-in-time	Ora puoi specificare un gruppo di parametri personalizzato quando ripristini uno snapshot o esegui un'operazione di ripristino point-in-time. Per ulteriori informazioni, consulta le sezioni Ripristino da uno snapshot DB e Ripristino a un'ora specifica per un'istanza database.	15 ottobre 2018
Amazon RDS for Oracle supporta lo storage a 32 TiB	Ora puoi creare istanze database di Oracle RDS con uno storage massimo di 32 TiB. Per ulteriori informazioni, consulta la sezione relativa allo storage per le istanze database.	15 ottobre 2018
Amazon RDS for MySQL supporta i GTID	Amazon RDS for MySQL ora supporta gli ID globali di transazione (GTID) che sono univoci in tutte le istanze database e in una configurazione della replica. Per ulteriori informazioni, consulta Utilizzo della replica basata su GTID per RDS for MySQL.	10 ottobre 2018
MySQL 5.7.23, 5.6.41 e 5.5.61	Puoi ora creare istanze database Amazon RDS che eseguono MySQL versioni 5.7.23, 5.6.41 e 5.5.61. Per ulteriori informazioni, consulta Versioni di MySQL in Amazon RDS.	8 Ottobre 2018

Amazon RDS for Oracle supporta una nuova versione di SQLT	Amazon RDS for Oracle ora supporta SQLT versione 12.2.180331. Per ulteriori informazioni, consulta Oracle SQLT.	4 ottobre 2018
Amazon RDS for PostgreSQL ora supporta l'autenticazione IAM	Amazon RDS for PostgreSQL ora supporta l'autenticazione IAM. Per ulteriori informazioni, consulta Autenticazione database IAM per MySQL e PostgreSQL.	27 settembre 2018
È possibile abilitare la protezione da eliminazione per le istanze database di Amazon RDS	Quando abiliti la protezione da eliminazione per un'istanza database, il database non può essere eliminato dagli utenti. Per ulteriori informazioni, consulta la sezione relativa all'eliminazione di un'istanza database.	26 settembre 2018
Supporto di classi di istanza database db.m5 in Amazon RDS for MySQL e Amazon RDS for MariaDB	Puoi ora creare istanze database Amazon RDS che eseguono MySQL o MariaDB con classi di istanza database db.m5. Per ulteriori informazioni, consulta la pagina relativa alla classe di istanza database.	18 settembre 2018

Supporto di Amazon RDS per gli aggiornamenti a SQL Server 2017	Puoi ora aggiornare l'istanza database esistente a SQL Server 2017 da qualsiasi versione, ad eccezione di SQL Server 2008. Per eseguire l'aggiornamento da SQL Server 2008, prima di tutto aggiorna l'istanza a una delle altre versioni. Per informazioni, consulta Aggiornamento del motore del database Microsoft SQL Server.	11 settembre 2018
Amazon RDS for PostgreSQL oggi supporta PostgreSQL versione 11 Beta 3 nell'ambiente di anteprima del database	In questa versione le dimensioni dei segmenti WAL (Write-Ahead Log) (wal_segment_size) sono ora impostate su 64 MB. Per ulteriori informazioni su PostgreSQL versione 11 Beta 3, consulta la pagina relativa alla disponibilità di PostgreSQL 11 Beta 3. Per informazioni sull'ambiente di anteprima del database, consulta Lavorare con l'ambiente di anteprima del database.	7 settembre 2018

Guida per l'utente di Amazon Aurora	La Guida per l'utente di Amazon Aurora presenta tutti i concetti correlati ad Amazon Aurora e contiene istruzioni sull'utilizzo delle diverse caratteristiche tramite la console e l'interfaccia a riga di comando. La Guida per l'utente di Amazon RDS include ora informazioni anche sui motori di database non Aurora.	31 agosto 2018
Amazon RDS Approfondimenti sulle prestazioni è disponibile per RDS for MySQL	Amazon RDS Approfond imenti sulle prestazioni è ora disponibile per RDS for MySQL. Per ulteriori informazi oni, consulta la pagina Utilizzo di Amazon RDS Approfond imenti sulle prestazioni.	28 agosto 2018
Supporto di Aurora Auto Scaling in Aurora edizione compatibile con PostgreSQL	È ora disponibile il servizio Auto Scaling di repliche Aurora per Aurora edizione compatibi le con PostgreSQL. Per ulteriori informazioni, consulta la pagina relativa all'utilizzo di Amazon Aurora Auto Scaling con repliche Aurora	16 agosto 2018
Aurora Serverless per Aurora MySQL	Aurora Serverless è una configurazione di Auto Scaling on demand per Amazon Aurora. Per ulteriori informazi oni, consulta Utilizzo di Amazon Aurora Serverless.	9 agosto 2018

MySQL 5.7.22 e 5.6.40	Puoi ora creare istanze database Amazon RDS che eseguono MySQL versioni 5.7.22 e 5.6.40. Per ulteriori informazioni, consulta Versioni di MySQL in Amazon RDS.	6 agosto 2018
Disponibilità di Aurora nella regione Cina (Ningxia)	Aurora MySQL e Aurora PostgreSQL sono ora disponibili nella regione Cina (Ningxia). Per ulteriori informazioni, consulta le pagine relative alla disponibilità per Amazon Aurora MySQL e alla disponibilità per Amazon Aurora PostgreSQL.	6 agosto 2018
Supporto della replica ritardata in Amazon RDS for MySQL	Amazon RDS for MySQL supporta ora la replica ritardata come strategia per il ripristino di emergenza . Per ulteriori informazioni, consulta la sezione Configurazione della funzione di replica ritardata con MySQL.	6 agosto 2018
Disponibilità di Amazon RDS Approfondimenti sulle prestazioni per Aurora MySQL	Amazon RDS Approfondimenti sulle prestazioni è ora disponibile per Aurora MySQL. Per ulteriori informazioni, consulta la pagina Utilizzo di Amazon RDS Approfondimenti sulle prestazioni.	6 agosto 2018

Integrazione di Amazon RDS Performance Insights con Amazon CloudWatch	Amazon RDS Performance Insights pubblica automatic amente i parametri su Amazon. CloudWatch Per ulteriori informazioni, consulta le metriche di Performan ce Insights pubblicate su. CloudWatch	6 agosto 2018
Consigli di Amazon RDS	Amazon RDS offre ora consigli automatici per le risorse di database. Per ulteriori informazioni, consulta la pagina relativa all'Utilizzo dei consigli per Amazon RDS.	25 luglio 2018
Copie incrementali di istantane e tra diverse regioni AWS	Amazon RDS supporta copie incrementali di snapshot in tutte le AWS regioni per istanze crittografate e non crittografate. Per ulteriori informazioni, consulta Copiare istantanee tra regioni. AWS	24 luglio 2018
Disponibilità di Amazon RDS Approfondimenti sulle prestazioni per Amazon RDS for PostgreSQL	Amazon RDS Approfond imenti sulle prestazioni è ora disponibile per Amazon RDS for PostgreSQL. Per ulteriori informazioni, consulta la pagina Utilizzo di Amazon RDS Approfondimenti sulle prestazioni.	18 luglio 2018

Supporto di Oracle APEX versione 5.1.4.v1 in Amazon RDS for Oracle	Amazon RDS for Oracle supporta ora Oracle Application Express (APEX) versione 5.1.4.v1. Per ulteriori informazioni, consulta Oracle Application Express.	10 luglio 2018
Amazon RDS per Oracle supporta la pubblicazione di log su Amazon Logs CloudWatch	Amazon RDS for Oracle ora supporta la pubblicazione di dati di log di alert, audit, trace e listener in un gruppo CloudWatch di log in Logs. Per ulteriori informazioni, consulta Pubblicazione dei log Oracle su Amazon CloudWatch Logs.	9 luglio 2018
MariaDB 10.2.15, 10.1.34 e 10.0.35	Puoi ora creare istanze database Amazon RDS che eseguono MariaDB versioni 10.2.15, 10.1.34 e 10.0.35. Per ulteriori informazioni, consulta Versioni di MariaDB in Amazon RDS.	5 luglio 2018
Disponibilità e compatibilità con PostgreSQL 9.6.8 di Aurora PostgreSQL 1.2	Aurora PostgreSQL 1.2 è ora disponibile ed è compatibile con PostgreSQL 9.6.8. Per ulteriori informazioni, consulta Versione 1.2.	27 giugno 2018

Implementazioni Multi-AZ supportate dalle repliche di lettura per Amazon RDS for PostgreSQL	Le repliche di lettura RDS in Amazon RDS for PostgreSQL supportano ora più zone di disponibilità. Per ulteriori informazioni, consulta la pagina relativa al funzionamento delle repliche di lettura PostgreSQL.	25 giugno 2018
Disponibilità di Approfondimenti sulle prestazioni per Aurora PostgreSQL	Approfondimenti sulle prestazioni è disponibile a livello generale per Aurora PostgreSQL, con supporto per la conservazione estesa dei dati sulle prestazioni. Per ulteriori informazioni, consulta Utilizzo di Amazon RDS Approfondimenti sulle prestazioni.	21 giugno 2018
Disponibilità di Aurora PostgreSQL nella regione Stati Uniti occidentali (California settentrionale)	Aurora PostgreSQL è ora disponibile nella regione Stati Uniti occidentali (California settentrionale). Per ulteriori informazioni, consulta la pagina relativa alla disponibilità per Amazon Aurora PostgreSQL.	11 giugno 2018

Supporto della configurazione della CPU in Amazon RDS for Oracle	Amazon RDS for Oracle supporta ora la configurazione del numero di core CPU e del numero di thread per ogni core per il processore di una classe dell'istanza database. Per ulteriori informazioni, consulta Configurazione del processore della classe di istanza database.	5 giugno 2018

Aggiornamenti precedenti

La tabella seguente descrive le modifiche importanti introdotte in ogni versione della Guida per l'utente di Amazon RDS prima di giugno 2018.

Modifica	Descrizione	Data della modifica
Amazon RDS for PostgreSQL oggi supporta PostgreSQL Version 11 Beta 1 nell'ambiente di anteprima del database	PostgreSQL versione 11 Beta 1 include vari miglioramenti, descritti in PostgreSQL 11 Beta 1 Released Per informazioni sull'ambiente di anteprima del database, consulta Utilizzo dell'ambiente di anteprima del database.	31 maggio 2018
Amazon RDS for Oracle oggi supporta TLS versione 1.0 e 1.2	Amazon RDS for Oracle supporta Transport Layer Security (TLS) versioni 1.0 e 1.2. Per ulteriori informazioni, consulta Versioni TLS per l'opzione SSL di Oracle.	30 maggio 2018
Aurora MySQL supporta la pubblicazione di	Aurora MySQL ora supporta la pubblicazione di dati di log generici, lenti, di controllo ed errori in un gruppo di log in Logs. CloudWatch Per ulteriori informazioni,	23 maggio 2018

Modifica	Descrizione	Data della modifica
log su Amazon Logs CloudWatch	consulta Pubblicazione di Aurora MySQL su Logs. CloudWatch	
Ambiente di anteprima del database per Amazon RDS PostgreSQL	Ora puoi avviare una nuova istanza Amazon RDS PostgreSQL in modalità di anteprima. Per ulteriori informazioni sull'ambiente di anteprima del database, consulta Utilizzo dell'ambiente di anteprima del database.	22 maggio 2018
Le istanze database di Amazon RDS for Oracle supportano nuove classi di istanze database	Le istanze database Oracle oggi supportano classi di istanze database db.x1e e db.x1. Per ulteriori informazioni, consulta Classi di istanze database e Classi di istanza RDS for Oracle.	22 maggio 2018
Amazon RDS PostgreSQL supporta ora postgres_fdw in una replica di lettura.	Ora puoi utilizzare postgres_fdw per la connessione a un server remoto da una replica di lettura. Per ulteriori informazioni, consulta, Utilizzo dell'estensione postgres_fdw per accedere a dati esterni.	17 maggio 2018
Amazon RDS for Oracle ora supporta i parametri sqlnet.ora	Oggi puoi impostare i parametri sqlnet.ora con Amazon RDS for Oracle. Per ulteriori informazioni, consulta Modifica delle proprietà di connessione tramite i parametri sqlnet.ora.	10 maggio 2018
Aurora PostgreSQL disponibile nella regione Asia Pacifico (Seoul).	Aurora PostgreSQL è ora disponibile nella regione Asia Pacifico (Seoul). Per ulteriori informazioni, consulta la pagina relativa alla disponibilità per Amazon Aurora PostgreSQL.	9 maggio 2018

Modifica	Descrizione	Data della modifica
Aurora MySQL supporta il backtrack	Aurora MySQL ora supporta il "riavvolgimento" di un cluster di database a un'ora specifica, senza ripristinare i dati di un backup. Per ulteriori informazioni, consulta la pagina relativa al backtrack di un cluster di database Aurora.	9 maggio 2018
Aurora MySQL supporta la migrazione e la replica crittografate da MySQL esterno	Aurora MySQL ora supporta la migrazione e la replica crittografate da un database MySQL esterno. Per ulteriori informazioni consulta le pagine relative alla migrazione di dati da un database MySQL esterno a un cluster di database Amazon Aurora MySQL e alla funzione di replica tra Aurora e MySQL o tra Aurora e un altro cluster di database Aurora.	25 Aprile 2018
Supporto del protocollo di copia su scrittura da parte di Aurora edizione compatibile con PostgreSQL.	Ora puoi clonare i database in un cluster di database Aurora PostgreSQL. Per ulteriori informazioni, consulta la pagina relativa alla clonazione di database in un cluster di database Aurora.	10 Aprile 2018
MariaDB 10.2.12, 10.1.31 e 10.0.34	Puoi ora creare istanze database Amazon RDS che eseguono MariaDB versioni 10.2.12, 10.1.31 e 10.0.34. Per ulteriori informazioni, consulta Versioni di MariaDB in Amazon RDS.	21 marzo 2018
Supporto delle nuove regioni da parte di Aurora PostgreSQL	Aurora PostgreSQL è ora disponibile nella regione UE (Londra) e Asia Pacifico (Singapore). Per ulteriori informazioni, consulta la pagina relativa alla disponibilità per Amazon Aurora PostgreSQL.	13 marzo 2018

Modifica	Descrizione	Data della modifica
MySQL 5.7.21, 5.6.39 e 5.5.59	Puoi ora creare istanze database Amazon RDS che eseguono MySQL versioni 5.7.21, 5.6.39 e 5.5.59. Per ulteriori informazioni, consulta Versioni di MySQL in Amazon RDS.	9 marzo 2018
Amazon RDS for Oracle ora supporta Oracle REST Data Services	Amazon RDS for Oracle ora supporta Oracle REST Data Services nell'ambito dell'opzione APEX. Per ulteriori informazioni, consulta Oracle Application Express (APEX).	9 marzo 2018
Edizione compatibile con Amazon Aurora con MySQL disponibile in una nuova regione AWS	Aurora MySQL è ora disponibile nella regione Asia Pacifico (Singapore). Per l'elenco completo delle AWS regioni per Aurora MySQL, consulta Disponibilità per Amazon Aurora MySQL.	6 marzo 2018
Le istanze database Amazon RDS che eseguono Microsoft SQL Server supportano la funzionalità CDC (Change Data Capture)	Le istanze database che eseguono Amazon RDS for Microsoft SQL Server ora supportano la funzionalità CDC (Change Data Capture). Per ulteriori informazioni, consulta Cambia il supporto Data Capture per le istanze del database di Microsoft SQL Server.	6 febbraio 2018
Aurora MySQL supporta una nuova versione principale	Puoi ora creare cluster di database Aurora MySQL che eseguono MySQL versione 5.7. Per ulteriori informazioni, consulta la pagina relativa agli aggiornamenti del motore del database Amazon Aurora MySQL del 6/2/2018.	6 febbraio 2018

Modifica	Descrizione	Data della modifica
Pubblica log MySQL e MariadB su Amazon Logs CloudWatch	Ora puoi pubblicare i dati di registro MySQL e MariaDB su Logs. CloudWatch Per ulteriori informazioni, consulta Pubblicazione dei log MySQL su Amazon Logs CloudWatch e Pubblicazione dei log di MariadB su Amazon Logs CloudWatch .	17 gennaio 2018
Supporto Multi-AZ per le repliche di lettura	È ora possibile creare una replica di lettura come istanza database Multi-AZ. Amazon RDS crea una replica di standby in un'altra zona di disponibilità per il supporto del failover per la replica. La creazione della replica di lettura come un'istanza database Multi-AZ non dipende dal fatto che il database di origine sia un'istanza database Multi-AZ. Per ulteriori informazi oni, consulta Uso delle repliche di lettura dell'istanza database.	11 gennaio 2018
Amazon RDS for MariaDB supporta una nuova versione principale	Puoi ora creare istanze database Amazon RDS che eseguono MariaDB versione 10.2. Per ulteriori informazioni, consulta Supporto per MariaDB 10.2 in Amazon RDS.	3 gennaio 2018
Amazon Aurora edizione compatibi le con PostgreSQ L disponibile in una nuova regione AWS	Aurora PostgreSQL è ora disponibile nella regione UE (Parigi). Per l'elenco completo delle AWS regioni per Aurora PostgreSQL, consulta Disponibilità per Amazon Aurora PostgreSQL.	22 dicembre 2017
Supporto di nuovi tipi di istanza in Aurora PostgreSQ L	Aurora PostgreSQL ora supporta nuovi tipi di istanze. Per l'elenco completo dei tipi di istanza, consulta la pagina relativa alla scelta della classe di istanza database.	20 dicembre 2017

Modifica	Descrizione	Data della modifica
Edizione compatibile con Amazon Aurora con MySQL disponibile in una nuova regione AWS	Aurora MySQL è ora disponibile nella regione UE (Parigi). Per l'elenco completo delle AWS regioni per Aurora MySQL, consulta Disponibilità per Amazon Aurora MySQL.	18 dicembre 2017
Aurora MySQL supporta gli hash join	Questa funzionalità può migliorare le prestazioni delle query se devi eseguire il join di un'ingente quantità di dati tramite una query equijoin. Per ulteriori informazioni, consulta la pagina relativa al funzionamento di hash join in Aurora MySQL.	11 dicembre 2017
Aurora MySQL supporta le funzioni native per richiamare le funzioni AWS Lambda	Puoi chiamare le funzioni native `lambda_sync` e `lambda_async` quando usi Aurora MySQL. Per ulteriori informazioni, consulta la pagina relativa alla chiamata di una funzione Lambda da un cluster di database Amazon Aurora MySQL.	11 dicembre 2017
Aggiunta l'idoneità HIPAA di Aurora PostgreSQL	Aurora PostgreSQL ora supporta la creazione di applicazioni conformi a HIPAA. Per ulteriori informazioni, consulta la pagina relativa al funzionamento di Amazon Aurora PostgreSQL.	6 dicembre 2017
AWS Regioni aggiuntive disponibili per Amazon Aurora con compatibilità PostgreSQL	Amazon Aurora con compatibilità PostgreSQL è ora disponibile in quattro nuove regioni. AWS Per ulteriori informazioni, consulta la pagina relativa alla disponibilità per Amazon Aurora PostgreSQL.	22 Novembre 2017

Modifica	Descrizione	Data della modifica
Modifica dello storage delle istanze database Amazon RDS che eseguono Microsoft SQL Server	Ora puoi modificare lo storage delle istanze database Amazon RDS che eseguono Microsoft SQL Server. Per ulteriori informazioni, consulta Modifica di un'istanza database Amazon RDS.	21 Novembre 2017
Amazon RDS lo storage a 16 TiB per i motori basati su Linux	Ora puoi creare istanze database RDS per MySQL, MariaDB, PostgreSQL e Oracle con uno storage fino a 16 TiB. Per ulteriori informazioni, consulta Storage delle istanze di database Amazon RDS.	21 Novembre 2017
Amazon RDS supporta l'aumento rapido delle dimensioni dello storage	Ora puoi aggiungere storage alle istanze database RDS per MySQL, MariaDB, PostgreSQL e Oracle in pochi minuti. Per ulteriori informazioni, consulta Storage delle istanze di database Amazon RDS.	21 Novembre 2017
Amazon RDS supporta MariaDB versione 10.1.26 e 10.0.32	Puoi ora creare istanze database Amazon RDS che eseguono MariaDB versioni 10.1.26 e 10.0.32. Per ulteriori informazioni, consulta Versioni di MariaDB in Amazon RDS.	20 Novembre 2017
Amazon RDS for Microsoft SQL Server ora supporta nuove classi di istanze database	Ora puoi creare istanze database Amazon RDS che eseguono SQL Server che utilizzano classi di istanze database db.r4 e db.m4.16xlarge. Per ulteriori informazioni, consulta Supporto classe istanza database per Microsoft SQL Server.	20 Novembre 2017

Modifica	Descrizione	Data della modifica
Amazon RDS for MySQL e MariaDB ora supporta nuove istanze database	Ora puoi creare istanze database Amazon RDS che eseguono MySQL e MariaDB che utilizzano classi di istanze database db.r4, db.m4.16xlarge, db.t2.xlarge e db.t2.2xlarge. Per ulteriori informazioni, consulta Classi di istanze database.	20 Novembre 2017
SQL Server 2017	Ora puoi creare istanze database Amazon RDS che eseguono Microsoft SQL Server 2017. Puoi anche creare istanze database che eseguono SQL Server 2016 SP1 CU5. Per ulteriori informazioni, consulta Amazon RDS for Microsoft SQL Server.	17 Novembre 2017
Ripristino dei backup MySQL da Simple Storage Service (Amazon S3)	Puoi ora creare un backup del database locale, archiviarlo in Simple Storage Service (Amazon S3) e quindi ripristinare il file di backup in una nuova istanza database Amazon RDS che esegue MySQL. Per ulteriori informazioni, consulta Ripristino di un backup in un'istanza database MySQL.	17 Novembre 2017
Auto Scaling con repliche Aurora	Amazon Aurora MySQL supporta ora Aurora Auto Scaling. Aurora Auto Scaling modifica in modo dinamico il numero delle repliche Aurora in base all'aumento o alla riduzione della connettività o del carico di lavoro. Per ulteriori informazioni, consulta la pagina Utilizzo di Amazon Aurora Auto Scaling con repliche Aurora.	17 Novembre 2017
Supporto dell'edizione predefinita Oracle	Le istanze database di Amazon RDS for Oracle ora supportano l'impostazione dell'edizione predefinita dell'istanza database. Per ulteriori informazioni, consulta Impostazione dell'edizione predefinita per un'istanza database.	3 Novembre 2017

Modifica	Descrizione	Data della modifica
Convalida dei file dell'istanza database Oracle	Le istanze database di Amazon RDS for Oracle DB ora supportano la convalida dei file delle istanze database con l'utilità di convalida logica Oracle Recovery Manager (RMAN). Per ulteriori informazioni, consulta Convalida dei file di database in RDS per Oracle .	3 Novembre 2017
Management Agent per OEM 13c	Le istanze database Amazon RDS for Oracle ora supportano il Management Agent per Oracle Enterpris e Manager (OEM) Cloud Control 13c. Per ulteriori informazioni, consulta Oracle Management Agent per Enterprise Manager Cloud Control.	1 Novembre 2017
Riconfigurazione dello storage per le snapshot di Microsoft SQL Server	Ora puoi riconfigurare lo storage durante il ripristino di uno snapshot in un'istanza database Amazon RDS che esegue Microsoft SQL Server. Per ulteriori informazi oni, consulta Ripristino da uno snapshot database.	26 ottobre 2017
Prefetch asincrono delle chiavi per Aurora edizione compatibile con MySQL	Il prefetch asincrono delle chiavi (AKP) migliora le prestazioni dei join degli indici non memorizzati nella cache, attraverso il prefetch delle chiavi in memoria prima che diventino necessarie. Per ulteriori informazi oni, consulta la pagina relativa al funzionamento del prefetch asincrono delle chiavi in Amazon Aurora.	26 ottobre 2017
MySQL 5.7.19, 5.6.37 e 5.5.57	Puoi ora creare istanze database Amazon RDS che eseguono MySQL versioni 5.7.19, 5.6.37 e 5.5.57. Per ulteriori informazioni, consulta Versioni di MySQL in Amazon RDS.	25 ottobre 2017

Modifica	Descrizione	Data della modifica
Disponibilità generale di Amazon Aurora con compatibilità PostgreSQL	Amazon Aurora con compatibilità PostgreSQL semplifica e rende più conveniente dal punto di vista dei costi la configurazione, l'utilizzo e il dimension amento delle implementazioni PostgreSQL nuove ed esistenti, consentendoti di concentrarti sulle tue attività e applicazioni. Per ulteriori informazioni, consulta la pagina relativa al funzionamento di Amazon Aurora PostgreSQL.	24 ottobre 2017
Le istanze database di Amazon RDS for Oracle supportan o nuove classi di istanze database	Le istanze database Amazon RDS for Oracle ora supportano classi di istanze di nuova generazione ottimizzate per la memoria (db.r4). Inoltre, le istanze database Amazon RDS for Oracle ora supportano le seguenti nuove classi di istanze di generazione corrente: db.m4.16xlarge, db.t2.xlarge e db.t2.2xlarge. Per ulteriori informazioni, consulta Classi di istanze database e Classi di istanza RDS for Oracle.	23 ottobre 2017
Nuova caratteri stica	Le istanze riservate nuove ed esistenti possono ora avere varie dimensioni nella stessa classe di istanze database. Le istanze riservate flessibili in termini di dimensioni sono disponibili per le istanze DB con la stessa AWS regione, motore di database e famiglia di istanze e in tutta la configurazione AZ. Le istanze riservate con dimensioni flessibili sono disponibili per i seguenti motori di database: Amazon Aurora, MariaDB, MySQL, Oracle (Bring-Your-Own-License (uso di licenze proprie)), PostgreSQL. Per ulteriori informazioni, consulta Istanze database riservate con dimensioni flessibili.	11 ottobre 2017
Nuova caratteri stica	Ora puoi utilizzare l'opzione Oracle SQLT per regolare un'istruzione SQL e ottimizzare le prestazioni. Per ulteriori informazioni, consulta Oracle SQLT.	22 settembre 2017

Modifica	Descrizione	Data della modifica
Nuova caratteristica	Se hai effettuato snapshot DB manuali delle istanze database Amazon RDS for Oracle, puoi ora aggiornarle a una versione successiva del motore del database Oracle. Per ulteriori informazioni, consulta Aggiornamento di uno shapshot DB Oracle.	20 settembre 2017
Nuova caratteristica	Ora puoi utilizzare Oracle Spatial per archiviare, recuperare ed effettuare query dei dati spaziali nelle istanze database Amazon RDS che eseguono Oracle. Per ulteriori informazioni, consulta Oracle Spatial.	15 settembre 2017
Nuova caratteristica	Ora puoi utilizzare Oracle Locator per il supporto di applicazioni Internet e wireless basate su servizi e di soluzioni GIS basate su partner con le istanze database Amazon RDS che eseguono Oracle. Per ulteriori informazioni, consulta Oracle Locator.	15 settembre 2017
Nuova caratteristica	Ora puoi utilizzare Oracle Multimedia per archiviare, gestire e recuperare immagini, audio, video e altri dati di supporti eterogenei nelle istanze database Amazon RDS che eseguono Oracle. Per ulteriori informazioni, consulta Oracle Multimedia.	15 settembre 2017
Nuova caratteristica	Ora puoi esportare i log di controllo dai cluster Amazon Aurora MySQL DB ad Amazon Logs. CloudWatch Per ulteriori informazioni, consulta Pubblicazione dei log di Aurora MySQL su Amazon Logs. CloudWatch	14 settembre 2017
Nuova caratteristica	Amazon RDS ora supporta più versioni di Oracle Application Express (APEX) per le istanze database che eseguono Oracle. Per ulteriori informazioni, consulta Oracle Application Express (APEX).	13 settembre 2017

Modifica	Descrizione	Data della modifica
Nuova caratteristica	Puoi ora usare Amazon Aurora per eseguire la migrazione di uno snapshot DB non crittografato o crittografato o di un'istanza database MySQL in un cluster di database Aurora MySQL crittografato. Per ulteriori informazioni, consulta Migrazione di uno snapshot RDS for MySQL in Aurora e Migrazione dei dati da un'istanza database MySQL in un cluster di database Amazon Aurora MySQL utilizzando una replica di lettura di Aurora.	5 settembre 2017
Nuova caratteristica	Puoi utilizzare i database Amazon RDS for Microsoft SQL Server per creare applicazioni conformi a HIPAA. Per ulteriori informazioni, consulta Supporto del Programma di Conformità per le istanze di database di Microsoft SQL Server.	31 agosto 2017
Nuova caratteristica	Ora puoi utilizzare i database Amazon RDS for MariaDB per creare applicazioni conformi a HIPAA. Per ulteriori informazioni, consulta Amazon RDS for MariaDB.	31 agosto 2017
Nuova caratteristica	Puoi ora creare istanze database Amazon RDS che eseguono Microsoft SQL Server con storage allocato fino a 16 TiB e Provisioned IOPS per intervalli di storage da 1:1 a 50:1. Per ulteriori informazioni, consulta Storage delle istanze di database Amazon RDS.	22 agosto 2017
Nuova caratteristica	Ora puoi utilizzare le implementazioni Multi-AZ per le istanze database che eseguono Microsoft SQL Server nella regione UE (Francoforte). Per ulteriori informazioni, consulta Implementazioni Multi-AZ per Amazon RDS for Microsoft SQL Server.	3 agosto 2017

Modifica	Descrizione	Data della modifica
Nuova caratteristica	Puoi ora creare istanze database Amazon RDS che eseguono MariaDB versioni 10.1.23 e 10.0.31. Per ulteriori informazioni, consulta Versioni di MariaDB in Amazon RDS.	17 luglio 2017
Nuova caratteristica	Amazon RDS ora supporta Microsoft SQL Server Enterprise Edition con il modello Licenza inclusa in tutte le AWS regioni. Per ulteriori informazioni, consulta Licenza per Microsoft SQL Server su Amazon RDS.	13 luglio 2017
Nuova caratteristica	Amazon RDS for Oracle ora supporta huge pages del kernel di Linux per una maggiore scalabilità del database. L'utilizzo di Huge Pages comporta tabelle di pagina più piccole e meno tempo CPU dedicato alla gestione della memoria, aumentando le prestazioni di istanze database grandi. Puoi ora usare pagine di grandi dimensioni con le istanze database Amazon RDS con tutte le edizioni di Oracle versioni 12.1.0.2 e 11.2.0.4. Per ulteriori informazioni, consulta Attivazione di HugePages per un'istanza RDS per Oracle.	7 luglio 2017
Nuova caratteristica	Versione aggiornata per il supporto della crittografia dei dati inattivi (EAR) per le istanze database db.t2.small e db.t2.medium per tutti motori di database diversi da Aurora. Per ulteriori informazioni, consulta Disponibilità della crittografia Amazon RDS.	27 giugno 2017
Nuova caratteristica	Versione aggiornata per il supporto di Amazon Aurora nella regione Europa (Francoforte). Per ulteriori informazioni, consulta la pagina relativa alla disponibilità per Amazon Aurora MySQL.	16 giugno 2017

Modifica	Descrizione	Data della modifica
Nuova caratteristica	Ora puoi specificare un gruppo di opzioni quando copi uno snapshot DB tra AWS regioni. Per ulteriori informazioni, consulta Considerazioni su gruppi di opzioni.	12 giugno 2017
Nuova caratteristica	Ora puoi copiare istantanee DB create da istanze DB specializzate tra regioni. AWS Puoi copiare snapshot da istanze database che utilizzano Oracle TDE, Microsoft SQL Server TDE e Microsoft SQL Server Multi-AZ con mirroring. Per ulteriori informazioni, consulta Copia di una snapshot DB..	12 giugno 2017
Nuova caratteristica	Amazon Aurora ti consente ora di copiare in modo rapido ed efficiente in termini di costi tutti i database in un cluster di database Amazon Aurora. Per ulteriori informazioni, consulta la pagina relativa alla clonazione di database in un cluster di database Aurora.	12 giugno 2017
Nuova caratteristica	Amazon RDS ora supporta Microsoft SQL Server 2016 SP1 CU2. Per ulteriori informazioni, consulta Amazon RDS for Microsoft SQL Server.	7 giugno 2017
Anteprima	Anteprima pubblica di Amazon Aurora con compatibilità PostgreSQL. Per ulteriori informazioni, consulta la pagina relativa al funzionamento di Amazon Aurora PostgreSQL.	19 Aprile 2017
Nuova caratteristica	Amazon Aurora permette ora di eseguire un'operazione ALTER TABLE nome_tabella ADD COLUMN nome_colonna definizione_colonna quasi istantaneamente. L'operazione si conclude senza che vi sia necessità di copiare la tabella e senza alcuna conseguenza materiale sulle altre istruzioni DML. Per ulteriori informazioni, consulta la pagina relativa alla modifica di tabelle in Amazon Aurora tramite Fast DDL.	5 Aprile 2017

Modifica	Descrizione	Data della modifica
Nuova caratteristica	Abbiamo aggiunto un nuovo comando di monitoraggio, SHOW VOLUME STATUS, per visualizzare il numero di nodi e di dischi di un volume. Per ulteriori informazioni, consulta la pagina relativa alla visualizzazione dello stato del volume per un cluster di database Aurora.	5 Aprile 2017
Nuova caratteristica	Ora puoi utilizzare la logica personalizzata nelle funzioni personalizzate di verifica della password per Oracle su Amazon RDS. Per ulteriori informazioni, consulta Creazione delle funzionalità personalizzate per verificare le password.	21 marzo 2017
Nuova caratteristica	Ora puoi accedere ai file di log redo online e archiviati nelle istanze database Oracle su Amazon RDS. Per ulteriori informazioni, consulta Accesso ai log di ripristino online e archiviati.	21 marzo 2017
Nuova caratteristica	Ora puoi copiare snapshot dei cluster di database crittografati e non crittografati fra account della stessa regione. Per ulteriori informazioni, consulta Copia di uno snapshot di un cluster di database tra account.	7 marzo 2017
Nuova caratteristica	Ora puoi condividere snapshot dei cluster di database crittografati fra account della stessa regione. Per ulteriori informazioni, consulta Condivisione di uno snapshot cluster di database.	7 marzo 2017
Nuova caratteristica	Puoi ora replicare cluster di database Amazon Aurora MySQL crittografati per creare repliche Aurora tra regioni. Per ulteriori informazioni, consulta Replica dei cluster Aurora MySQL DB tra le regioni. AWS	7 marzo 2017

Modifica	Descrizione	Data della modifica
Nuova caratteristica	Ora puoi richiedere che tutte le connessioni all'istanza database che esegue Microsoft SQL Server utilizzino il Secure Sockets Layer (SSL). Per ulteriori informazioni, consulta Utilizzo di SSL con un'istanza database Microsoft SQL Server.	27 febbraio 2017
Nuova caratteristica	Ora puoi impostare il fuso orario locale su uno degli altri 15 fusi orari. Per ulteriori informazioni, consulta Fusi orari supportati.	27 febbraio 2017
Nuova caratteristica	Puoi ora usare la procedura Amazon RDS `msdb.dbo.rds_shrink_tempdbfile` per ridurre il database tempdb nelle istanze database che eseguono Microsoft SQL Server. Per ulteriori informazioni, consulta Riduzione del database tempdb.	17 febbraio 2017
Nuova caratteristica	Puoi ora comprimere il file di backup quando esporti il database Microsoft SQL Server Enterprise e Standard Edition da un'istanza database Amazon RDS a Simple Storage Service (Amazon S3). Per ulteriori informazioni, consulta Compressione dei file di backup.	17 febbraio 2017
Nuova caratteristica	Amazon RDS ora supporta server DNS personalizzati per la risoluzione dei nomi DNS nell'accesso di rete in uscita sulle istanze database che eseguono Oracle. Per ulteriori informazioni, consulta Impostazione di un server DNS personalizzato.	26 gennaio 2017
Nuova caratteristica	Amazon RDS ora supporta la creazione di una replica di lettura crittografata in un'altra regione. Per ulteriori informazioni, consulta e CreateDB. Creazione di una replica di lettura in un altro Regione AWS InstanceReadReplica	23 gennaio 2017

Modifica	Descrizione	Data della modifica
Nuova caratteristica	Amazon RDS supporta ora l'aggiornamento di uno snapshot DB MySQL da MySQL 5.1 a MySQL 5.5.	20 gennaio 2017
Nuova caratteristica	Amazon RDS support ora la copia di uno snapshot DB crittografata in un'altra regione per i motori di database MariaDB, MySQL, Oracle, PostgreSQL e Microsoft SQL Server. Per ulteriori informazioni, consulta Copia di una snapshot DB. e CopyDBSnapshot.	20 dicembre 2016
Nuova caratteristica	Amazon Aurora MySQL ora supporta l'indicizzazione spaziale. L'indicizzazione spaziale consente di migliorare le prestazioni delle query su set di dati di grandi dimensioni per le query che usano i dati spaziali. Per ulteriori informazioni, consulta la pagina relativa a Amazon Aurora MySQL e dati spaziali.	14 dicembre 2016
Nuova caratteristica	Amazon RDS ora supporta l'accesso di rete in uscita sull'istanza database che esegue Oracle. È possibile utilizzare utl_http, utl_tcp e utl_smtp per connetterti alla rete dall'istanza database. Per ulteriori informazioni, consulta Configurazione dell'accesso UTL_HTTP utilizzando certificati e un portafoglio Oracle.	5 dicembre 2016
Nuova caratteristica	Amazon RDS ha sospeso il supporto per MySQL versione 5.1. Puoi tuttavia ripristinare gli snapshot MySQL 5.1 esistenti in un'istanza MySQL 5.5. Per ulteriori informazioni, consulta Motori di storage supportati per RDS for MySQL.	15 Novembre 2016
Nuova caratteristica	Amazon RDS ora supporta Microsoft SQL Server 2016 RTM CU2. Per ulteriori informazioni, consulta Amazon RDS for Microsoft SQL Server.	4 Novembre 2016

Modifica	Descrizione	Data della modifica
Nuova caratteristica	Amazon RDS ora supporta gli aggiornamenti principali delle istanze database che eseguono Oracle. Ora puoi aggiornare le istanze database Oracle da 11g a 12c. Per ulteriori informazioni, consulta Aggiornamento del motore di database RDS per Oracle.	2 Novembre 2016
Nuova caratteristica	Ora puoi creare istanze database che eseguono Microsoft SQL Server 2014 Enterprise Edition. Amazon RDS supporta ora SQL Server 2014 SP2 per tutte le edizioni e le regioni. Per ulteriori informazioni, consulta Amazon RDS for Microsoft SQL Server.	25 ottobre 2016
Nuova caratteristica	Amazon Aurora MySQL ora si integra con altri AWS servizi: puoi caricare testo o dati XML in una tabella da un bucket Amazon S3 o richiamare una funzione dal codice del database. AWS Lambda Per ulteriori informazioni, consulta Integrazione di Aurora MySQL con altri servizi. AWS	18 Ottobre 2016
Nuova caratteristica	Puoi ora accedere al database tempdb nelle istanze database Amazon RDS che eseguono Microsoft SQL Server. Puoi accedere al database tempdb con Transact-SQL tramite Microsoft SQL Server Management Studio (SSMS) o qualsiasi altra applicazione client SQL standard. Per ulteriori informazioni, consulta Accesso al database tempdb sulle istanze database Microsoft SQL Server su Amazon RDS.	29 settembre 2016
Nuova caratteristica	Ora puoi utilizzare il pacchetto UTL_MAIL con le istanze database Amazon RDS che eseguono Oracle. Per ulteriori informazioni, consulta UTL_MAIL di Oracle.	20 settembre 2016

Modifica	Descrizione	Data della modifica
Nuove caratteristiche	Ora puoi impostare il fuso orario delle nuove istanze database Microsoft SQL Server su un fuso orario locale, per farlo corrispondere a quello delle applicazioni. Per ulteriori informazioni, consulta Fuso orario locale per le istanze di database di Microsoft SQL Server.	19 settembre 2016
Nuova caratteristica	Ora puoi utilizzare l'opzione Oracle Label Security per controllare l'accesso a singole righe delle tabelle nelle tue istanze database Amazon RDS che eseguono Oracle Database 12c. Con Oracle Label Security, puoi applicare la conformità normativa con un modello amministrativo basato su policy e garantire che l'accesso a dati sensibili sia limitato solo agli utenti con il livello adeguato di autorizzazioni. Per ulteriori informazioni, consulta Oracle Label Security.	8 settembre 2016
Nuova caratteristica	Ora puoi effettuare la connessione a un cluster di database Amazon Aurora con l'endpoint di lettura, che bilancia il carico delle connessioni nelle repliche Aurora disponibili nel cluster di database. Man mano che i client richiedono nuove connessioni all'endpoint di lettura, Aurora distribuisce tali richieste fra le repliche di Aurora nel cluster di database. Questa funzionalità consente di bilanciare il carico di lavoro di lettura fra più repliche Aurora nel cluster di database. Per ulteriori informazioni, consulta la pagina relativa agli endpoint Amazon Aurora.	8 settembre 2016

Modifica	Descrizione	Data della modifica
Nuova caratteristica	Le istanze database Amazon RDS che eseguono Oracle ora supportano Oracle Enterprise Manager Cloud Control. Puoi abilitare il Management Agent nelle istanze database e condividere i dati con Oracle Management Service (OMS). Per ulteriori informazioni, consulta Oracle Management Agent per Enterprise Manager Cloud Control.	1 settembre 2016
Nuova caratteristica	Con questa versione è stato aggiunto il supporto per l'ottenimento di un ARN per una risorsa. Per ulteriori informazioni, consulta Recupero di un ARN esistente.	23 agosto 2016
Nuova caratteristica	Ora puoi assegnare fino a 50 tag a ciascuna risorsa Amazon RDS per gestire le risorse e tenere traccia dei costi. Per ulteriori informazioni, consulta Tagging delle risorse Amazon RDS.	19 agosto 2016
Nuova caratteristica	Amazon RDS ora supporta il modello Licenza inclusa per Oracle Standard Edition Two. Per ulteriori informazioni, consulta Creazione di un'istanza database Amazon RDS. Ora puoi modificare il modello della licenza delle istanze database Amazon RDS che eseguono Microsoft SQL Server e Oracle. Per ulteriori informazioni, consulta Licenza per Microsoft SQL Server su Amazon RDS e Opzioni di licenza per RDS per Oracle.	5 agosto 2016

Modifica	Descrizione	Data della modifica
Nuova caratteristica	Amazon RDS ora supporta il backup nativo e il ripristino dei database Microsoft SQL Server tramite file di backup completi (file .bak). Ora puoi migrare facilmente i database di SQL Server su Amazon RDS e importare ed esportare i database in un unico file facilmente trasportabile, utilizzando Amazon S3 per lo storage e la crittografia. AWS KMS Per ulteriori informazioni, consulta Importazione ed esportazione di database SQL Server mediante backup e ripristino nativi.	27 luglio 2016
Nuova caratteristica	Puoi ora copiare i file di origine da un database MySQL a un bucket Amazon Simple Storage Service (Amazon S3) e quindi ripristinare un cluster di database Amazon Aurora da questi file. Questa opzione può essere molto più rapida rispetto alla migrazione dei dati con `mysqldump` . Per ulteriori informazioni, consulta la pagina relativa alla migrazione di dati da un database MySQL esterno a un cluster di database Aurora MySQL.	20 luglio 2016
Nuova caratteristica	Ora puoi ripristinare uno snapshot del cluster Amazon Aurora DB non crittografato per creare un cluster Amazon Aurora DB crittografato includendo AWS Key Management Service una chiave di crittografia AWS KMS() durante l'operazione di ripristino. Per ulteriori informazioni, consulta Crittografia delle risorse Amazon RDS.	30 giugno 2016
Nuova caratteristica	Puoi utilizzare Oracle Repository Creation Utility (RCU) per creare un repository in Amazon RDS for Oracle. Per ulteriori informazioni, consulta Utilizzo di Oracle Repository Creation Utility in RDS for Oracle .	17 giugno 2016

Modifica	Descrizione	Data della modifica
Nuova caratteristica	Aggiunta del supporto per le repliche di lettura fra regioni PostgreSQL. Per ulteriori informazioni, consulta Creazione di una replica di lettura in un altro Regione AWS.	16 giugno 2016
Nuova caratteristica	È ora possibile utilizzare il AWS Management Console per aggiungere facilmente Multi-AZ con Mirroring a un'istanza DB di Microsoft SQL Server. Per ulteriori informazioni, consulta Aggiunta di Multi-AZ a un'istanza database di Microsoft SQL Server.	9 giugno 2016
Nuova caratteristica	Ora puoi utilizzare le implementazioni Multi-AZ con servizio di mirroring di SQL Server anche nelle seguenti regioni: Asia Pacifico (Sydney), Asia Pacifico (Tokyo) e Sud America (San Paolo). Per ulteriori informazioni, consulta Implementazioni Multi-AZ per Amazon RDS for Microsoft SQL Server.	9 giugno 2016
Nuova caratteristica	Versione aggiornata per il supporto di MariaDB versione 10.1. Per ulteriori informazioni, consulta Amazon RDS for MariaDB.	1 giugno 2016
Nuova caratteristica	Versione aggiornata per il supporto dei cluster di database fra regioni Amazon Aurora che sono repliche di lettura. Per ulteriori informazioni, consulta Replica di cluster di database Aurora MySQL tra regioni AWS.	1 giugno 2016
Nuova caratteristica	Monitoraggio avanzato ora disponibile per le istanze database Oracle. Per ulteriori informazioni, consulta Monitoraggio dei parametri del sistema operativo con il monitoraggio avanzato e Modifica di un'istanza database Amazon RDS.	27 maggio 2016

Modifica	Descrizione	Data della modifica
Nuova caratteristica	Versione aggiornata per il supporto della condivisione manuale delle snapshot dei cluster di database Amazon Aurora. Per ulteriori informazioni, consulta Condivisione di uno snapshot cluster di database.	18 maggio 2016
Nuova caratteristica	Ora puoi utilizzare il plug-in per audit MariaDB per registrare le attività nelle istanze database MariaDB e MySQL. Per ulteriori informazioni, consulta Opzioni per il motore di database MariaDB e Opzioni per le istanze database MySQL.	27 Aprile 2016
Nuova caratteristica	Sono ora disponibili aggiornamenti delle versioni principali per l'aggiornamento da MySQL versione 5.6 alla versione 5.7. Per ulteriori informazioni, consulta Aggiornamento del motore di database MySQL.	26 Aprile 2016
Nuova caratteristica	Monitoraggio avanzato ora disponibile per le istanze database Microsoft SQL Server. Per ulteriori informazioni, consulta Monitoraggio dei parametri del sistema operativo con il monitoraggio avanzato.	22 Aprile 2016
Nuova caratteristica	Versione aggiornata per offrire una visualizzazione Clusters (Cluster) di Amazon Aurora nella console Amazon RDS. Per ulteriori informazioni, consulta la pagina relativa alla visualizzazione di un cluster di database Aurora.	1 Aprile 2016
Nuova caratteristica	Versione aggiornata per il supporto di Multi-AZ in SQL Server con mirroring nella regione Asia Pacifico (Seoul). Per ulteriori informazioni, consulta Implementazioni Multi-AZ per Amazon RDS for Microsoft SQL Server.	31 marzo 2016

Modifica	Descrizione	Data della modifica
Nuova caratteristica	Versione aggiornata per il supporto di Multi-AZ in Amazon Aurora con mirroring nella regione Asia Pacifico (Seoul). Per ulteriori informazioni, consulta la pagina relativa alla disponibilità per Amazon Aurora MySQL.	31 marzo 2016
Nuova caratteristica	Le istanze database PostgreSQL possono richiedere la connessione per l'utilizzo dell'SSL. Per ulteriori informazioni, consulta Utilizzo del protocollo SSL con un'istanza database PostgreSQL.	25 marzo 2016
Nuova caratteristica	Monitoraggio avanzato ora disponibile per le istanze database PostgreSQL. Per ulteriori informazioni, consulta Monitoraggio dei parametri del sistema operativo con il monitoraggio avanzato.	25 marzo 2016
Nuova caratteristica	Le istanze database di Microsoft SQL Server ora possono utilizzare l'autenticazione Windows per l'autenticazione utente. Per ulteriori informazioni, consulta Utilizzo di Active Directory gestito da AWS con RDS per SQL Server.	23 marzo 2016
Nuova caratteristica	Monitoraggio avanzato è ora disponibile nella regione Asia Pacifico (Seoul). Per ulteriori informazioni, consulta Monitoraggio dei parametri del sistema operativo con il monitoraggio avanzato.	16 marzo 2016
Nuova caratteristica	Ora puoi personalizzare l'ordine di promozione a istanze principali delle repliche di Aurora durante un failover. Per ulteriori informazioni, consulta la pagina relativa alla tolleranza ai guasti per un cluster di database Aurora.	14 marzo 2016

Modifica	Descrizione	Data della modifica
Nuova caratteristica	Versione aggiornata per il supporto della crittografia durante la migrazione a un cluster di database Aurora. Per ulteriori informazioni, consulta la pagina relativa alla migrazione di dati in un cluster di database Aurora.	2 marzo 2016
Nuova caratteristica	Versione aggiornata per il supporto del fuso orario locale per i cluster di database Aurora. Per ulteriori informazioni, consulta la pagina relativa al fuso orario locale per i cluster di database Aurora.	1 marzo 2016
Nuova caratteristica	Versione aggiornata per l'aggiunta del supporto di MySQL versione 5.7 per le classi di istanze database Amazon RDS della generazione corrente.	22 febbraio 2016
Nuova caratteristica	Aggiornato per supportare le classi di istanze DB db.r3 e db.t2 nella regione (Stati Uniti occidentali). AWS GovCloud	11 febbraio 2016
Nuova caratteristica	Versione aggiornata per il supporto delle copie crittografate delle snapshot DB e la condivisione delle snapshot DB crittografate. Per ulteriori informazioni, consulta Copia di una snapshot DB. e Condivisione di uno snapshot del database.	11 febbraio 2016
Nuova caratteristica	Versione aggiornata per il supporto di Amazon Aurora nella regione Asia Pacifico (Sydney). Per ulteriori informazioni, consulta la pagina relativa alla disponibilità per Amazon Aurora MySQL.	11 febbraio 2016
Nuova caratteristica	Versione aggiornata per il supporto di SSL per le istanze database Oracle. Per ulteriori informazioni, consulta Utilizzo di SSL con un'istanza database RDS per Oracle.	9 febbraio 2016

Modifica	Descrizione	Data della modifica
Nuova caratteristica	Versione aggiornata per il supporto del fuso orario locale per le istanze database MySQL e MariaDB. Per ulteriori informazioni, consulta Fuso orario locale per le istanze database MySQL e Fuso orario locale per le istanze database MariaDB.	21 dicembre 2015
Nuova caratteristica	Versione aggiornata per il supporto di Monitoraggio avanzato dei parametri OS delle istanze database MySQL e MariaDB e dei cluster di database Aurora. Per ulteriori informazioni, consulta Visualizzazione dei parametri nella console Amazon RDS.	18 dicembre 2015
Nuova caratteristica	Versione aggiornata per il supporto delle classi di istanza database db.t2, db.r3 e db.m4 per MySQL versione 5.5. Per ulteriori informazioni, consulta Classi di istanze database.	4 dicembre 2015
Nuova caratteristica	Versione aggiornata per il supporto della modifica della porta del database per un'istanza database esistente.	3 dicembre 2015
Nuova caratteristica	Versione aggiornata per il supporto degli aggiornamenti delle versioni principali del motore del database per le istanze PostgreSQL. Per ulteriori informazioni, consulta Aggiornamento del motore del database PostgreSQL per Amazon RDS.	19 Novembre 2015
Nuova caratteristica	Versione aggiornata per il supporto dell'accessibilità pubblica di un'istanza database esistente. Versione aggiornata per il supporto delle classi di istanze database standard db.m4.	11 Novembre 2015
Nuova caratteristica	Versione aggiornata per il supporto della condivisione manuale delle snapshot DB. Per ulteriori informazioni, consulta Condivisione di uno snapshot del database.	28 Ottobre 2015

Modifica	Descrizione	Data della modifica
Nuova caratteristica	Versione aggiornata per il supporto di Microsoft SQL Server 2014 per le edizioni Web, Express e Standard.	26 ottobre 2015
Nuova caratteristica	Versione aggiornata per il supporto del motore del database MariaDB basato su MySQL. Per ulteriori informazioni, consulta Amazon RDS for MariaDB.	7 Ottobre 2015
Nuova caratteristica	Versione aggiornata per il supporto di Amazon Aurora nella regione Asia Pacifico (Tokyo). Per ulteriori informazioni, consulta la pagina relativa alla disponibilità per Amazon Aurora MySQL.	7 Ottobre 2015
Nuova caratteristica	Versione aggiornata per il supporto delle classi di istanze database db.t2 con ottimizzazione delle prestazioni per tutti i motori di database e l'aggiunta di classi di istanze database db.t2.large. Per ulteriori informazioni, consulta Classi di istanze database.	25 settembre 2015
Nuova caratteristica	Versione aggiornata per il supporto delle istanze database Oracle nelle classi di istanze database R3 e T2. Per ulteriori informazioni, consulta Classi di istanze database.	5 agosto 2015
Nuova caratteristica	Microsoft SQL Server Enterprise Edition è ora disponibile con il modello di servizio Licenza inclusa. Per ulteriori informazioni, consulta Licenza per Microsoft SQL Server su Amazon RDS.	29 luglio 2015
Nuova caratteristica	Amazon Aurora è stato pubblicato ufficialmente. Il motore database Amazon Aurora supporta più istanze database in un cluster di database. Per informazioni dettagliate, consulta la pagina Che cos'è Amazon Aurora?.	27 luglio 2015

Modifica	Descrizione	Data della modifica
Nuova caratteristica	Versione aggiornata per il supporto della copia di tag nelle snapshot DB.	20 luglio 2015
Nuova caratteristica	Versione aggiornata per il supporto dell'aumento delle dimensioni dello storage di tutti i motori di database e dell'aumento degli Provisioned IOPS per SQL Server.	18 giugno 2015
Nuova caratteristica	Opzioni aggiornate per le istanze database riservate.	15 giugno 2015
Nuova caratteristica	Versione aggiornata per il supporto all'utilizzo di Amazon CloudHSM con le istanze database Oracle tramite TDE.	8 gennaio 2015
Nuova caratteristica	Versione aggiornata per il supporto alla crittografia dei dati inattivi della nuova API versione 2014-10-31.	6 gennaio 2015
Nuova caratteristica	Versione aggiornata per l'inclusione del nuovo motore database Amazon: Aurora. Il motore database Amazon Aurora supporta più istanze database in un cluster di database. Amazon Aurora è in versione di anteprima ed è soggetta a modifiche. Per informazioni dettagliate, consulta la pagina Che cos'è Amazon Aurora?.	12 Novembre 2014
Nuova caratteristica	Versione aggiornata per il supporto delle repliche di lettura PostgreSQL.	10 Novembre 2014
Nuove API e caratteristiche	Versione aggiornata per il supporto del tipo di storage GP2 e la nuova API versione 2014-09-01. Versione aggiornata per il supporto della possibilità di copiare un'opzione o un gruppo di parametri esistente per creare una nuova opzione o un nuovo gruppo di parametri.	7 Ottobre 2014

Modifica	Descrizione	Data della modifica
Nuova caratteristica	Versione aggiornata per il supporto di InnoDB Cache Warming per le istanze database che eseguono MySQL versione 5.6.19 e successive.	3 settembre 2014
Nuova caratteristica	Versione aggiornata per il supporto della verifica dei certificati SSL durante la connessione ai motori di database MySQL versione 5.6, SQL Server e PostgreSQL.	5 agosto 2014
Nuova caratteristica	Versione aggiornata per il supporto delle classi di istanze database db.t2.	4 agosto 2014
Nuova caratteristica	Versione aggiornata per il supporto delle classi di istanze database db.r3 con ottimizzazione della memoria per l'uso con i motori di database MySQL (versione 5.6), SQL Server e PostgreSQL.	28 maggio 2014
Nuova caratteristica	Versione aggiornata per il supporto delle implementazioni Multi-AZ di SQL Server tramite il servizio di mirroring di SQL Server.	19 maggio 2014
Nuova caratteristica	Versione aggiornata per il supporto di aggiornamenti da MySQL versione 5.5 alla versione 5.6.	23 Aprile 2014
Nuova caratteristica	Aggiornato per supportare Oracle. GoldenGate	3 Aprile 2014
Nuova caratteristica	Versione aggiornata per il supporto delle istanze database M3.	20 febbraio 2014
Nuova caratteristica	Versione aggiornata dell'opzione Timezone (Fuso orario) di Oracle.	13 gennaio 2014
Nuova caratteristica	Versione aggiornata per il supporto di replica fra istanze database MySQL in regioni diverse.	26 Novembre 2013

Modifica	Descrizione	Data della modifica
Nuova caratteristica	Versione aggiornata per il supporto motore del database PostgreSQL.	14 Novembre 2013
Nuova caratteristica	Versione aggiornata per il supporto di Transparent Data Encryption (TDE) di SQL Server	7 Novembre 2013
Nuova API e nuova caratteristica	Versione aggiornata per il supporto delle copie di snapshot DB fra le regini; nuova versione API, 09-09-2013.	31 ottobre 2013
Nuove caratteristiche	Versione aggiornata per il supporto di Oracle Statspack.	26 settembre 2013
Nuove caratteristiche	Versione aggiornata per il supporto dell'uso di repliche per importare o esportare dati tra istanze di MySQL in esecuzione in Amazon RDS e istanze di MySQL in esecuzione in locale o in Amazon EC2.	5 settembre 2013
Nuove caratteristiche	Versione aggiornata per il supporto della classe di istanze database db.cr1.8xlarge per MySQL 5.6.	4 settembre 2013
Nuova caratteristica	Versione aggiornata per il supporto della replica delle repliche di lettura.	28 agosto 2013
Nuova caratteristica	Versione aggiornata per il supporto della creazione della replica di lettura parallela.	22 luglio 2013
Nuova caratteristica	Versione aggiornata per il supporto di autorizzazioni e tagging fine-grained per tutte le risorse Amazon RDS.	8 luglio 2013
Nuova caratteristica	Versione aggiornata per il supporto di MySQL 5.6 per le nuove istanze, incluso il supporto dell'interfaccia Memcached MySQL 5.6 e l'accesso ai log binari.	1 luglio 2013
Nuova caratteristica	Versione aggiornata per il supporto di aggiornamenti delle versioni principali da MySQL 5.1 a MySQL 5.5.	20 giugno 2013

Modifica	Descrizione	Data della modifica
Nuova caratteristica	Gruppi di parametri database aggiornati per permettere le espressioni per i valori dei parametri.	20 giugno 2013
Nuova API e nuova caratteristica	Versione aggiornata per il supporto dello stato delle repliche di lettura; nuova versione API, 15-05-2013.	23 maggio 2013
Nuove caratteristiche	Versione aggiornata per il supporto delle funzioni di Oracle Advanced Security per la crittografia di rete nativa e di Oracle Transparent Data Encryption.	18 Aprile 2013
Nuove caratteristiche	Versione aggiornata per il supporto degli aggiornamenti delle versioni principali di SQL Server e ulteriore funzionalità di Provisioned IOPS.	13 marzo 2013
Nuova caratteristica	Versione aggiornata per il supporto di VPC per impostazione predefinita per RDS.	11 marzo 2013
Nuova API e caratteristica	Versione aggiornata per il supporto dell'accesso al log; nuova versione API, 12-02-2013.	4 marzo 2013
Nuova caratteristica	Versione aggiornata per il supporto dell'abbonamento alle notifiche degli eventi RDS.	4 febbraio 2013
Nuova API e caratteristica	Versione aggiornata per il supporto delle ridenominazione dell'istanza database e la migrazione dei membri del gruppo di sicurezza DB di un VPC in un gruppo di sicurezza VPC.	14 gennaio 2013
Nuova caratteristica	Aggiornato per il supporto AWS GovCloud (Stati Uniti occidentali).	17 dicembre 2012
Nuova caratteristica	Versione aggiornata per il supporto delle classi di istanza database m1.medium e m1.xlarge.	6 Novembre 2012

Modifica	Descrizione	Data della modifica
Nuova caratteristica	Versione aggiornata per il supporto della promozione delle repliche di lettura.	11 Ottobre 2012
Nuova caratteristica	Versione aggiornata per il supporto SSL nelle istanze database Microsoft SQL Server.	10 Ottobre 2012
Nuova caratteristica	Versione aggiornata per il supporto delle istanze database micro Oracle.	27 settembre 2012
Nuova caratteristica	Versione aggiornata per il supporto di SQL Server 2012.	26 settembre 2012
Nuova API e caratteristica	Versione aggiornata per il supporto di Provisioned IOPS. Versione API 17-09-2012.	25 settembre 2012
Nuove caratteristiche	Versione aggiornata per il supporto di SQL Server per le istanze database in VPC e il supporto Oracle per Data Pump.	13 settembre 2012
Nuova caratteristica	Versione aggiornata per il supporto di SQL Server Agent.	22 agosto 2012
Nuova caratteristica	Versione aggiornata per il supporto del tagging delle istanze database.	21 agosto 2012
Nuove caratteristiche	Versione aggiornata per il supporto di Oracle APEX e XML DB, dei fusi orari Oracle e delle istanze database Oracle in un VPC.	16 agosto 2012
Nuove caratteristiche	Versione aggiornata per il supporto delle istanze database SQL Server Database Engine Tuning Advisor e Oracle in VPC.	18 luglio 2012
Nuova caratteristica	Versione aggiornata per il supporto dei gruppi di opzioni e della prima opzione, Oracle Enterprise Manager Database Control.	29 maggio 2012

Modifica	Descrizione	Data della modifica
Nuova caratteristica	Versione aggiornata per il supporto delle repliche di lettura in Amazon Virtual Private Cloud.	17 maggio 2012
Nuova caratteristica	Versione aggiornata per il supporto di Microsoft SQL Server.	8 maggio 2012
Nuove caratteristiche	Versione aggiornata per il supporto del failover forzato, implementazione multi-AZ delle istanze database Oracle e set di caratteri non predefiniti delle istanze database Oracle.	2 maggio 2012
Nuova caratteristica	Versione aggiornata per il supporto di Amazon Virtual Private Cloud (VPC).	13 febbraio 2012
Contenuti aggiornati	Versione aggiornata per i nuovi tipi di istanza riservata.	19 dicembre 2011
Nuova caratteristica	Versione aggiornata per il supporto del motore Oracle.	23 maggio 2011
Contenuti aggiornati	Aggiornamenti alla console.	13 maggio 2011
Contenuti aggiornati	Contenuti modificati per la visualizzazione abbreviata delle finestre di manutenzione e backup.	28 febbraio 2011
Nuova caratteristica	Aggiunto supporto di MySQL 5.5.	31 gennaio 2011
Nuova caratteristica	Aggiunto supporto delle repliche di lettura.	4 Ottobre 2010
Nuova caratteristica	È stato aggiunto il supporto per AWS Identity and Access Management (IAM).	2 settembre 2010

Modifica	Descrizione	Data della modifica
Nuova caratteristica	Aggiunta della gestione della versione del motore di database.	16 agosto 2010
Nuova caratteristica	Aggiunta delle istanze database riservate.	16 agosto 2010
Nuova caratteristica	Amazon RDS ora supporta le connessioni SSL alle istanze database.	28 giugno 2010
Nuova guida	La prima versione della Guida per l'utente di Amazon RDS.	7 giugno 2010

AWS Glossario

Per la AWS terminologia più recente, consultate il AWS glossario nella sezione Reference.Glossario AWS

Le traduzioni sono generate tramite traduzione automatica. In caso di conflitto tra il contenuto di una traduzione e la versione originale in Inglese, quest'ultima prevarrà.